



# Monitorización de experiencia de usuario UX y WUX



om:

<https://pandorafms.com/manual/!776/>

permanent link:

[https://pandorafms.com/manual/!776/es/documentation/pandorafms/monitoring/13\\_user\\_monitorization](https://pandorafms.com/manual/!776/es/documentation/pandorafms/monitoring/13_user_monitorization)

2024/06/10 14:34



# Monitorización de experiencia de usuario UX y WUX

## Introducción

**E** La monitorización de experiencia de usuario consiste en realizar grabaciones de tareas automatizadas tanto de navegación web (PWR/WUX) como de interacción con el escritorio y las aplicaciones del sistema Windows (PDR). Estas grabaciones pueden consistir desde un clic de un ratón sobre un navegador web, escribir un texto o realizar una búsqueda en la web, hasta abrir una aplicación en el escritorio. Esto permite configurar las funciones y grabarlas de manera automática para su posterior ejecución en busca de resultados.

## Diferencias entre monitorización UX y WUX

Ambos sistemas de monitorización sirven para la ejecución de tareas automatizadas de navegación web mediante el sistema Pandora Web Robot Daemon (PWRD).

El sistema UX realiza estas monitorizaciones mediante el Agente Software instalado en una máquina, mientras que la monitorización WUX (*Web User Experience*) se basa en un servidor integrado dentro de Pandora FMS.

PWRD es un servicio que provee las herramientas necesarias para automatizar las sesiones de navegación web. Para ello utiliza un archivo que contiene una lista de las acciones necesarias para navegar por el portal web a monitorizar.

Adicionalmente, la monitorización UX permite la ejecución de tareas automatizadas que interactúan con el escritorio y las aplicaciones del sistema MS Windows®. Este tipo de tareas no se pueden realizar con WUX.

Pandora FMS UX es un sistema que ejecuta tareas automatizadas, entregando a Pandora FMS un informe con los resultados de las ejecuciones, tiempos empleados y capturas de pantalla con los posibles errores encontrados.

Si dispone de un sistema de automatización de tareas, Pandora FMS UX también permite ejecutar los guiones (*scripts*) que ya tenga realizados y monitorizar su ejecución.

Es importante concretar el uso de este tipo de monitorización. Ambos métodos se basan en la ejecución de las monitorizaciones mediante complemento (*plugin*) en los Agentes Software instalados en las máquinas correspondientes.

Para la ejecución de transacciones web Pandora FMS utiliza Selenium Engine:

- Selenium versión 2.
- Selenium versión 3.

## Monitorización PWR UX

Teniendo en cuenta las **diferencias entre monitorización UX y WUX**, a grandes rasgos la monitorización UX debe seguir una serie de prerequisites relacionados al entorno a monitorizar, con los siguientes pasos previos:

- Instalar Java®.
- Configurar un perfil en Mozilla Firefox®.
- Instalar un servicio Selenium®.
- Distribuir PWR en el sistema.
- Instalar el IDE de Selenium® para Mozilla Firefox®.
- Grabar una sesión PWR.

Para la ejecución estándar de sesiones pregrabadas utilice el **Plugin Pandora UX** en su última versión 28-04-2022.

## Experiencia de Usuario Web (WUX) con Selenium 3

Para el despliegue de Selenium en los servidores WUX se utilizará un pila o *stack* de contenedores para un rápido despliegue y un fácil escalado.

### Configuraciones previas

Deben estar instalados Docker y Docker Compose; *se recomienda el uso de Rocky Linux 8 como sistema operativo base.*

Para instalar Docker siga la documentación, de acuerdo a su sistema operativo:

<https://docs.docker.com/engine/install/>

Para Selenium utilice las imágenes oficiales para la instalación y despliegue del *stack*:

<https://hub.docker.com/u/selenium>

En el repositorio de Selenium están disponibles distintas imágenes con navegadores. Para Pandora FMS se recomienda contenedores de Mozilla Firefox® y Google Chrome®.

## Despliegue del stack de Selenium

Para desplegar el *stack* de Selenium debe crear un fichero en formato YAML con la configuración necesaria:

```
# To execute this docker-compose yml file use `docker-compose -f up`
# Add the `-d` flag at the end for detached execution
version: "3"
services:
  selenium-hub:
    image: pandorafms/pandorafms-selenium-hub
    mem_limit: 2G
    container_name: selenium-hub-v3
    logging:
      driver: "json-file"
      options:
        max-file: "5"
        max-size: "4m"
        mode: "non-blocking"
    environment:
      - TZ=Europe/Amsterdam
    ports:
      - "4444:4444"

  chrome:
    image: pandorafms/pandorafms-selenium-node-chrome
    mem_limit: 2G
    volumes:
      - /dev/shm:/dev/shm
    depends_on:
      - selenium-hub
    logging:
      driver: "json-file"
      options:
        max-file: "5"
        max-size: "4m"
        mode: "non-blocking"
    environment:
      - TZ=Europe/Amsterdam
      - HUB_HOST=selenium-hub
      - HUB_PORT=4444

  firefox:
    image: pandorafms/pandorafms-selenium-node-firefox
    mem_limit: 2G
    volumes:
```

```
- /dev/shm:/dev/shm
depends_on:
  - selenium-hub
logging:
  driver: "json-file"
  options:
    max-file: "5"
    max-size: "4m"
    mode: "non-blocking"
environment:
  - TZ=Europe/Amsterdam
  - HUB_HOST=selenium-hub
  - HUB_PORT=4444
```

En el ejemplo anterior debe realizar las modificaciones necesarias para cada caso, tales como límite de memoria, zona horaria, cantidad de nodos, etcétera.

Puede guardar el fichero con el nombre `docker-compose.yml` para hacer más fácil su identificación.

Para levantar el contenedor con la configuración que ha definido, basta con ejecutar (sustituya `<compose-file>` con el nombre que escogió para el archivo):

```
docker-compose -f <compose-file> up -d
```

Para comprobar los servicios corriendo en el contenedor use:

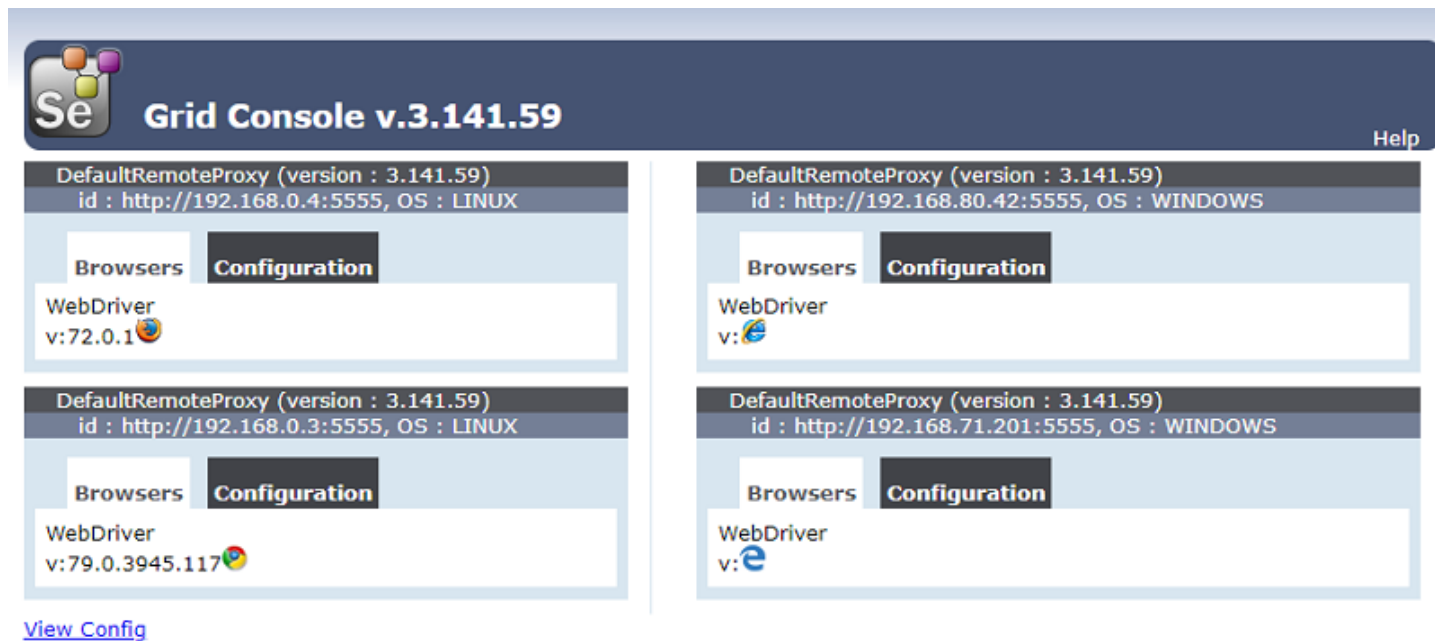
```
docker-compose -f <compose-file> ps
```

Para ver el estado y los *logs* de los servicios de Selenium utilice el siguiente comando:

```
docker-compose -f <compose-file> logs
```

Una vez realizadas las comprobaciones pertinentes, para verificar que el *grid* está funcionando correctamente y los *workers* se han registrado tal y como hemos definido en el fichero de configuración será necesario acceder a la siguiente URL:

```
http://<ip_selenium_server>:4444/grid/console
```



The screenshot displays the Selenium Grid Console v.3.141.59 interface. At the top left is the Selenium logo (Se) and the version number. A 'Help' link is located at the top right. The main area is divided into four panels, each representing a worker node. Each panel has a header with 'DefaultRemoteProxy (version : 3.141.59)' and an 'id' field. Below the header are two tabs: 'Browsers' and 'Configuration'. The 'WebDriver' version and browser icon are shown in the 'Browsers' tab.

Worker ID	OS	WebDriver Version	Browser Icon
http://192.168.0.4:5555	LINUX	v:72.0.1	Chrome
http://192.168.80.42:5555	WINDOWS	v:	Edge
http://192.168.0.3:5555	LINUX	v:79.0.3945.117	Chrome
http://192.168.71.201:5555	WINDOWS	v:e	Edge

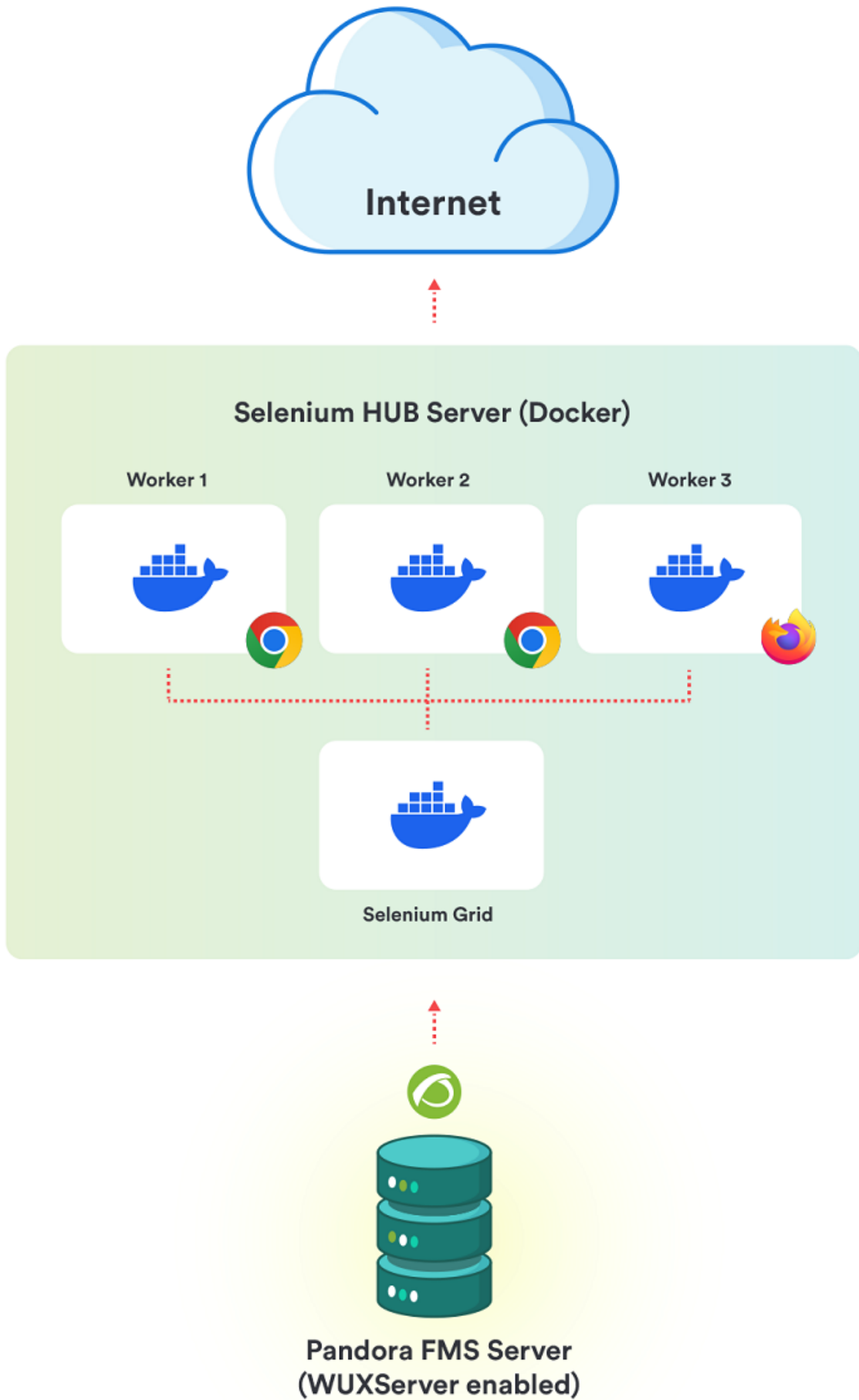
[View Config](#)

Si necesita aumentar el número de *workers*:

```
docker-compose -f <compose-file> scale chrome=X firefox=Y
```

## Infraestructura del servicio Selenium

Selenium funciona como un HUB en el que se levanta un contenedor que actúa como *grid* o cuadrícula donde desplegar los contenedores *worker* que se necesiten.





## Configuración de Pandora FMS

Para la utilización del modo centralizado (WUX) es necesario aplicar la siguiente configuración al servidor de Pandora FMS.

Independientemente del modo elegido, una vez iniciado puede empezar a asignar ejecuciones de sesiones de navegación, agregando los parámetros de configuración de WUX Server al fichero de configuración del servidor de Pandora FMS.

Agregue la siguiente configuración al final del fichero `/etc/pandora/pandora_server.conf` (sustituya `<ip_wux_host>` por la dirección IP del servidor, si es el mismo servidor que corre el Servidor Pandora FMS use `127.0.0.1`):

```
wuxserver 1
wux_host <ip_wux_host>
wux_port 4444
```

El fichero de configuración del servidor cuenta con un nuevo *token* para limpiar las sesiones que puedan estar encoladas cada vez que se inicia el servidor de Pandora FMS.

```
clean_wux_sessions 1 #(default)
```

## Gestión de hilos de Pandora FMS

- La gestión de los hilos del `wuxserver` se hace automáticamente al iniciar el servicio de `pandora_server`.
- Se realiza teniendo en cuenta el número de nodos de un navegador MENOR que haya en el *hub* de Selenium. Por ejemplo:
  - Si en el *hub* hay configurados 2 nodos de Firefox y 2 de Chrome, el número de hilos del `wuxserver` será de 2.
  - Si en el *hub* hay configurados 1 nodo de Firefox y 4 de Chrome, el número de hilos será de 1.
  - Si en el *hub* hay configurados 6 nodos de Firefox, el número de hilos será de 6.

Tenga en cuenta que cada hilo indica las sesiones que pueden enviarse de manera simultánea desde el `wuxserver` al *hub* de Selenium.

## Anexo: Añadir workers para Internet Explorer y Microsoft Edge

En caso de ser necesario lanzar transacciones web contra navegadores de Microsoft®, es necesario configurar una máquina (física o virtual) con la versión de Windows® deseada y configurar el controlador o *driver* siguiendo la documentación oficial.

Documentación para la instalación del *driver* de Internet Explorer®:

[github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver](https://github.com/SeleniumHQ/selenium/wiki/InternetExplorerDriver)

Recomendamos el uso del driver 3.141 en su versión de 32bit para evitar los problemas de rendimiento de la versión de 64bit.

Documentación para la instalación del *driver* de Microsoft Edge®:

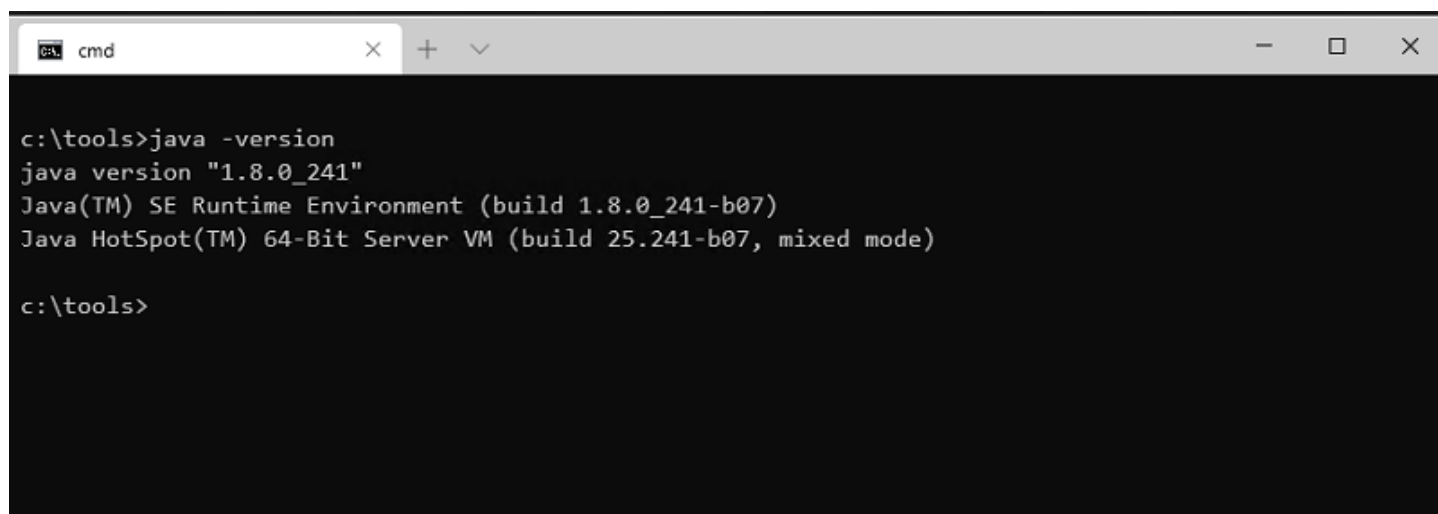
<https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/>

Para la ejecución de Selenium, será necesario tener instalado Java® en el dispositivo Windows®.

Para comprobar si disponemos de Java®, ejecute el comando:

```
java -version
```

Debe obtener una salida parecida a la siguiente:



```
c:\tools>java -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)

c:\tools>
```

Es necesario también el fichero JAR de Selenium para ejecutar el servidor localmente y registrarlo en nuestro *grid*. Puede obtenerlo en:

<https://www.selenium.dev/downloads/>

## Downloads

Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and additional information for Maven users.

[Previous Releases](#)

### Selenium Server (Grid)

The Selenium Server is needed in order to run Remote Selenium WebDriver (Grid).

Latest stable version [3.141.69](#)

To use the Selenium Server in a Grid configuration see the [documentation](#)

Latest Selenium 4 Alpha version [4.0.0-alpha-5](#)

### The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver.

Please make sure that this is available on your %PATH% (or %PATH% on Windows) in order for the IE Driver

Download version 3.150.1 for:

[32 bit Windows IE](#) (recommended) [64 bit Windows IE](#) [CHANGELOG](#)

Para iniciar el servidor Microsoft Edge® se debe ejecutar en una terminal en el directorio donde tengamos el fichero JAR lo siguiente:

```
java -jar selenium-server-standalone-<VER>.jar -port 5555 -role node -hub
http://<ip_selenium_server>:4444/grid/register -browser
"browserName=MicrosoftEdge, platform=WINDOWS, maxInstances=1"
```

Para iniciar el servidor Internet Explorer® el comando es similar, pero se debe especificar la ruta <PATH> del *driver* que ha descargado:

```
java -Dwebdriver.ie.driver=<PATH>IEDriverServer.exe -jar selenium-server-
standalone<VER>.jar -port 5555 -role node -hub
http://ip_selenium_server:4444/grid/register -browser "browserName=internet
explorer, platform=WINDOWS, maxInstances=1"
```

```
c:\Selenium
λ java -Dwebdriver.ie.driver=C:/tools/IEDriverServer.exe -jar selenium-server-standalone-3.141.59.jar
-port 5555 -role node -hub http://192.168.80.44:4444/grid/register -browser "browserName=internet explorer,platform=WINDOWS,maxInstances=1"
14:29:14.742 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
14:29:14.977 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 5555
2020-04-13 14:29:16.258:INFO::main: Logging initialized @2433ms to org.seleniumhq.jetty9.util.log.StdErrLog
14:29:16.742 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
14:29:16.898 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 5555
14:29:16.898 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Selenium Grid node is up and ready to register to the hub
14:29:17.227 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
14:29:18.336 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://192.168.80.44:4444/grid/register
14:29:18.367 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
|
```

Se debe configurar el cortafuegos de MS Windows® para permitir el tráfico en los puertos especificados en el comando de ejecución. En el caso de los ejemplos anteriores: 5555 y 4444.

## Grabación

Debe tener en cuenta que las grabaciones realizadas en Selenium 2 pueden no funcionar correctamente.

Para la versión 3 de Selenium serán compatibles tanto las grabaciones antiguas como las nuevas, aunque no se garantiza la compatibilidad total con versiones antiguas.

Para poder grabar una nueva sesión, se deberá instalar la extensión Selenium IDE en el navegador que se desee.

Firefox

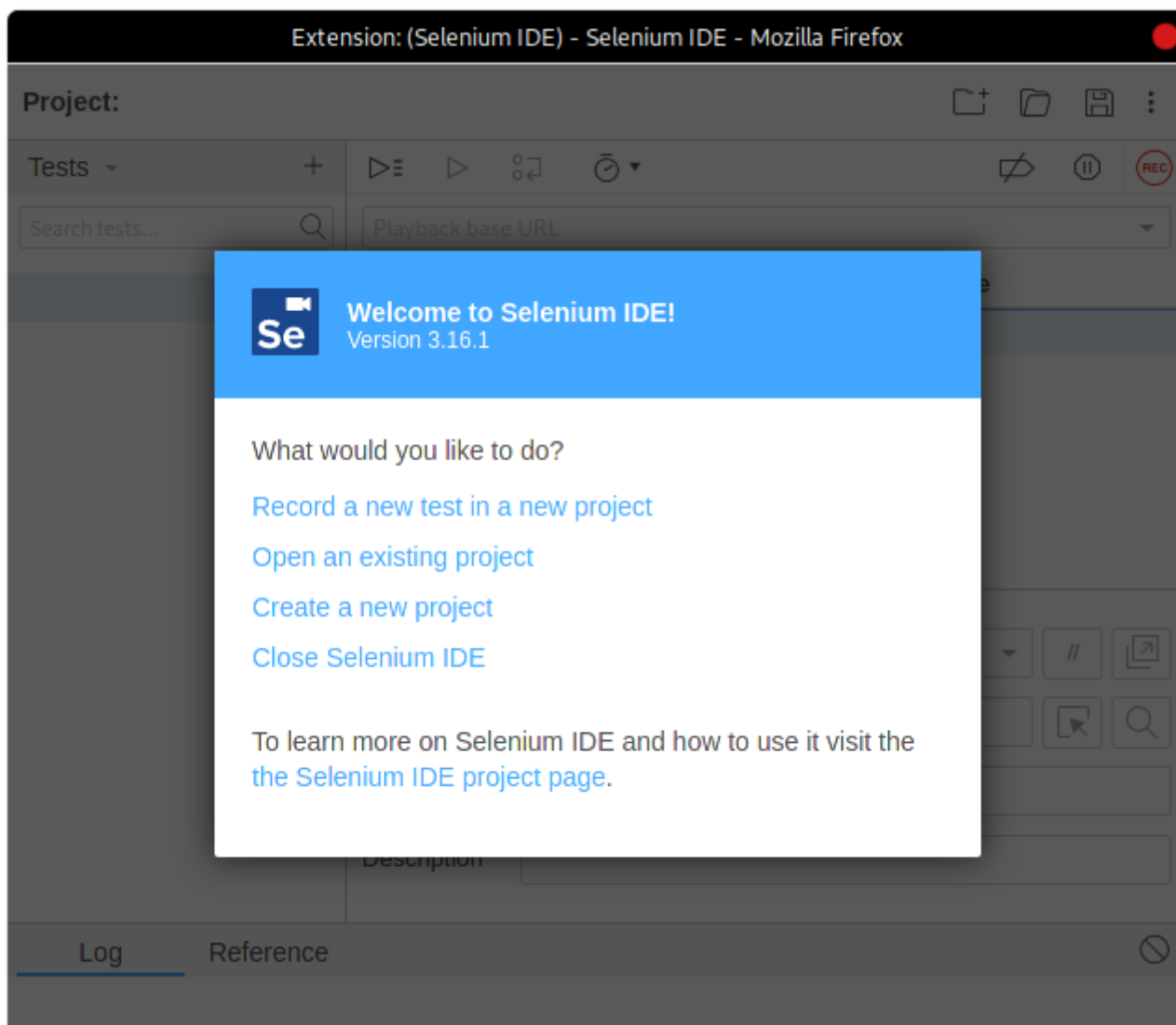
```
https://addons.mozilla.org/es/firefox/addon/selenium-ide/
```

Chrome

```
https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd
```

El icono  será el que permita iniciar el entorno de grabación una vez instalada la extensión.

Al hacer clic abrirá un menú que permite iniciar nuevos proyectos de grabación.



Una vez realizada la grabación se tendrá como resultado algo parecido a esto:

Extension: (Selenium IDE) - Selenium IDE - pruebaswux - Mozilla Firefox

Project: pruebaswux

Tests +

Search tests...

http://varian.artica.lan

	Command	Target	Value
1	// phase_start:Inicio		
2	open	/web	
3	unchecked	name=check1	
4	check	name=check1	
5	// phase_end:Inicio		
6	// phase_start:Captura		
7	store attribute	xpath=//input[@name='check1']@checked	estado
8	store xpath count	xpath=//*[name='check1']	aa
9	store attribute	xpath=//input[@name='check1']@name	name
10	// storeExtraction	input name="(*?)"	inputs
11	// phase_end:Captura		
12	// phase_start:Muestra		
13	echo	\${aa}	
14	echo	\${estado}	
15	if	\${name} != "check1" && \${aa} == 1	

Command  #

Target

Value

Description

Log Reference

Aparte de la funcionalidad provista por Selenium, Pandora FMS dispone de comandos personalizados para extraer información del objetivo a monitorizar.

## Comandos

La lista de comandos compatibles para usarlos en Pandora FMS se puede encontrar en el [siguiente enlace](#).

## Recomendaciones y sugerencias al realizar las grabaciones

A continuación se detalla una serie de recomendaciones y consejos a la hora tanto de realizar la grabación de las transacciones con Selenium IDE como para integrar con Pandora FMS los comandos más complejos:

- Dividir la transacción en fases siempre que sea posible. De esta forma los módulos que se crearán tanto de estado como tiempos y las capturas de pantalla estarán segmentados y será más sencillo localizar dónde ha fallado la transacción.
- Utilizar el comando de Selenium `set speed` y `wait for` para evitar falsos negativos. Al ejecutar una transacción, los comandos de Selenium no tienen ningún retraso o *delay* por defecto desde que termina uno hasta que se ejecuta el siguiente, y algunos tampoco tienen tiempo de finalización o *timeout*. Esto hace que la transacción se ejecute muy rápidamente en el menor tiempo posible, pero al hacerlo tan rápidamente, si la web es algo lenta o tarda un segundo de más en cargar es posible que el chequeo termine en fallo. Por ejemplo, tras ejecutar un comando `click` y cambiar de página, si después tiene un comando que interactúe con un elemento a la nueva página y la carga de esta se retrasa un segundo, no encontrará el elemento en la nueva página y el chequeo acabará en fallo. Para evitar estas situaciones, existe el comando `set speed`, que añade un *delay* con la cantidad de milisegundos que se le indique en el objetivo o *Target* entre cada comando. Se recomienda establecerlo al inicio de la transacción. También existen, para los casos en los que es sabido que la carga de una página o aparición de un elemento se puedan demorar unos segundos, los comandos `wait for element present`, `wait for visible` y `wait for text` en los que puede establecer el tiempo en milisegundos que esperará para que el elemento aparezca en la página antes de marcar la transacción como fallo. Es importante destacar que el uso de estos comandos, si bien aumentan sobremanera la fiabilidad del chequeo, también incrementarán el tiempo que se tarda en ejecutar la transacción.
- Realizar comprobaciones de los elementos. Para esto se usarán comandos como `assert` y `verify`, en sus distintas vertientes. Acabar una transacción con un clic, por ejemplo, no nos garantiza que la nueva página que deba abrir el elemento en el que se hace clic se abra, solo que es posible hacer clic en el elemento. Si después del clic introducimos un `verify text` a un texto que se sepa que solo se va a cargar después de realizar el clic, esa sí sería una forma de comprobar que la página a la que nos manda el clic está disponible.
- Utilizar `store window handle` en transacciones en las que se vaya a cambiar de ventanas. El cambio de ventana (con un `select window`) puede dar fallo si no se ha almacenado previamente un identificador a la ventana inicial.
- Utilizar `xpath` cuando falle el *Target* por identificador de CSS o cuando quiera buscar contenido en la página. Por defecto el grabador de Selenium IDE introduce en el *Target* del elemento el localizador del CSS, pero también graba el localizador por `xpath`, por tanto es posible ver todos los localizadores que guarda si pulsa en la caja del *Target* en el grabador:

Command	assert text	//	[Icon]
Target	css=#\36 fn79666fiykxpysdro91x1ccr_message sp	[Icon]	[Icon]
Value	css=#\36 fn79666fiykxpysdro91x1c cr_message span	css:finder	[Input]
Description	xpath=//div[@id='6fn796 66fiykxpysdro91x1ccr_me ssage']/div/h2/span	xpath:idRelative	[Input]
rovided value. T	xpath=//h2/span	xpath:position	[Input]
	xpath=//span[contains(. , 'Innovadores de la monitorización')]	xpath:innerText	[Input]

Además, utilizando rutas por xpath, es posible buscar textos dentro de las etiquetas de las páginas para realizar grabaciones mucho más dinámicas. En la captura de pantalla anterior observe que es posible usar un xpath que busque el texto “Innovadores de la monitorización” en todas las etiquetas span de la página, no en un locator en concreto.

- Uso correcto del comando `execute script`. Este comando ejecuta un fragmento de código de JavaScript en la ventana en la que está ubicado en la transacción. Se recomienda leer la guía para conocer su uso y las diferentes opciones que ofrece: <https://ui.vision/rpa/docs/selenium-ide/executescript>. No obstante, el uso de las variables almacenadas previamente (por un comando `store text` por ejemplo) debe ir entre comillas dobles para que el `webdriver` de Pandora FMS las interprete correctamente. Aquí hay un ejemplo de una variable almacenada con `store text` y su posterior uso en `execute script` de forma que el servidor de Pandora FMS la interpreta correctamente. Note que este uso de la variable entre comillas fallará al ejecutar el script en el grabador de Selenium IDE:

```
store text      css=.folder:nth-child(1) abbr      test
execute script  if ("${test}".match(/(_thismonth_)(\d)+ago/)) { return 1; } else if ("${test}".match(/(_lastmonth_)(\d)+/)) { return 2; } else { return 0; } x
```

## Experiencia de Usuario Web (WUX) con Selenium 2

Selenium 2 ha quedado obsoleto, la versión actual es Selenium 3.

### Configuraciones previas

#### Selenium

#### Despliegue con Pandora Web Robot Daemon (PWRD)



Pandora FMS Web Robot Daemon es un servicio que provee las herramientas necesarias para automatizar las sesiones de navegación web, utilizando un archivo que contiene una lista de las acciones necesarias para navegar por el portal web del usuario.

Viene integrado con el servidor de Pandora FMS y lo podemos encontrar en la carpeta `/usr/share/pandora_server/util/pwr` al instalar el servidor Enterprise (Linux) o en la biblioteca de módulos (Windows).

Para descargarlo

<https://pandorafms.com/library/pandora-ux-and-wux-pwr-reorder/>

Contiene:

- Binario del navegador Firefox versión 47.
- Perfil preconstruído para la ejecución de sesiones de navegación web.
- Servidor de automatización de sesiones (Selenium server).

#### **Despliegue del servidor Selenium en sistemas Windows**

Prerrequisitos:

- Instalar Java® 1.8 en la máquina que dará el servicio.
- Instalar Firefox® 47.0.1 (descargable en <https://ftp.mozilla.org/pub/firefox/releases/47.0.1/>).
- Preparar un perfil de Firefox® que se utilizará para probar las sesiones automatizadas (opcional): <http://toolsqa.com/selenium-webdriver/custom-firefox-profile/>
- Crear el siguiente directorio: `C:\PWR`.

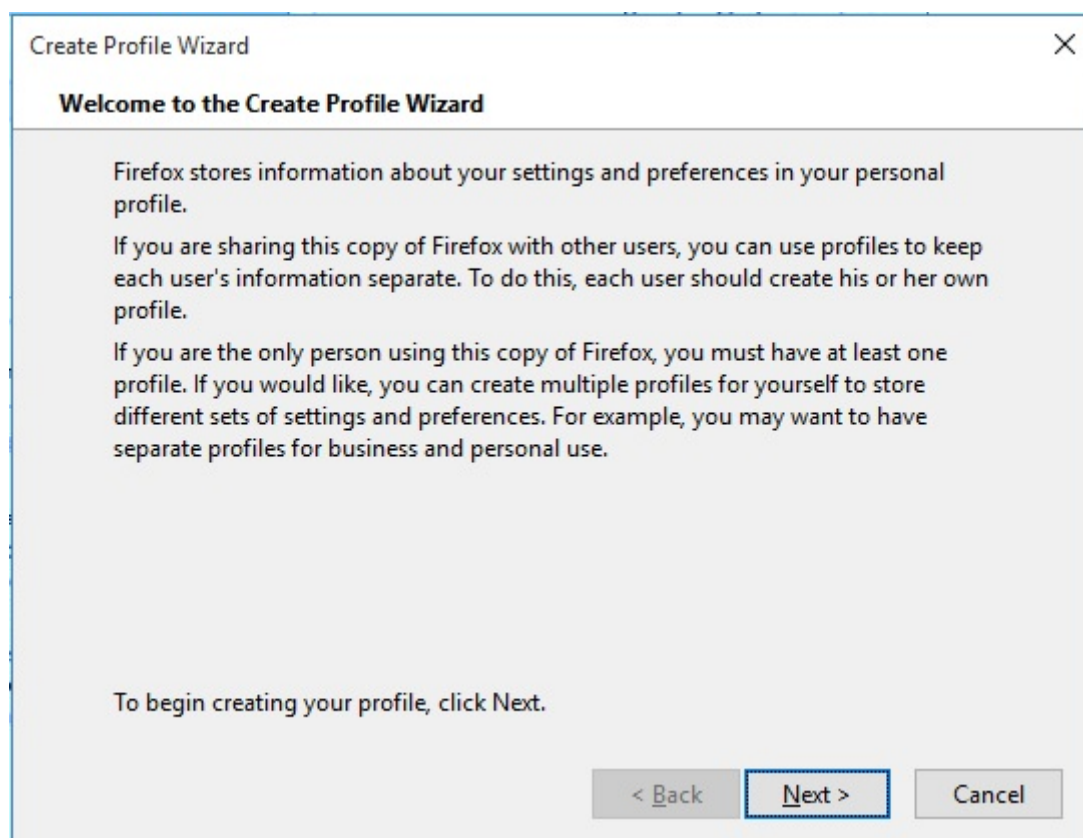
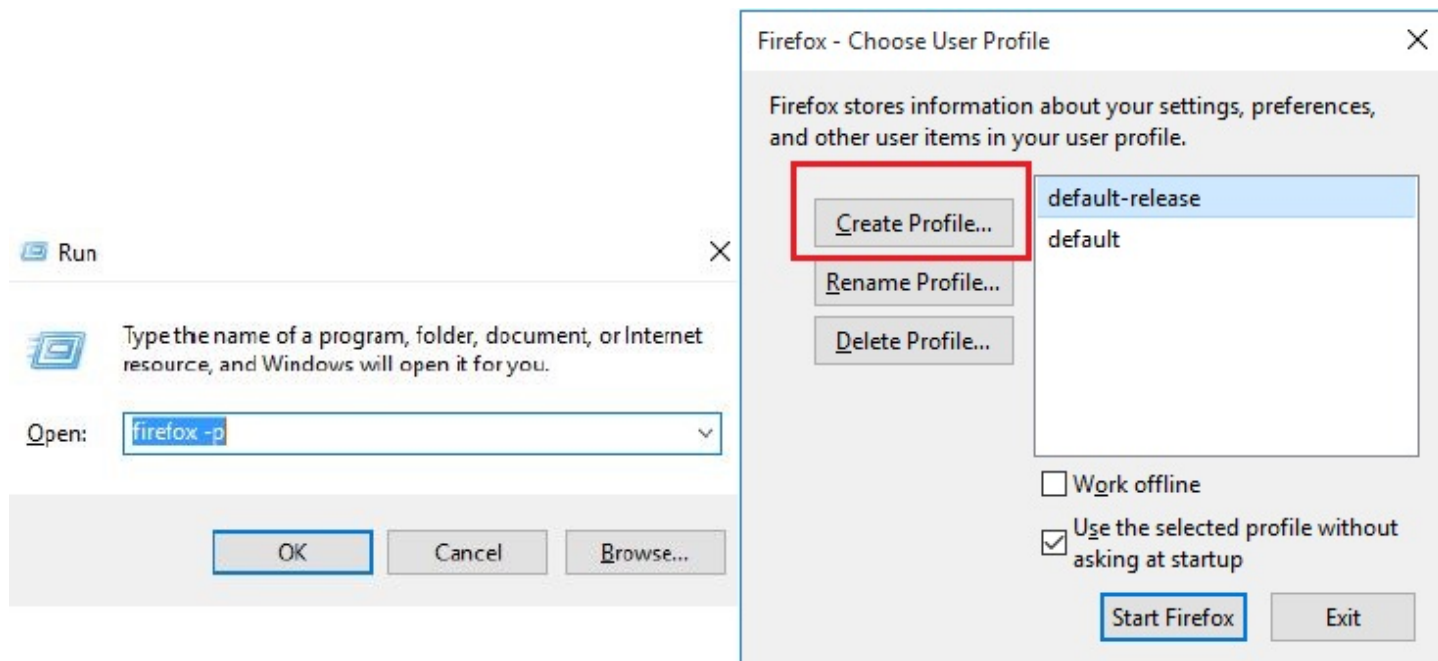
Para descargar el `PWR_Server.zip`:

<https://pandorafms.com/library/pwr-server-for-ux-monitoring/>

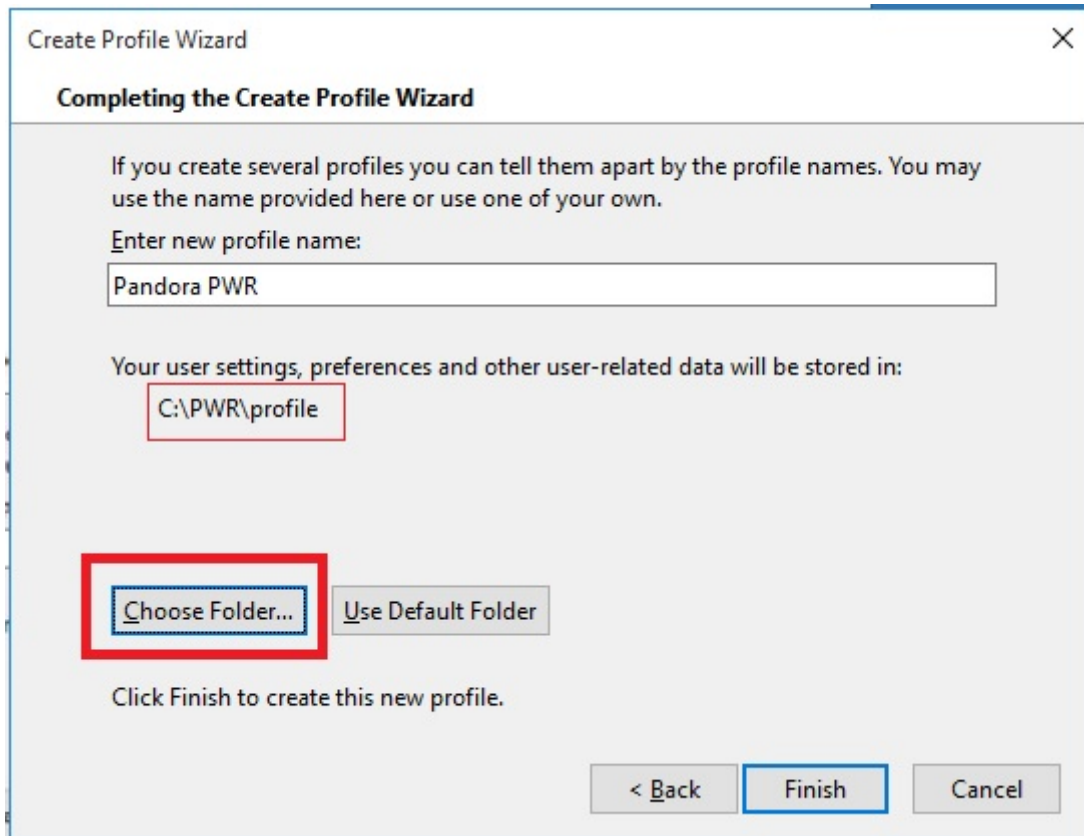
Haga la siguiente distribución de ficheros:

- Descompresión del fichero `PWR_Server.zip` a `C:\PWR\`.
- Exportar el perfil de Firefox® a `C:\PWR\profile`

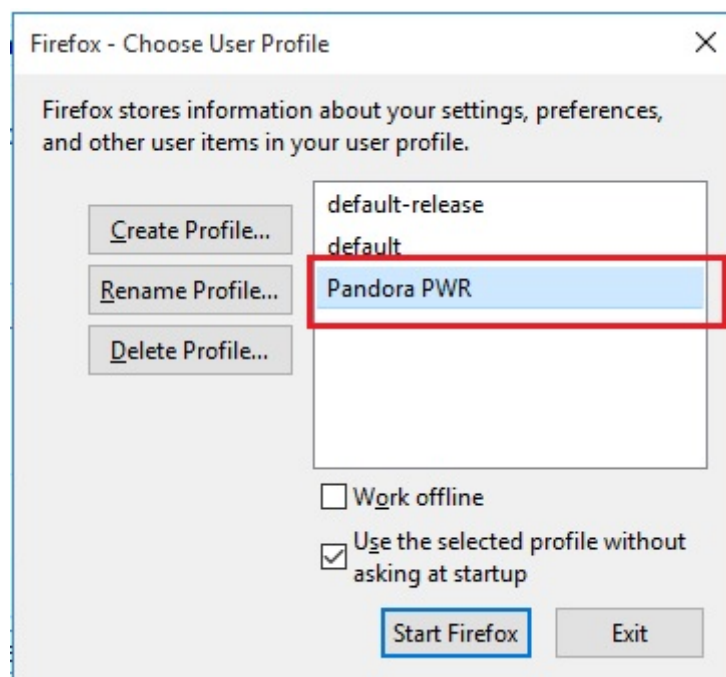
El uso de un perfil para realizar los chequeos no es obligatorio; no obstante, se recomienda su uso sobre todo en caso de utilizar *proxy* o cuando se requiera emplear autocompletado de contraseñas. Para crear un perfil de Firefox® siga los pasos de las siguientes:



Elección del directorio destino:



Debe iniciar Firefox® con el nuevo perfil para personalizar opciones como el proxy, mostrado de menú emergentes o *popups*, etc:



A continuación instale el servicio ejecutando el fichero BAT `service_installer.bat` que fue proporcionado. Para un correcto funcionamiento del `service_installer.bat` tendrá que modificar el contenido, colocando las rutas que existen en el fichero como las tenga instaladas en su máquina. Por ejemplo, en el caso de Java®, podrá servir únicamente como Java® si tiene correctamente configurado el PATH del mismo; si no, tendrá que dar todo el PATH dentro del

fichero. Finalmente, inicie el servicio:

```
net start PWRSRV
```

Desde este momento ya tendrá el servidor de Selenium corriendo en su máquina. No obstante, puede ejecutarlo a mano (deteniendo el servicio previamente) en caso de que quiera realizar tareas de depurado o *debugging*, mediante el siguiente comando:

```
java -jar C:\PWR\server\selenium-server-standalone-2.53.0.jar -  
firefoxProfileTemplate C:\PWR\profile -port 4444 -v
```

### Despliegue del servidor Selenium en sistemas GNU/Linux

Prerrequisitos:

- Instalar Java® 1.8 en la máquina que dará el servicio.
- Instalar Firefox® 47.0.1 (descargable en <https://ftp.mozilla.org/pub/firefox/releases/47.0.1/>).
- Preparar un perfil de Firefox® que se utilizará para probar las sesiones automatizadas (opcional): <http://toolsqa.com/selenium-webdriver/custom-Firefox-profile/>
- Instalar xorg-x11-server-xvfb.
- Instalar Java®.

El componente Selenium del demonio PWRD requiere Java® para funcionar, para instalarlo ejecute:

```
yum install java
```

Para que PWRD sea capaz de lanzar Firefox® en el servidor GNU/Linux, es necesario instalar, además, gtk2 y gtk3 para poder *virtualizar* un entorno gráfico:

```
yum install xorg-x11-server-Xvfb gtk2 gtk3
```

Si no está disponible en los repositorios puede encontrar los ficheros .rpm en los siguiente enlaces:

[ftp://rpmfind.net/linux/centos/6.6/os/x86\\_64/Packages/xorg-x11-server-Xvfb-1.15.0-22.el6.centos.x86\\_64.rpm](ftp://rpmfind.net/linux/centos/6.6/os/x86_64/Packages/xorg-x11-server-Xvfb-1.15.0-22.el6.centos.x86_64.rpm)

[ftp://rpmfind.net/linux/centos/7.4.1708/os/x86\\_64/Packages/gtk2-2.24.31-1.el7.x86\\_64.rpm](ftp://rpmfind.net/linux/centos/7.4.1708/os/x86_64/Packages/gtk2-2.24.31-1.el7.x86_64.rpm)

Para realizar la instalación de los paquetes .rpm de forma manual:

```
yum install xorg-x11-server-Xvfb-1.15.0-22.el6.centos.x86_64.rpm
yum install gtk2-2.24.31-1.el7.x86_64.rpm
```

Una vez instalados los prerequisites continuamos con la instalación de `install_pwr.sh`. Dicho instalador se encuentra por defecto en la carpeta `/usr/share/pandora_server/util/pwr/install_pwr.sh` y se ejecuta de la siguiente manera:

```
cd /usr/share/pandora_server/util/pwr/
./install_pwr.sh --install
```

Una vez instalado debe iniciar el servicio:

```
/etc/init.d/pwr start
```

Puede utilizar el siguiente *script* para iniciar el servidor Selenium:

```
#!/bin/sh
# Monitoring selenium process
if [[ "`ps aux |grep selenium` ]]; then
    exit
else
    if [[ "`ps aux |grep Xvfb` ]]; then
        Xvfb :99 -ac &
        export DISPLAY=:99
    fi
    export DISPLAY=:99
    java -jar /usr/share/pandora_server/util/pwr/selenium-server-standalone-2.53.1.jar &
fi
```

O bien manualmente con los siguientes comandos:

```
$ Xvfb :99 -ac &
-> Press Enter to continue
$ export DISPLAY=:99
$ java -jar /usr/share/pandora_server/util/pwr/selenium-server-standalone-2.53.1.jar -port 4444 &
```

También tiene la posibilidad de realizar la instalación personalizada con un usuario y un directorio diferente de la instalación por defecto.

### Modos de funcionamiento de PWRD

PWRD provee varios modos de funcionamiento:

Standalone

Modo estándar, inicia una única instancia de PWRD. Cada una de estas instancias irá asociada a un servidor de Pandora FMS.

## HUB

Modo concentrador. En este modo el servicio PWRD no evaluará las sesiones de navegación directamente, sino que deberán registrarse “nodos” para ejecutar las tareas. Es el modo *clúster* del servicio PWRD. Cada HUB irá asignado a un servidor de Pandora FMS.

### **PWRD en modo standalone**

PWRD en modo *standalone* iniciará el demonio y lo dejará listo para ejecutar las acciones indicadas por el usuario a través de WUX Server.



### WUXServer PWRD Standalone



PWRD



Pandora FMS Server  
(WUXServer enabled)

Para iniciar:

```
/etc/init.d/pwrld start
```

Para ver el estado:

```
/etc/init.d/pwrld status
```

Para detener:

```
/etc/init.d/pwrld stop
```

### **PWRD en modo HUB**

El modo concentrador (HUB) iniciará el demonio como un balanceador de carga. En este modo de trabajo, el sistema irá balanceando la carga entre todos los nodos que se hayan registrado en él, asignando la ejecución de las sesiones de navegación a los nodos en función de su carga de trabajo.

Puede ver en todo momento el estado del HUB accediendo a la consola de administración del mismo (sustituya <dir\_ip\_HUB> por la dirección IP del HUB):

```
http://<ip_addr_HUB>:4444/grid/console
```





**WUXServer - PWRD HUB**

Node 1



Node 2



Node 3



**PWRD**



**Pandora FMS Server  
(WUXServer enabled)**

Para iniciar:

```
/etc/init.d/pwrld start-hub
```

Para ver el estado:

```
/etc/init.d/pwrld status-hub
```

Para detener:

```
/etc/init.d/pwrld stop-hub
```

### **Agregar nodos de PWRD a un HUB**

Para agregar un nuevo nodo PWRD necesita:

- Un concentrador (PWRD en modo HUB).
- Los archivos de PWRD, en la misma o una máquina diferente.
- Conectividad TCP/4444 desde el equipo que aloja el nodo hacia el equipo que aloja el HUB.

En este modo de trabajo, el servicio procesará todas aquellas solicitudes encoladas desde el concentrador (HUB), devolviendo a este los resultados de las ejecuciones. Será el HUB quien “hable” de manera exclusiva con el WUX Server, siendo transparente para quien esté ejecutando las acciones del usuario.

Para iniciar y registrar en el HUB (sustituir hub por la dirección IP del servidor PWRD HUB):

```
/etc/init.d/pwrld start-node http://hub:4444/grid/register
```

Para ver el estado:

```
/etc/init.d/pwrld status-node
```

Para detener:

```
/etc/init.d/pwrld stop-node
```

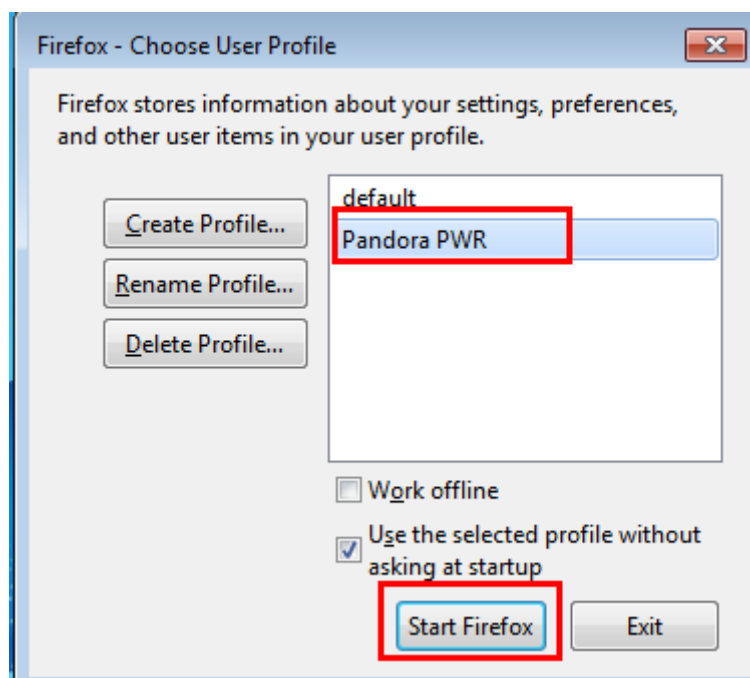
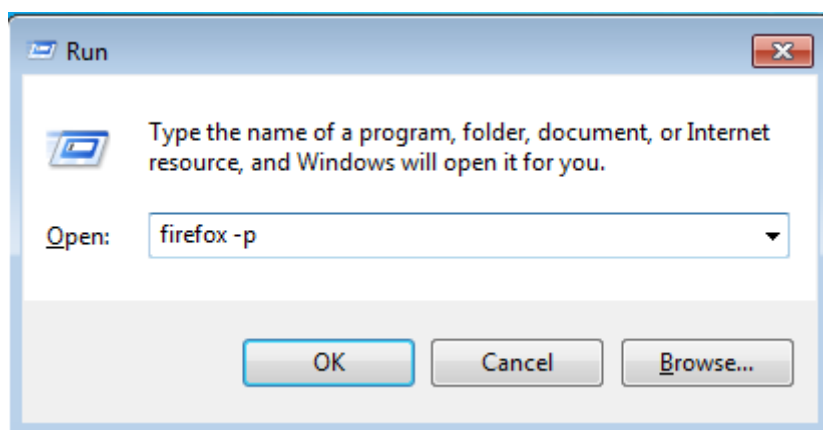
### **Carga de certificados para Firefox**

Es posible que algunos de los chequeos configurados se ejecuten contra páginas web que usen certificados autofirmados o firmados por una CA no incluida en las aceptadas por defecto en Firefox®, por lo que en esos casos será necesario cargar el certificado en cuestión en el perfil de Firefox® que se esté usando.

Para hacer esto, lo más sencillo es iniciar el navegador en un entorno gráfico, acceder a la URL y agregar el certificado SSL. A continuación se explica cómo hacerlo tanto en MS Windows® como en GNU/Linux®:

Con PWRD desplegado en sistemas Windows

En este caso, ya que cuenta de una vez con un entorno gráfico, solo tiene que iniciar el navegador Firefox® con el perfil que esté usando para los chequeos:



Una vez iniciado, acceda a la URL con el certificado que quiera cargar y añádalo como excepción para el navegador:



## Your connection is not secure

The owner of artica.es has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

[Learn more...](#)

Report errors like this to help Mozilla identify and block malicious sites

Go Back

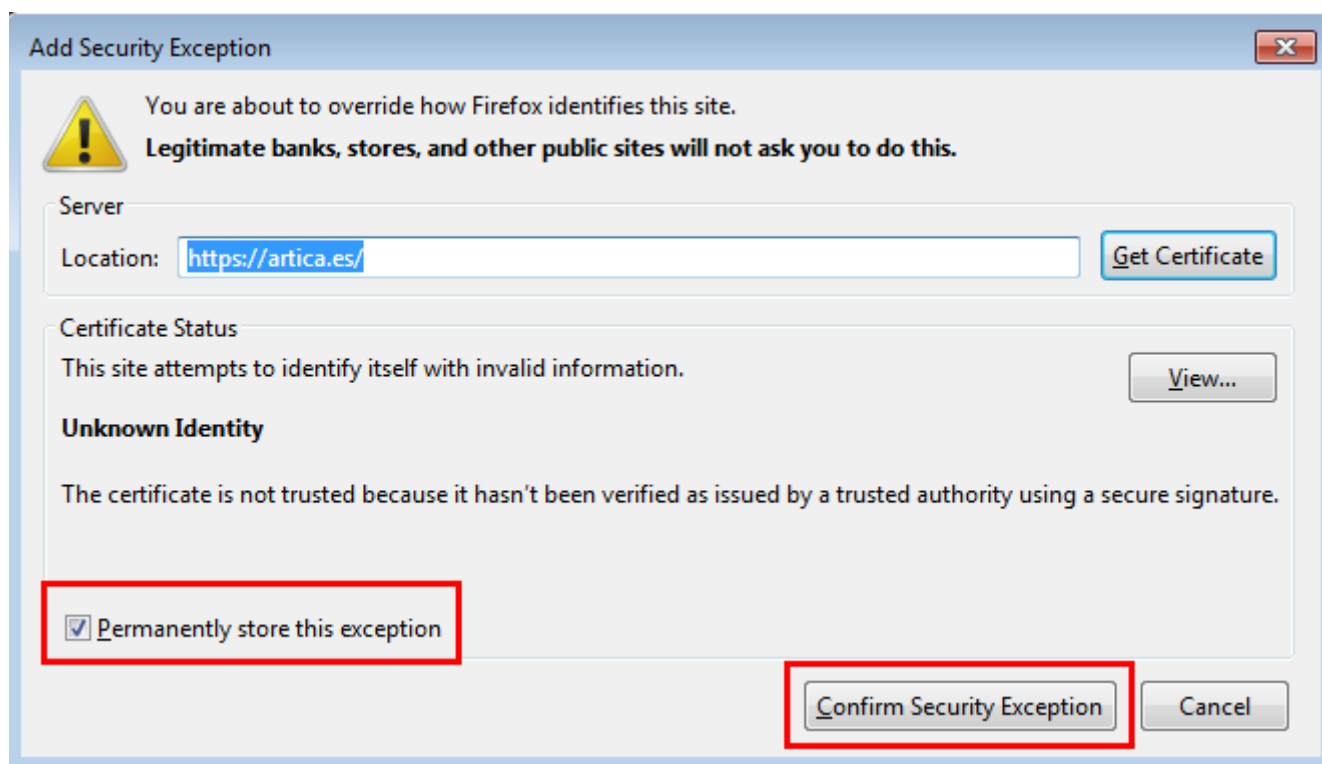
Advanced

artica.es uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is unknown.  
The server might not be sending the appropriate intermediate certificates.  
An additional root certificate may need to be imported.

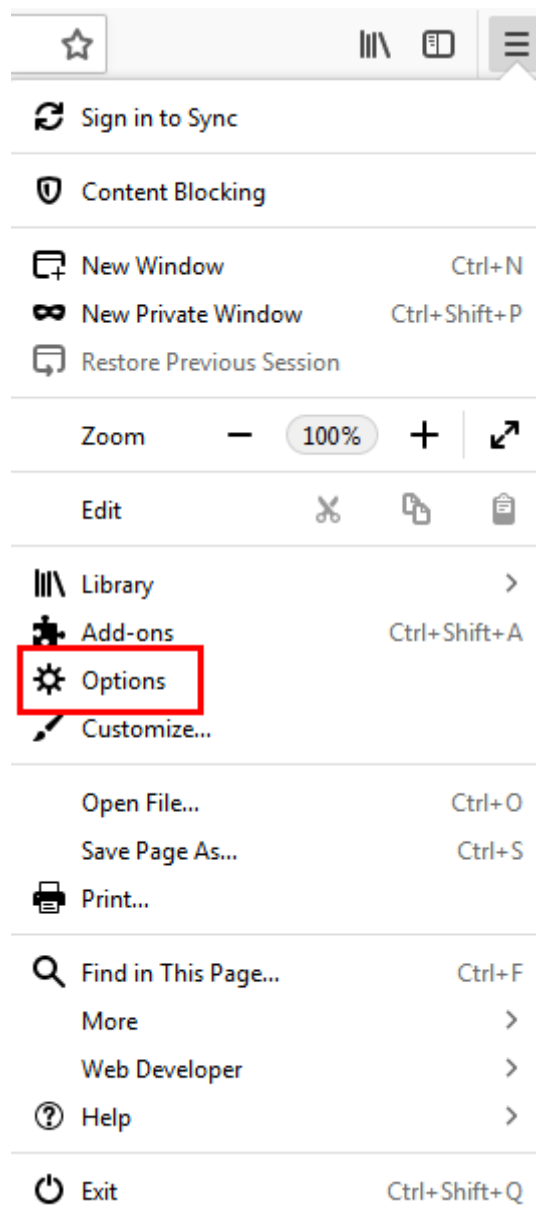
Error code: [SEC\\_ERROR\\_UNKNOWN\\_ISSUER](#)

Add Exception...



Otra posibilidad, si se quiere aceptar cualquier certificado SSL y considerando que es un entorno de pruebas y/o monitorización, sería acceder a las opciones de Firefox®, a la pestaña de “Privacidad & Seguridad”, y desmarcar el campo “Consultar a los servidores *respondedores* OCSP

para confirmar la validez actual de los certificados”:



Find in Options

- General
- Home
- Search
- Privacy & Security**
- Firefox Account

### Firefox Data Collection and Use

We strive to provide you with choices and collect only what we need to provide and improve Firefox for everyone. We always ask permission before receiving personal information.

[Privacy Notice](#)

- Allow Firefox to send technical and interaction data to Mozilla [Learn more](#)
- Allow Firefox to install and run studies [View Firefox studies](#)
- Allow Firefox to send backlogged crash reports on your behalf [Learn more](#)

---

### Security

#### Deceptive Content and Dangerous Software Protection

- Block dangerous and deceptive content [Learn more](#)
  - Block dangerous downloads
  - Warn you about unwanted and uncommon software

#### Certificates

When a server requests your personal certificate

- Select one automatically
- Ask you every time
- Query OCSP responder servers to confirm the current validity of certificates [View Certificates...](#)

[Security Devices...](#)

Firefox Support

Con PWRD desplegado en sistemas Linux

Por lo general, cuando se instala un servidor GNU/Linux®, no se le incluye un entorno de escritorio gráfico, por lo que para poder seguir los mismos pasos que en el caso anterior tiene que hacer una redirección de las X gráficas a un equipo donde sí tenga un entorno de escritorio gráfico. Esto se hace de forma distinta si redirigimos las X a un escritorio gráfico en GNU/Linux o MS Windows®.

#### Redirigiendo X11 a un escritorio en Linux

Establezca una conexión SSH al servidor PWRD con el parámetro “-X”:

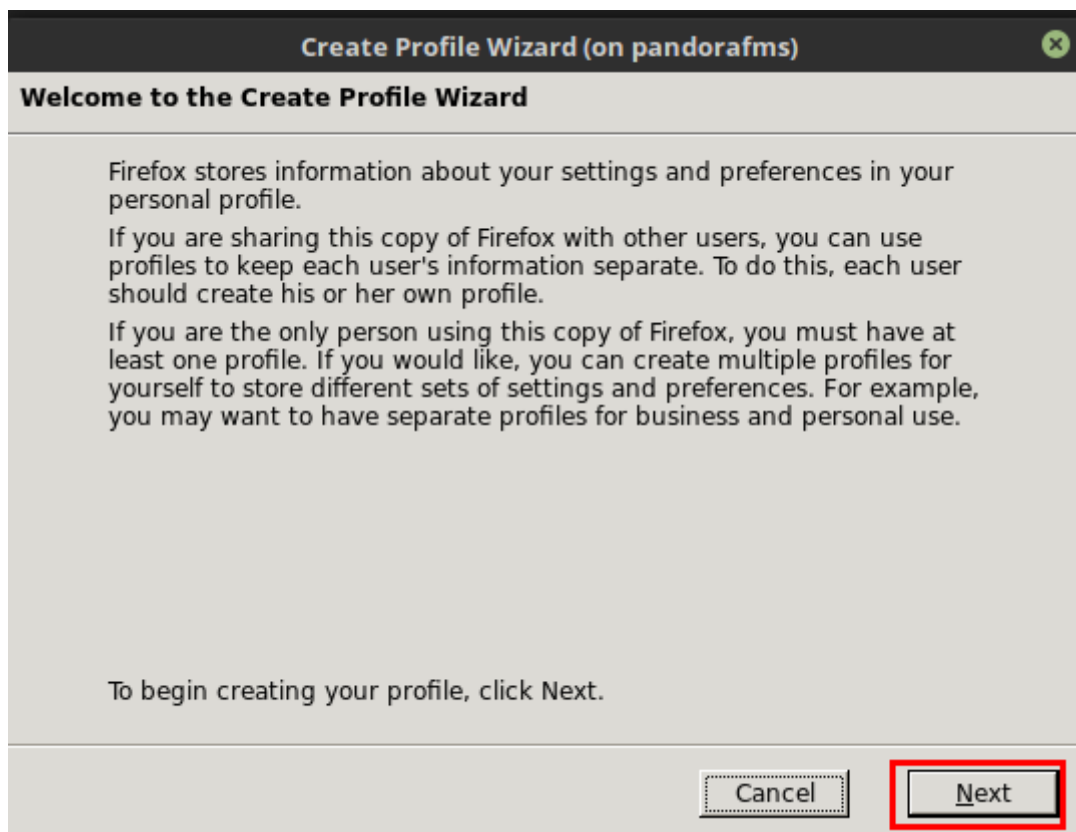
```
ssh -X user@pwrд_ip_address
```

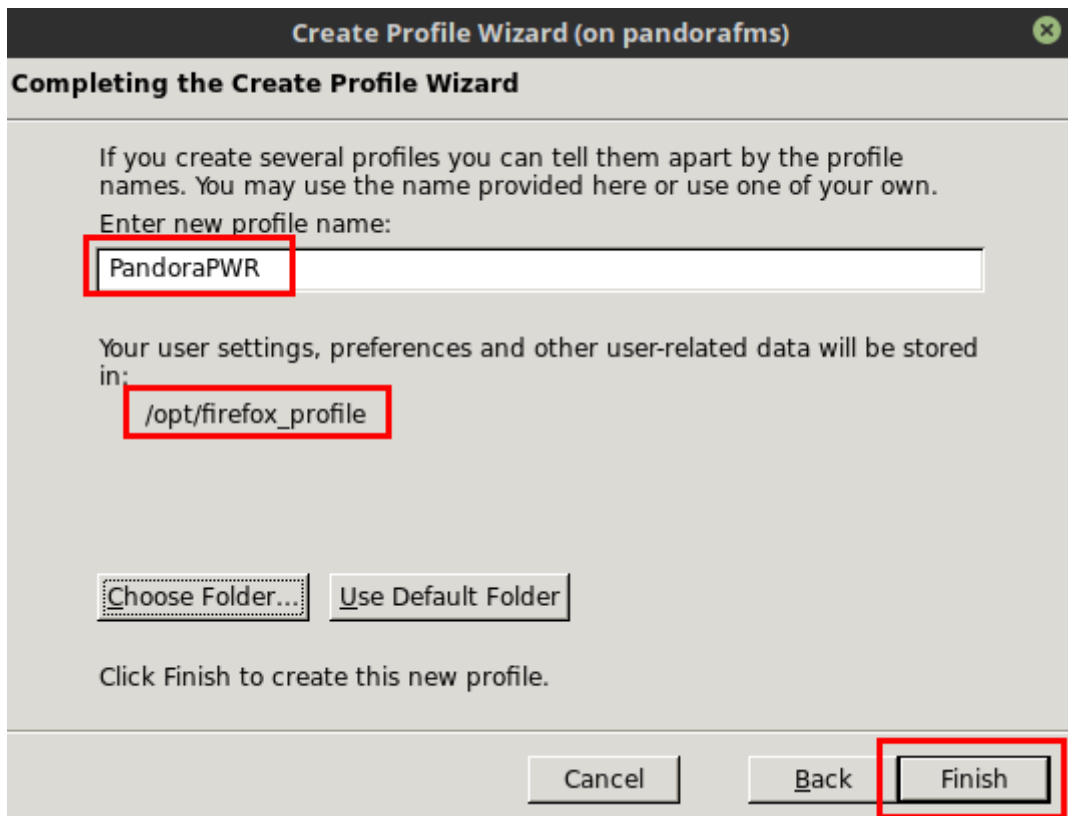
Hecho esto, al lanzar el navegador Firefox, podrá ver como se muestra en el escritorio:

```
firefox -p
```

En una instalación por defecto solo veremos el perfil “default”, por lo que es recomendable hacer

tal como hizo en la instalación en MS Windows® y crear un nuevo perfil a utilizar:

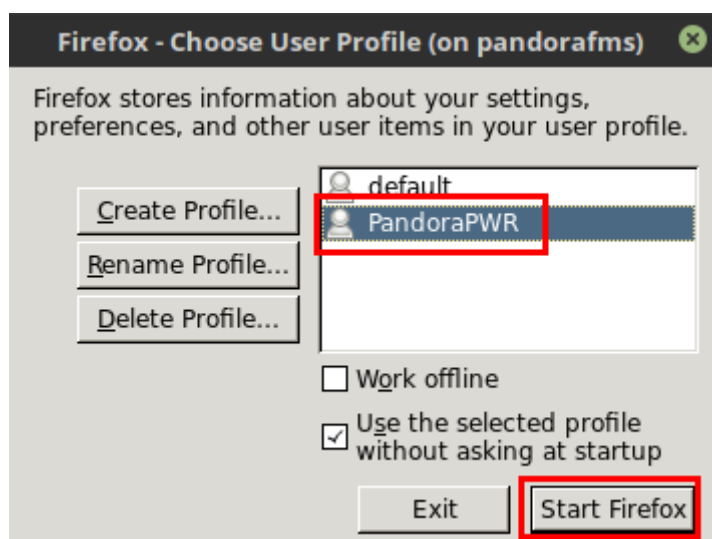




En caso de guardar el perfil en una ruta distinta a /opt/firefox\_profile es necesario editar el *script* de arranque del PWRD /etc/init.d/pwr para indicar la ruta al nuevo perfil:

```
PWROPTS=" -firefoxProfileTemplate /path/to/profile/folder"
```

Con el perfil creado, puede iniciar el navegador:



Una vez iniciado, acceda a la URL con el certificado que se quiera cargar y añadir como excepción para el navegador:





## This Connection is Untrusted

You have asked Firefox to connect securely to **artica.es**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

#### ▶ Technical Details

##### ▼ I Understand the Risks

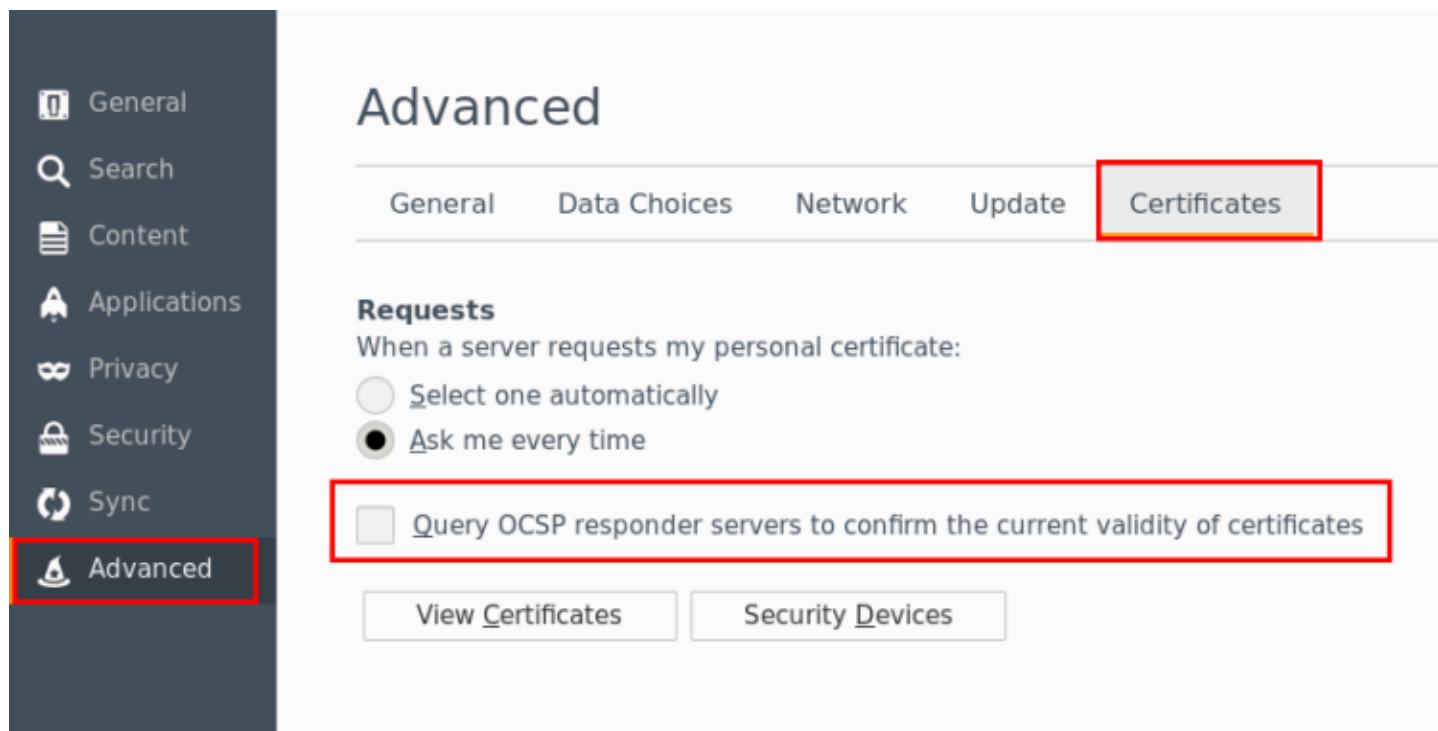
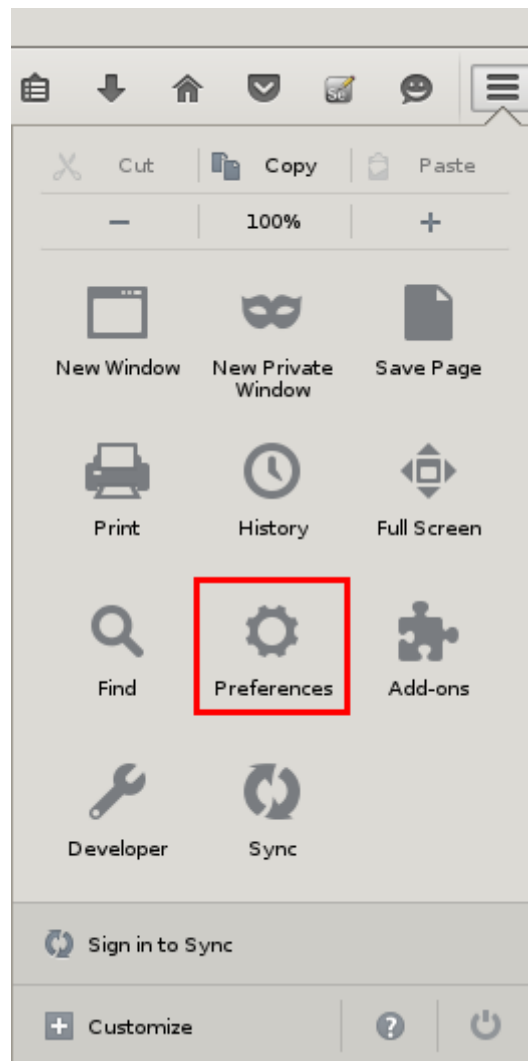
If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Add Exception...



Otra posibilidad, si se quiere aceptar cualquier certificado SSL y considerando que es un entorno de prueba y/o monitorización, sería acceder a las opciones de Firefox, a la pestaña de "Avanzado > Certificados", y desmarcar el campo "Consultar a los servidores *respondedores* OCSP para confirmar la validez actual de los certificados":

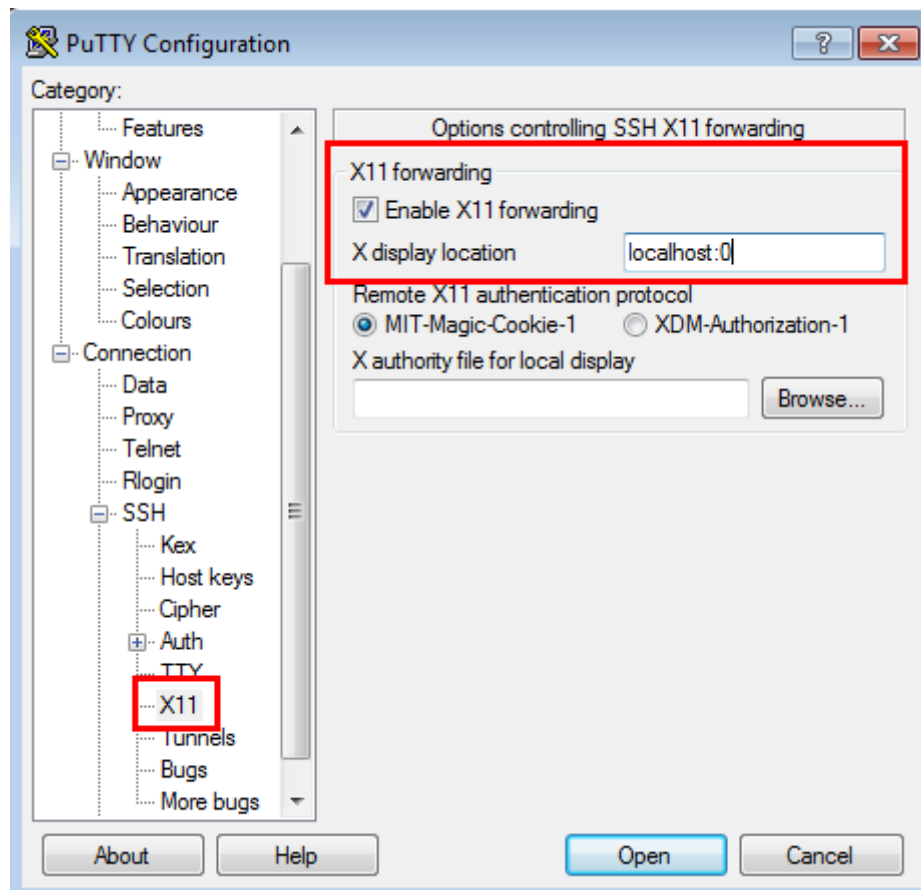


### Redirigiendo X11 a un escritorio en Windows

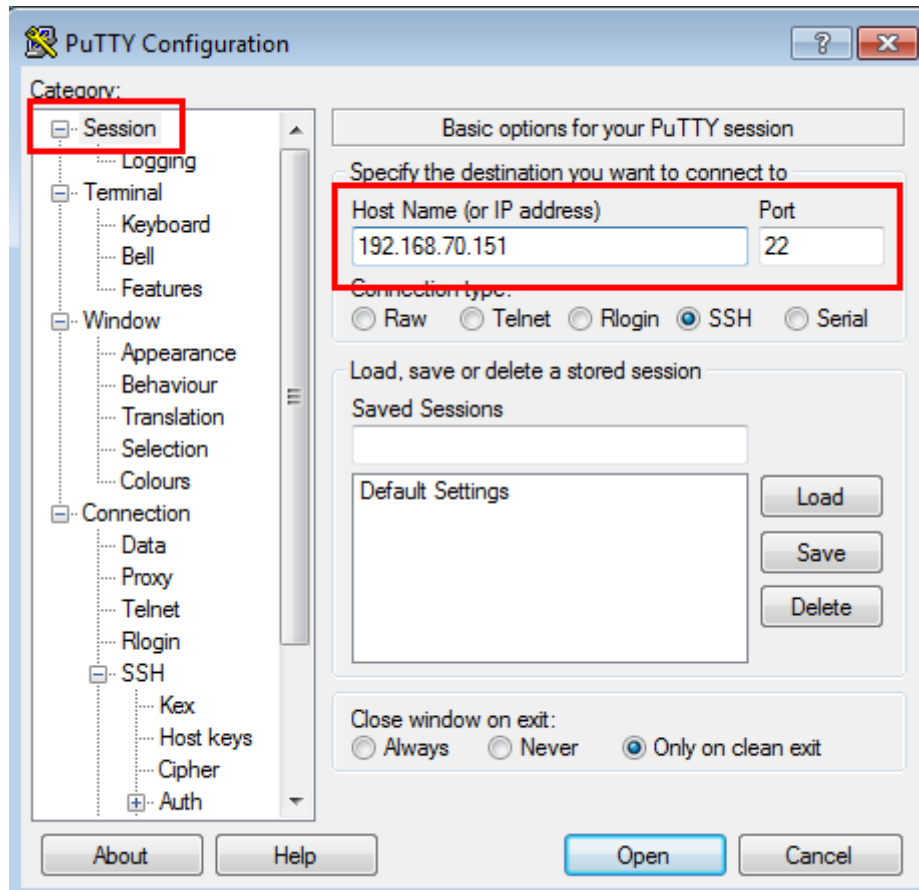
En el caso de Windows, primero tiene que instalar un servidor de X para poder hacer la

redirección, como por ejemplo Xming. Una vez instalado e iniciado, puede hacer la redirección de las X.

Usando el cliente SSH Putty, vaya al apartado “Connection > SSH > X11” antes de hacer la conexión, y asegúrese de marcar la opción “Enable X11 forwarding” así como de rellenar el campo “X display location” como “localhost:0”:



Después vuelva al apartado “Session” y establezca la conexión:

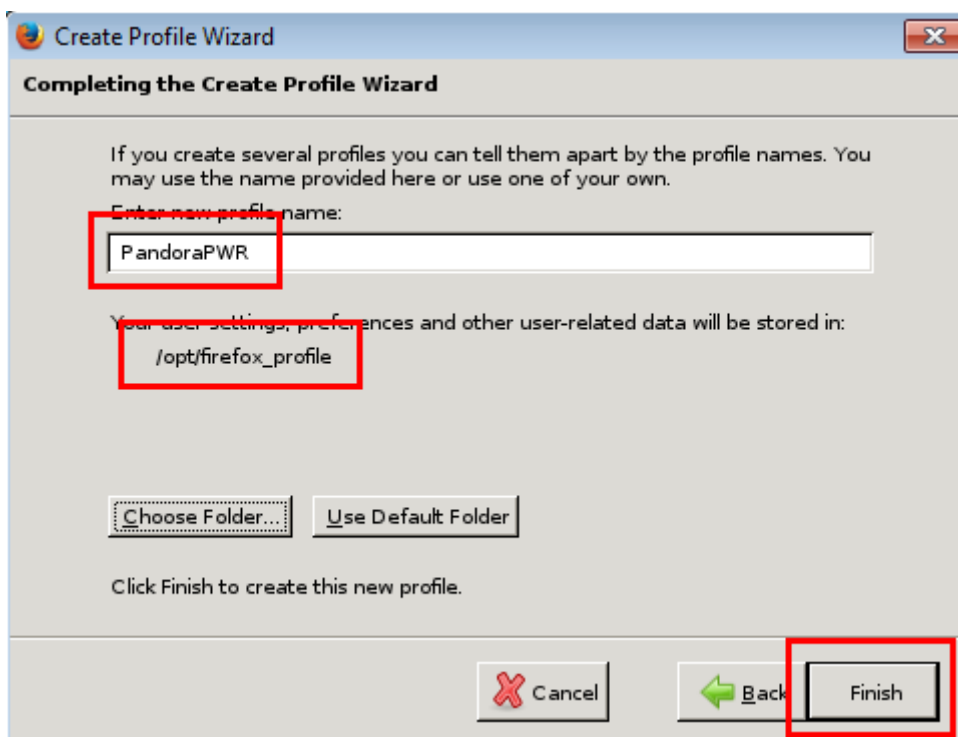
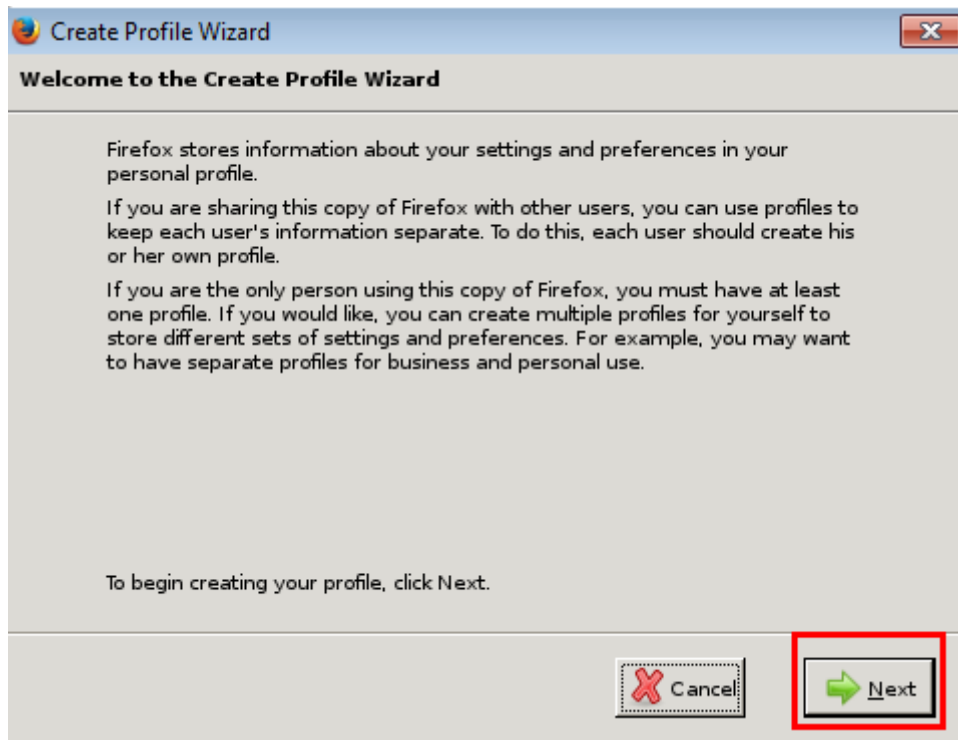


Una vez establecida la conexión, si lanza el navegador Firefox® en el servidor PWRD, podremos visualizarlo en el escritorio de MS Windows®.

```
firefox -p
```

En una instalación por defecto solo veremos el perfil "default", por lo que sería recomendable hacer como en la instalación en Windows® y crear un nuevo perfil el cual utilizar:

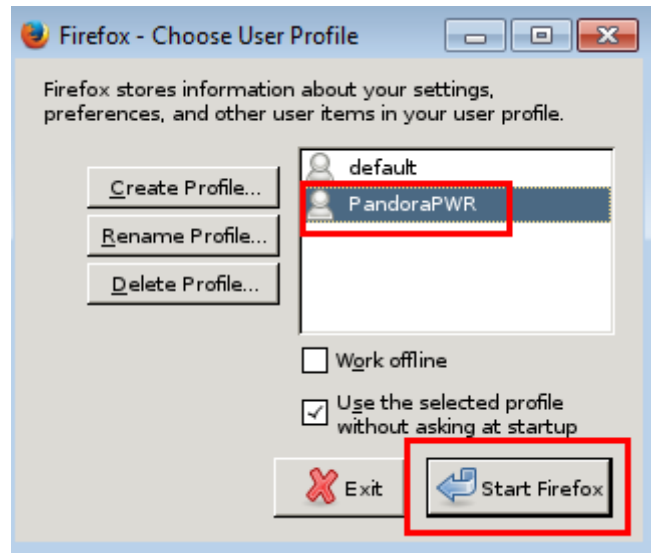




En caso de guardar el perfil en una ruta distinta a /opt/firefox\_profile, será necesario editar el *script* de arranque del PWRD /etc/init.d/pwr

```
PWROPTS =" -firefoxProfileTemplate /path/to/profile/folder"
```

Con el perfil ya creado, ya puede iniciar el navegador web:



Una vez iniciado, acceda a la URL con el certificado que quiera cargar y añádalo como excepción para el navegador web:



## This Connection is Untrusted

You have asked Firefox to connect securely to **artica.es**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

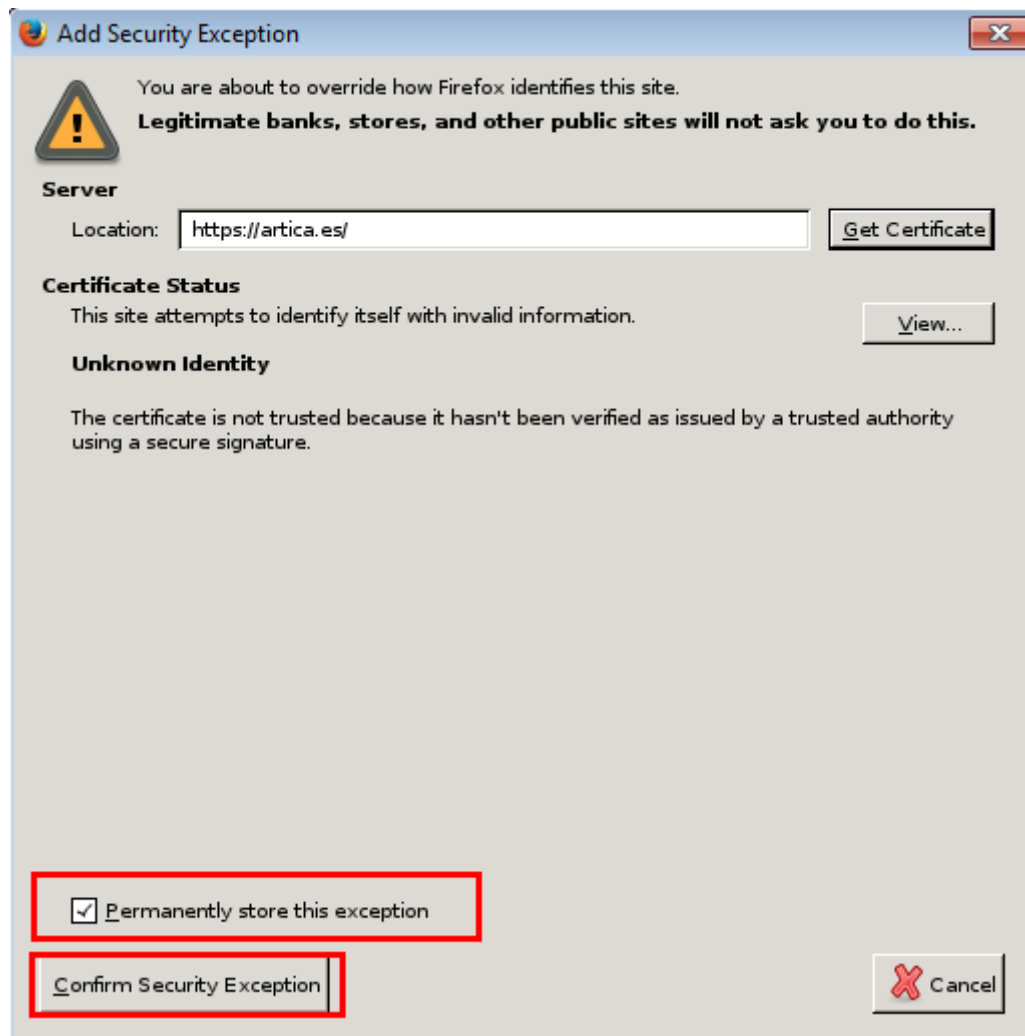
#### ► Technical Details

#### ▼ I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

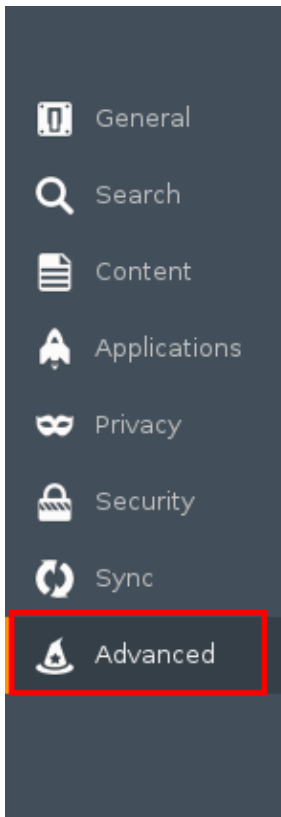
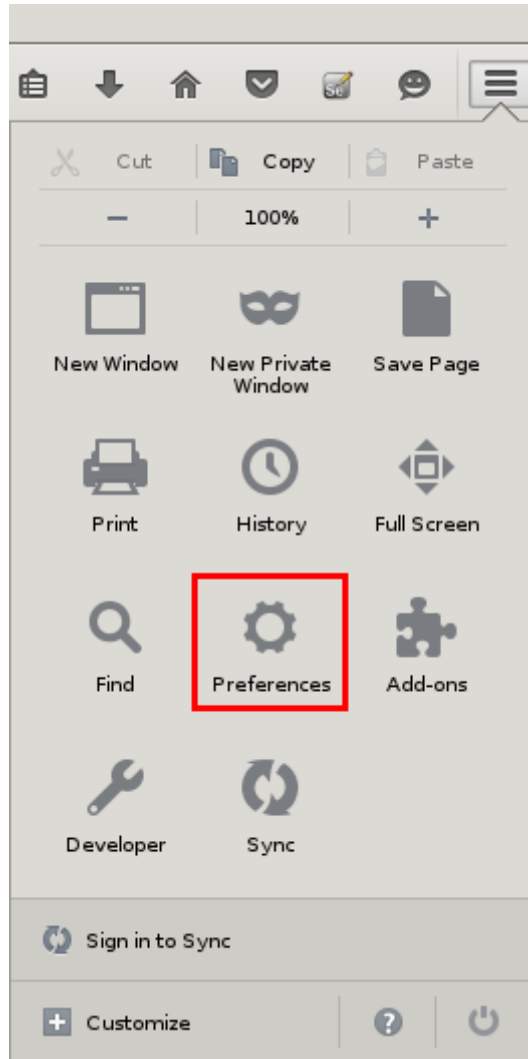
Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

[Add Exception...](#)

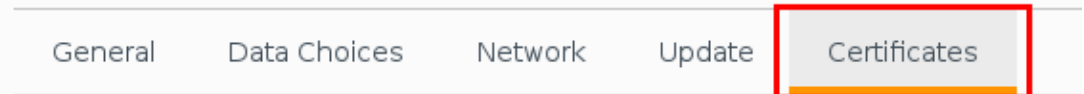


Otra posibilidad, si quiere aceptar cualquier certificado SSL, sería ir a las opciones de Firefox, ir a "Privacidad y Seguridad" y luego "Certificados", y desmarcar el campo "Consultar a los servidores respondedores OCSP para confirmar la validez actual de los certificados":





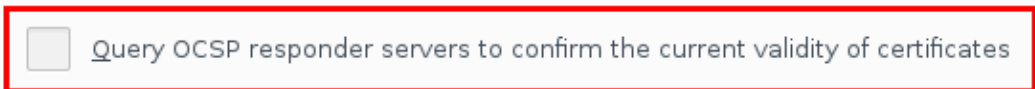
## Advanced



### Requests

When a server requests my personal certificate:

- Select one automatically
- Ask me every time



[View Certificates](#)

[Security DeVICES](#)

## Configuración del servidor de Pandora FMS

Para la utilización del modo centralizado (WUX) es necesario aplicar la siguiente configuración al servidor de Pandora FMS.

Independientemente del modo elegido, una vez iniciado puede empezar a asignar ejecuciones de sus sesiones de navegación, agregando los parámetros de configuración de WUX Server al fichero de configuración del servidor de Pandora FMS.

Suponiendo que ha desplegado el servidor PWRD en el mismo servidor en que corre el servidor Pandora FMS, agregaría la siguiente configuración en `/etc/pandora/pandora_server.conf`:

```
wuxserver 1
wux_host 127.0.0.1
wux_port 4444
wux_timeout 30
```

El parámetro `wux_timeout` establece el tiempo máximo de las transacciones a 30 segundos: de ser necesario ajuste este valor a su entorno particular.

### Gestión de hilos en PFMS

En caso de utilizar el PWRD en modo *hub*:

- La gestión de los hilos del `wuxserver` se hace automáticamente al iniciar el servicio de `pandora_server`.
- Se realiza teniendo en cuenta el número de nodos de un navegador MENOR que haya en el *hub* de Selenium. Por ejemplo:
  - Si en el *hub* hay configurados 2 nodos de Firefox y 2 de Chrome, el número de hilos del `wuxserver` será de 2.
  - Si en el *hub* hay configurados 1 nodo de Firefox y 4 de Chrome, el número de hilos será de 1.
  - Si en el *hub* hay configurados 6 nodos de Firefox, el número de hilos será de 6.

Tenga en cuenta que cada hilo indica las sesiones que pueden enviarse de manera simultánea desde el `wuxserver` al *hub* de Selenium.

### Grabador de sesiones (PWR)

La nueva versión de Selenium IDE es compatible a partir de la versión 745 de Pandora FMS (Selenium 3). Las transacciones grabadas en Selenium 2 no se verán afectadas.

Antes de monitorizar una experiencia de usuario debe hacer la grabación. Dependiendo del tipo de tecnología que haya elegido utilizará un sistema de grabación u otro.

Las sesiones grabadas con Selenium 2 sólo podrán realizarse con Firefox®.

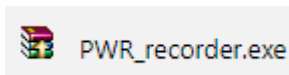
Para realizar la grabación de una navegación con PWR es necesario el grabador PWRD disponible en la biblioteca de módulos:

<https://pandorafms.com/library/pandora-ux-and-wux-pwr-recorder/>

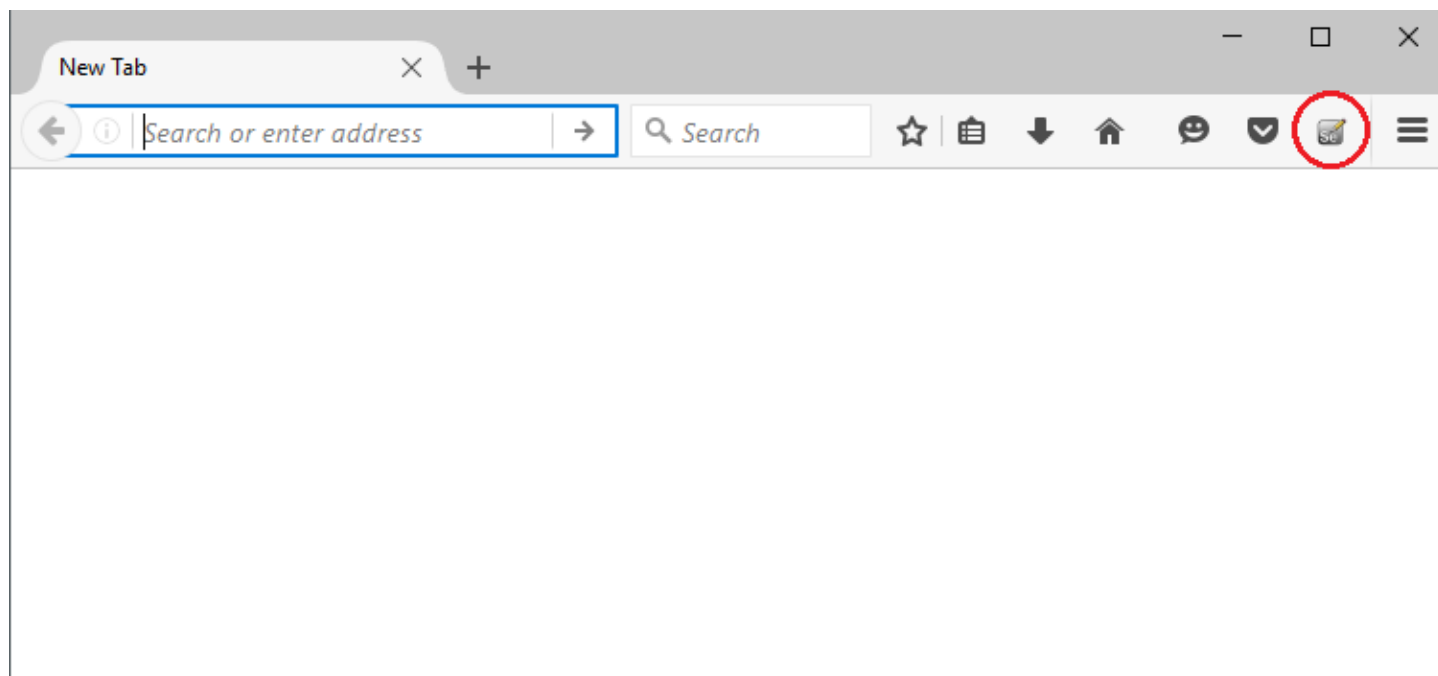
Contiene:

- Navegador web **Firefox versión 47.0.1**.
- Extensión **Selenium IDE**.

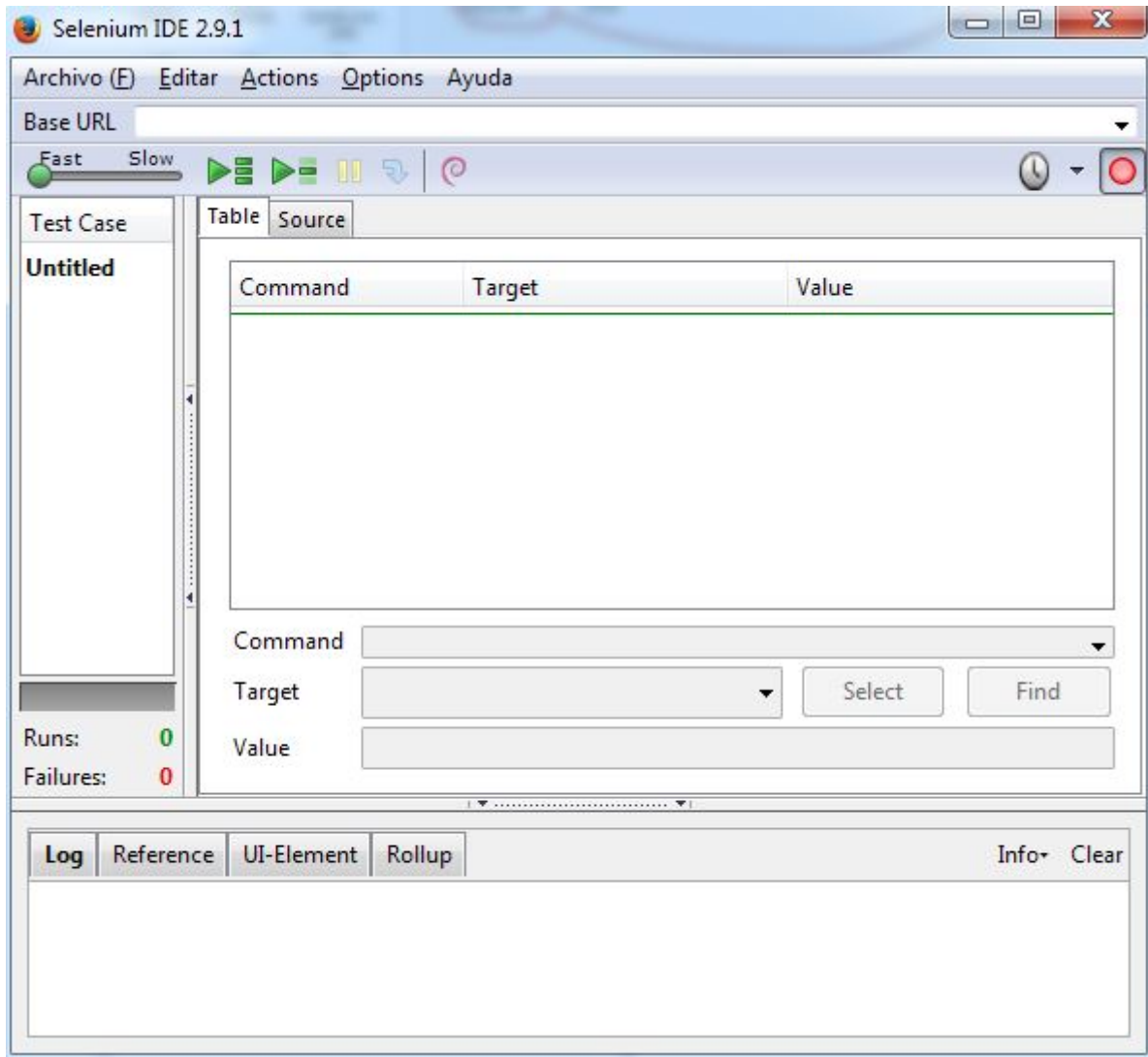
Se debe iniciar el grabador `PWR_recorder.exe`



Automáticamente iniciará Firefox con el entorno preparado para grabar sesiones PWR:



Tras acceder al IDE de Selenium se puede iniciar así la grabación de experiencia de usuario:

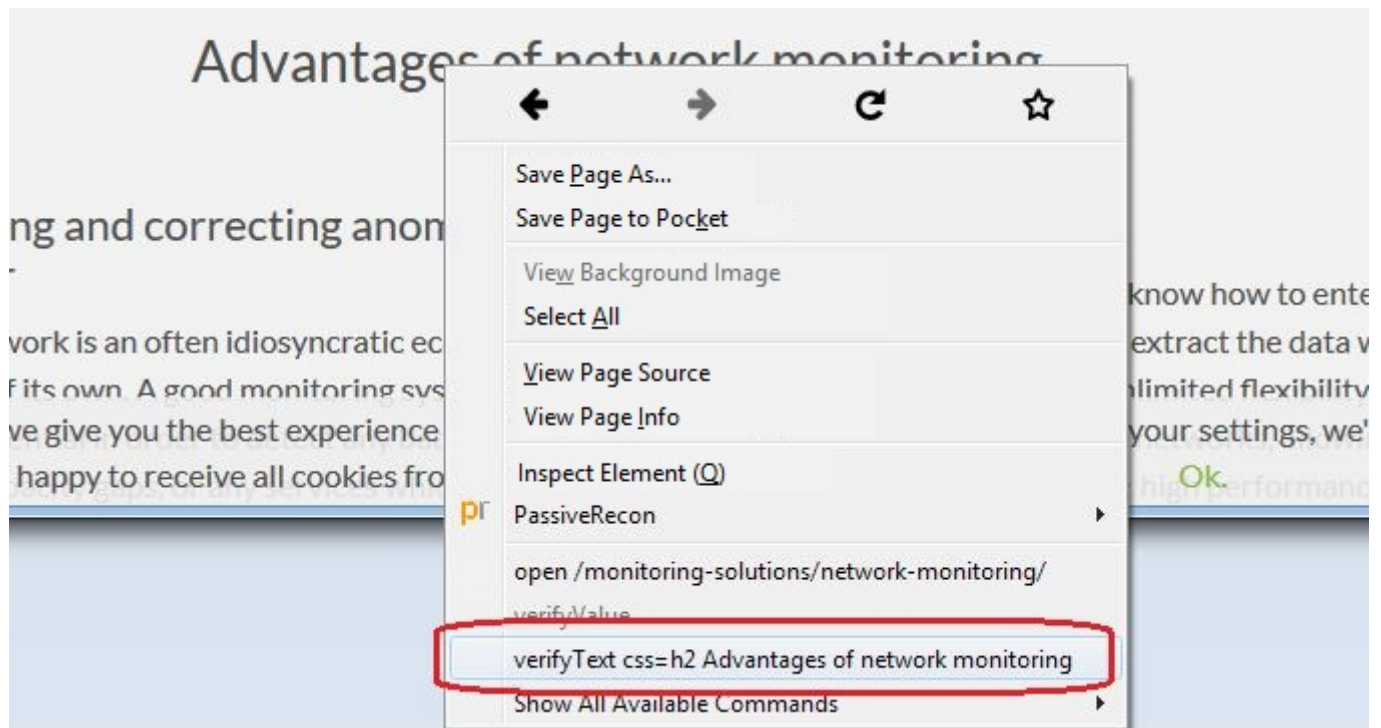


Navegue por el sitio web a monitorizar; las diferentes acciones de cada paso hecho irán apareciendo en el grabador.

Para detener la grabación utilice el siguiente botón, situado en la parte superior derecha del grabador:



Una vez completadas las acciones, realice las comprobaciones sobre la página; por ejemplo, verificar la existencia de un texto determinado para asegurar que la página cargada es la correcta. Para ello, hacer clic secundario sobre una sección de texto en la ventana del navegador mientras continúa la grabación, y seleccionar la opción *verifyText*:



Aparecerá un nuevo paso en el grabador indicando la acción de comprobación de texto indicada:

The screenshot shows the test suite recorder interface. It features a table with columns for Command, Target, and Value. The table contains the following entries:

Command	Target	Value
open	/monitoring-software/	
clickAndWait	xpath=//img[@alt='Pandora FMS']][2]	
clickAndWait	xpath=//a[contains(text(),'Network Monitoring')]]][2]	
verifyText	css=h2	Advantages of network monitoring

Below the table, there is a configuration area for the selected command:

Command: verifyText  
 Target: css=h2  
 Value: Advantages of network monitoring

Buttons for 'Select' and 'Find' are visible next to the Target field.

Puede reproducir la secuencia completa mediante el botón Play entre test suite y comprobar que finaliza correctamente:

Archivo (F) Editar Actions Options Ayuda

Base URL <https://pandorafms.com/>

Fast Slow

Test Case

Untitled 5 \*

Command	Target	Value
open	/monitoring-software/	
clickAndWait	xpath=//img[@alt='Pandora FMS']	[2]
clickAndWait	xpath=//a[contains(text(),'Network Monitoring')]	[2]
verifyText	css=h2	Advantages of network monitoring

Command

Target

Value

Select Find

Runs: 1

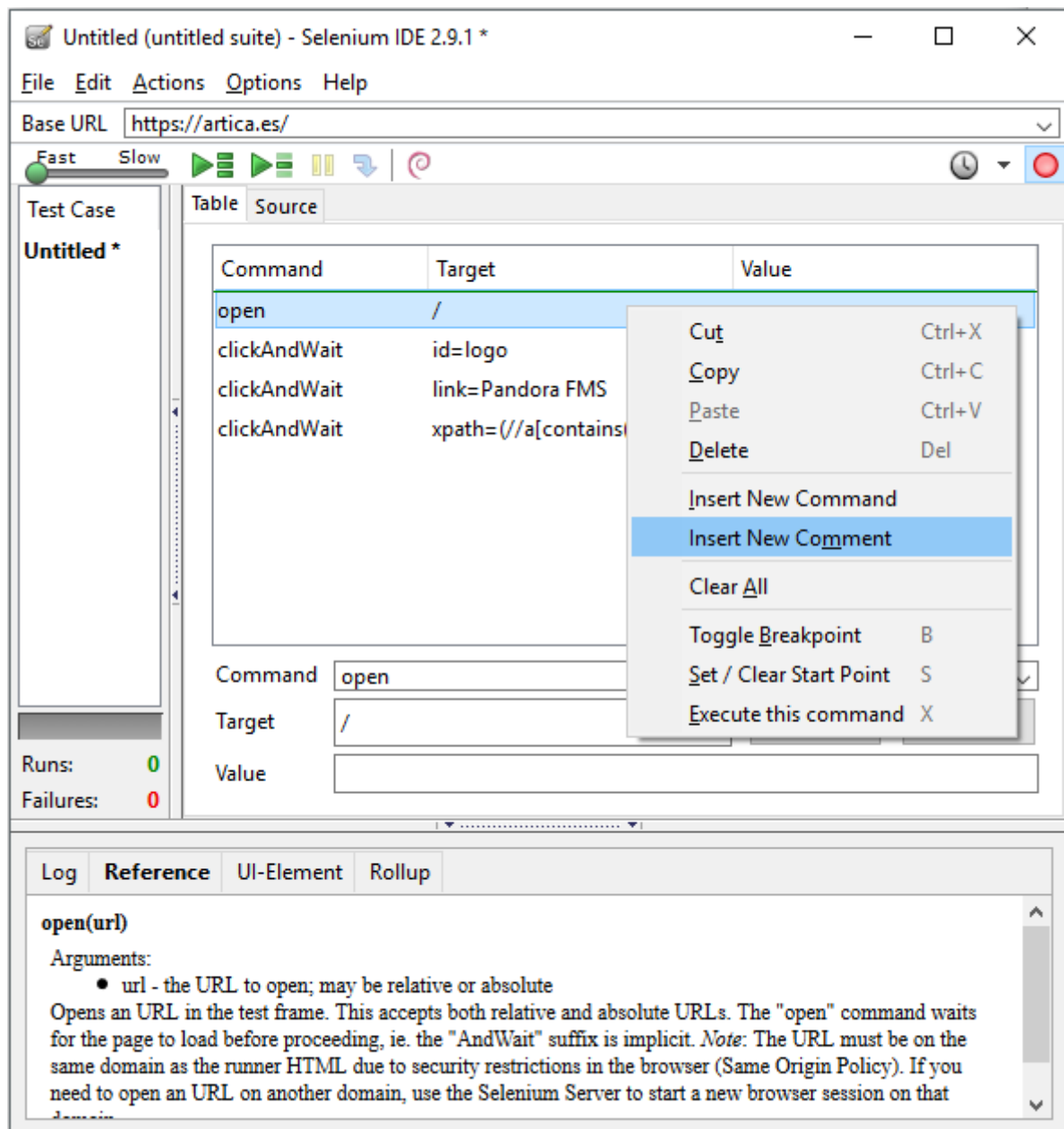
Failures: 0

Una vez verificada la validez de la secuencia de navegación, guarde (Archive → Save Test Case) para ejecutarla posteriormente con Pandora FMS UX. El fichero resultante será un documento HTML que Pandora FMS UX interpretará.

#### Grabar una sesión transaccional para PFMS WUX

Pandora FMS WUX permite dividir la monitorización de la navegación de un sitio web en múltiples módulos, que representarán cada uno de los pasos realizados.

Para insertar un nuevo punto de control y generar los módulos de fase (hasta ese punto) haga clic secundario en el punto donde desea identificar el comienzo de fase.



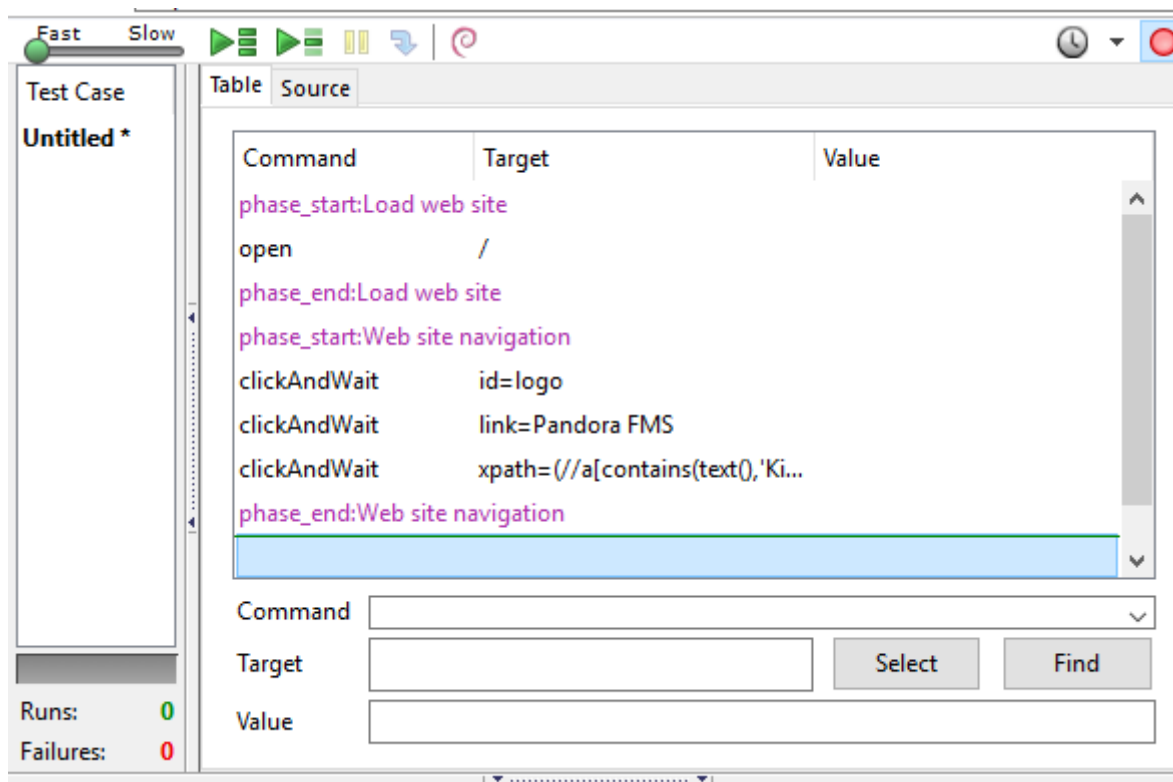
Como comentario coloque el siguiente texto:

```
phase_start:<phase_name>
```

La fase englobará el tiempo y resultado de todos los comandos que se encuentren hasta el siguiente comentario:

```
phase_end:<phase_name>
```

Todos los comandos que se ejecuten entre una etiqueta `phase_start` y `phase_end` se englobarán dentro de esa fase.



Se debe tener en cuenta que las grabaciones realizadas en Selenium 3 pueden no funcionar correctamente.

## Ejecución de transacciones web

### Ejecución estándar

Para lanzar sesiones pregrabadas de PWR llame al ejecutable `pandora_ux_x64.exe`, el cual puede descargar del siguiente enlace:

<https://pandorafms.com/library/user-experience-pandora-ux>

Indique que el modo de trabajo es PWR, y el archivo que contiene las directrices de la sesión. En la ruta indicada en el parámetro `-folder` se guardarán las capturas de pantalla a mostrar en caso de error (parámetro opcional). También se puede introducir el número de reintentos consecutivos en caso de fallo (parámetro opcional). Su ejecución en Windows es:

```
pandora_ux_x64.exe -exe PWR -script C:\tests\std.html -retries 3
```

Se devolverán los siguientes módulos:

- `UX_Status_<nombre_proyecto>` si la secuencia ha tenido éxito o ha fallado.
- `UX_Time_<nombre_proyecto>` tiempo empleado en completar la secuencia.
- `UX_Snapshot_<nombre_proyecto>` captura de pantalla con el último punto antes del error, si lo hubiere.



Ejemplo de salida con ejecución correcta:

```
<module>
  <name><![CDATA[UX_Status_std.html]]></name>
  <type>generic_proc</type>
  <data><![CDATA[1]]></data>
  <description><![CDATA[Test OK]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
</module>
<module>
  <name><![CDATA[UX_Time_std.html]]></name>
  <type>generic_data</type>
  <data><![CDATA[16.317]]></data>
  <description><![CDATA[Test OK]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_std.html</module_parent>
</module>
```

Ejemplo de salida con ejecución incorrecta:

```
<module>
  <name><![CDATA[UX_Status_std.html]]></name>
  <type>generic_proc</type>
  <data><![CDATA[0]]></data>
  <description><![CDATA[Failed to execute verifyText]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
</module>
<module>
  <name><![CDATA[UX_Time_std.html]]></name>
  <type>generic_data</type>
  <data><![CDATA[15.463]]></data>
  <description><![CDATA[Failed to execute verifyText]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_std.html</module_parent>
</module>

<module>
  <name><![CDATA[UX_Snapshot_std.html]]></name>
  <type>async_string</type>
  <data><![CDATA[data:image/png;base64,
iVBORw0KGgoAAAANSUhEUgAA...JRU5ErkKg==]]></data>
  <description><![CDATA[Image (last error)]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_std.html</module_parent>
</module>
```

Si todo ha resultado correcto puede añadir la línea de ejecución al agente de Pandora FMS instalado en la máquina que haya preparado para ejecutar los chequeos. La línea a añadir al fichero de configuración del agente tendrá un *aspecto similar* a este (en una sola línea):

En Windows®:

```
module_plugin C:\Users\pfms\Documents\Product\UX-Trans\UX\pandora_ux_x64.exe -
exe PWR -script C:\Users\pfms\Documents\Product\UX-Trans\PWR\sesion.html -folder
<screenshots path>
```

En GNU/Linux:

```
module_plugin /usr/share/pandora_server/tool/pwr/firefox/pandora_ux.64 -exe PWR
-script /usr/share/pandora_server/tool/pwr/firefox/TestUX.html -pwr_port 4444
```

Una vez que se encuentre reportando la información al servidor de Pandora FMS verá cómo aparecen los módulos correspondientes. Puede habilitar la vista en *modo jerárquico* para ver la relación entre ellos:

	UX_Status_sesion.html	Test OK	N/A - N/A	1			45 seconds
	L UX_Snapshot_sesion.html	Image (last error)	N/A - N/A	No image ava			20 hours
	L UX_Time_sesion.html	Test OK	N/A - N/A	4.7			45 seconds

Como se ha indicado anteriormente, verá el resultado de la ejecución (si ha tenido éxito o no) en el módulo `UX_Status_sesion.html`, el tiempo empleado (en segundos) en el módulo `UX_Time_sesion.html`, y otro con la captura de pantalla del último error en `UX_Snapshot_sesion.html`, que en este caso se encuentra vacío. Será sobre estos módulos sobre los que puede crear alertas y mostrar las gráficas individuales de cada módulo.

### Ejecución basada en fases

Si tiene una grabación transaccional con Pandora FMS UX PWR, como se indicó en los apartados anteriores, será el propio sistema quien genere los módulos necesarios para identificar cada una de las fases indicadas, por lo que la ejecución no variará con respecto al caso anterior; únicamente indique el fichero html correspondiente, que en este caso contendrá las diferentes fases. Ejemplo de ejecución en Windows:

```
pandora_ux_x64 -exe PWR -script C:\tests\std.html --folder <screenshots path>
```

Se devolverán los siguientes módulos por fase:

- UX\_Time\_nombre\_proyecto.fase\_orden

- UX\_Status\_nombre\_proyecto.fase\_orden

Si hay alguna fase con error, se creará también el módulo siguiente:

- UX\_Snapshot\_nombre\_proyecto.fase\_orden

Mostrará una imagen de la web en el momento del error.

También se devolverán los módulos de resumen globales identificados con los siguientes nombres:

- UX\_Global\_Time\_nombre\_proyecto
- UX\_Global\_Status\_nombre\_proyecto
- UX\_Global\_Snapshot\_nombre\_proyecto

Y se mostrará una imagen de la web en el momento del error.

La línea de ejecución del agente sería la misma que en el caso anterior, pero con el fichero html que contiene las fases estipuladas.

Cuando la información esté llegando correctamente al servidor de Pandora FMS, se visualizará en forma de módulos de esta forma. Habilitar la vista en *modo jerárquico* en la lista de módulos ayuda a visualizar la información de forma mucho más clara:

  UX_Global_Status_sesion_fases.html	Phase fase1 OK		N/A - N/A	1	 	4 seconds
  L UX_Global_Snapshot_sesion_fases.html	Image (last error)		N/A - N/A		 	6 hours
  L UX_Global_Time_sesion_fases.html	Phase fase1 OK		N/A - N/A	9.1	 	4 seconds
  L UX_Status_sesion_fases.html.fase1_0	Phase fase1 OK		N/A - N/A	1	 	4 seconds
  L UX_Time_sesion_fases.html.fase1_0	Phase fase1 OK		N/A - N/A	2.1	 	4 seconds
  L UX_Snapshot_sesion_fases.html.fase1_0	Image (last error)		N/A - N/A	No image ava 	 	6 hours

Accediendo a la sección *WUX* del agente podrá ver detalles adicionales de la transacción y sus fases.

## Recuperación de valores

Pandora FMS UX es capaz de recuperar datos directamente de la web durante la ejecución de los tests de experiencia de usuario.

Para utilizar esta nueva característica, lo único que hay que agregar es la orden de capturar en forma de comentario en el test de Selenium:

The screenshot shows the Selenium IDE 2.9.1 interface. The window title is 'ejemplito (untitled suite) - Selenium IDE 2.9.1'. The Base URL is 'http://fringe.lab.artica.lan/'. The test case 'ejemplito' is selected in the left sidebar. The main area displays a table of commands and their targets/values.

Command	Target	Value
phase_start:	Entrada a página	
open	/prueba.php	
type	id=campo	ejemplito
clickAndWait	name=enviar	
phase_end:	Entrada a página	
getValue;numero;	generic_data;	<span style="font-size:1.1em;">(\d+\.\d*)</spa
getValue;resultado en texto;	generic_data_string;	<h1>(.*?)</h1>

Below the table, the 'Command' field contains 'getValue;resultado en texto;generic\_data\_string;<h1>(.\*?)</h1>'. The 'Target' and 'Value' fields are empty. There are 'Select' and 'Find' buttons next to the 'Target' field.

The bottom panel shows the 'Log' tab with the following content:

```

open(url)
Arguments:
  • url - the URL to open; may be relative or absolute
Opens an URL in the test frame. This accepts both relative and absolute URLs. The "open" command waits for the page to load before proceeding, ie. the "AndWait" suffix is implicit. Note: The URL must be on the same domain as the runner HTML due to security restrictions in the browser (Same Origin Policy). If you need to open an URL on another domain, use the Selenium Server to start a new browser session on that domain.

```

En el ejemplo adjunto, se recuperan dos valores después de realizar la navegación, y se representarán como módulos en Pandora FMS (número y resultado en texto).

Los pasos para ordenar al *plugin* la recogida de datos son los siguientes:

- Agregar un nuevo comentario al fichero de pruebas con el grabador Selenium.
- Agregar la directiva al comentario.

Directiva de captura de datos

```
getValue;<module name>;<module data type>;<Perl regular expression for data capture>
```

¿Qué representa cada campo separado por punto y comas?

- getValue: Instrucción para el sistema UX.
- <module name>: Nombre del módulo. El nombre del módulo tal y como aparecerá en Pandora FMS.

- `<module data type>` (Tipo de dato del módulo): Qué tipo se utilizará; puede ser cualquiera de los **tipos de datos** soportados por Pandora FMS
- `<Perl regular expression for data capture>` (Expresión regular Perl a utilizar para la captura de datos): Debe ser una expresión regular Perl, con la orden de captura de información entre paréntesis.

Por ejemplo, en el texto:

```
<p> The temperature of the solar panel is: <span class="temperature">54°C</span></p>
```

Si quiere recuperar la temperatura para llevar un seguimiento del valor, especifique la expresión regular como sigue:

```
<span class="temperature">(\d+\.\.*\, *\d*) .*</span>
```

De tal manera que recuperará el valor 54 del ejemplo, o posibles valores con posiciones decimales.

La directiva de captura completa quedaría como sigue:

```
getValue;solar panel temperature;generic_data;<span class="temperature">(\d+\.\.*\, *\d*) .*</span>
```

Los comandos que generan módulos son:

- `getValue`: Extrae un valor.

```
getValue;module_name;module_type;REGEX_string_match
```

- `getVariable`: (Versión NG 753 o superior ) Extrae un valor específico de una variable.

```
getVariable;module_name;module_type;var_name
```

- `getScreenshot`: Extrae una captura de pantalla.

```
getScreenshot;module_name
```

`getValue` es un comando que pertenece al tipo que **generan los módulos** PFMS. Visita el enlace anterior para saber más sobre ellos.

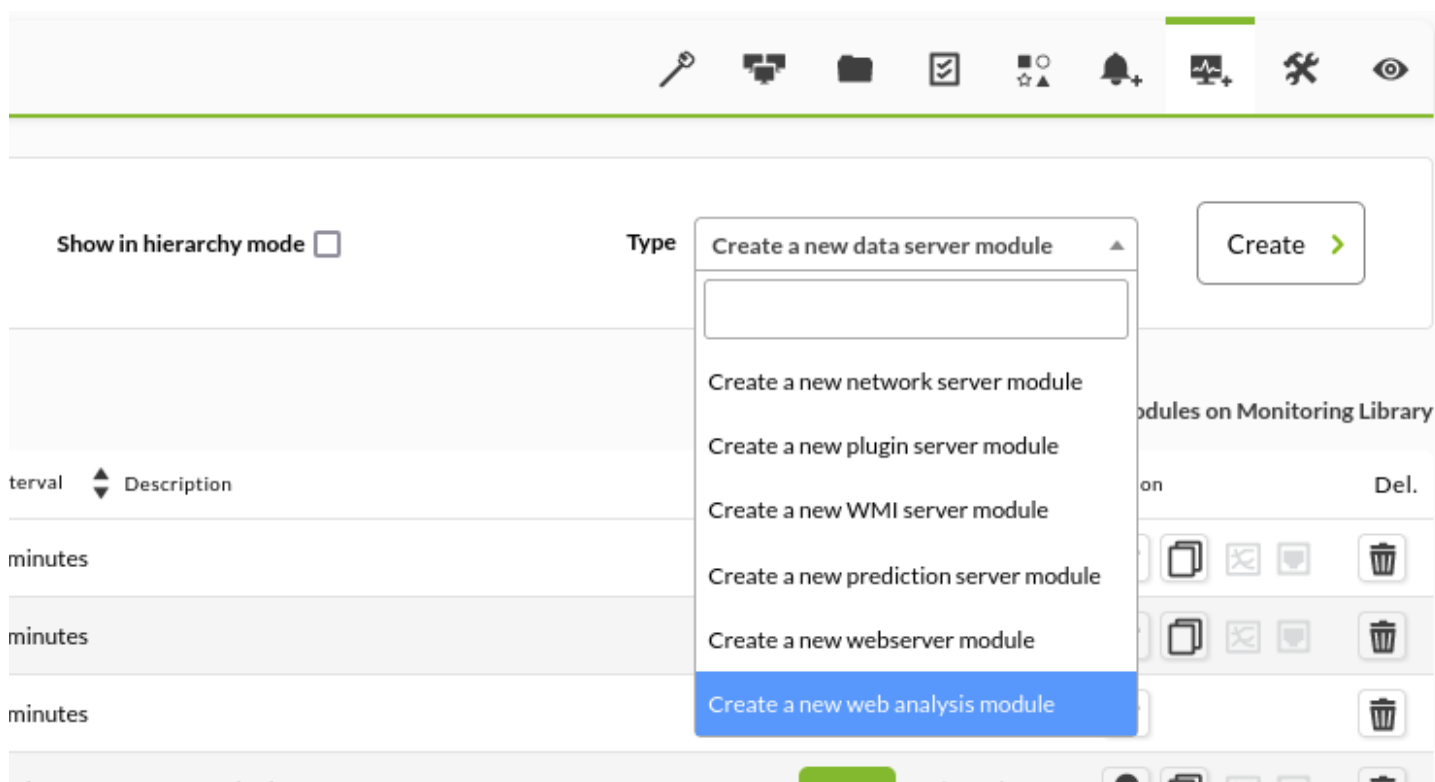
Todos los comandos propios deben aparecer en el fichero de test (`.side`) como comentarios. En caso contrario, fallará a la hora de probar el test en la extensión de Selenium.

## Gestión y visualización de datos

### Crear un módulo de análisis web en Pandora FMS Console

Una vez tenga una sesión de navegación grabada, es hora de desplegarlo como Módulo en la Consola de Pandora FMS.

Para ello acceda a cualquier agente asociado a un servidor con la funcionalidad WUX habilitada y cree un Módulo de tipo Análisis web seleccionando la opción Create a new web analysis module:



Haga clic en el botón Create y luego debe rellenar todos los campos que presenta el formulario:

✓ **Base options**

Name  Disabled   
ID 107

Run performance tests

Target web site

Execute tests from

Browser

User data dir

Profile

Accept insecure certificates

Keep counters

FF threshold  Change all statuses :   
 Change each status : To `normal`  To `warning`  To `critical`

Historical data

Paste your selenium test, exported as HTML, here

No file selected. Upload your selenium test in html or side (only Selenium 3) format

## Name

Módulo WUX (Padre); recuerde que todos los submódulos con los resultados de la monitorización serán dependientes de este nuevo módulo.

## Run performance tests

Especifica el lanzar pruebas de rendimiento, es decir, monitorizar no solo la experiencia de navegación, sino también las estadísticas de rendimiento del acceso al sitio web objetivo.

## Execute tests from

Indica el servidor WUX que ejecutará el chequeo.

## Browser

Navegador web con el cual ejecutar el chequeo.

El navegador indicado debe figurar en el *grid* o servidor de Selenium utilizado por el servidor WUX.

User data dir

(*Versión 764 o posterior*) Opcional: Permite especificar un directorio para datos.

La opción User data dir solamente trabaja en Google Chrome web browser.

Profile

(*Versión 764 o posterior*) Opcional: Permite especificar un perfil de usuario. Si el nombre de perfil no existe, utiliza el perfil por defecto.

La opción Profile solamente trabaja en Google Chrome web browser.

Accept insecure certificates

Si se habilita se aceptará cualquier certificado no seguro (autofirmado, expirado, etc.) que forme parte de la navegación.

Esta opción solo está disponible para los navegadores Google Chrome® y Mozilla Firefox®, y solo tendrá efecto si el chequeo es ejecutado por un servidor de Selenium 3.

Historical data

Guarda o no la información histórica de los módulos que monitorizan esta experiencia de usuario.

Área de texto

Copie (o cargue con el botón Browse...) el contenido de los ficheros de la sesión de navegación que ha grabado previamente.



Advanced options

Description

Custom ID

Interval

FF interval  ⓘ

Retries  ⓘ

FF interval

Intervalo de tiempo de ejecución del módulo (en segundos).

Retries

Número de reintentos de lanzamiento del módulo.

## Macros personalizadas

Custom macros

Custom macros (+)

Name

Value

El funcionamiento de las macros personalizadas implica la sustitución de determinadas cadenas de texto presentes en el fichero de sesión de navegación por ciertos valores personalizables.

En el caso de módulos de análisis web, se ha mejorado esta funcionalidad, permitiendo macros dinámicas que permiten que estos valores se traduzcan en fechas y horas variables.

*¿Por qué esta funcionalidad?*

Suponga que necesita monitorizar, mediante una sesión de navegación, el correcto

funcionamiento de una aplicación web de reserva de salas de reuniones.

Si estableciera una fecha y hora fija a la hora de rellenar los datos del formulario, es posible que el sistema cancelase la reserva porque en algún momento quede en el pasado; por ejemplo, podría estar intentando reservar una sala para la semana pasada.

También es posible que encontremos con un máximo de tiempo para poder efectuar esa reserva, y que el sistema nos obligue a reservar la sala dentro de un periodo de tiempo concreto, por ejemplo no superior al mes vigente.

Para evitar tener que editar la macro cada pocos días, y despreocuparse del apartado de configuración, puede usar las macros dinámicas, indicando al sistema que reserve la sala siempre para el día siguiente al momento de realizar el test.

Para poder usar esta funcionalidad, los valores deberán tener un formato específico, logrando las siguientes sustituciones posibles:

- @DATE\_FORMAT: fecha/hora actual con formato definido por el usuario.
- @DATE\_FORMAT\_nh: horas.
- @DATE\_FORMAT\_nm: minutos.
- @DATE\_FORMAT\_nd: días.
- @DATE\_FORMAT\_ns: segundos.
- @DATE\_FORMAT\_nM: mes.
- @DATE\_FORMAT\_nY: años.

Donde “n” puede ser un número sin signo (positivo) o negativo.

Y FORMAT sigue el estándar de [strftime de Perl](#)

Ejemplos:

```
@DATE_%Y-%m-%d %H:%M:%S
@DATE_%H:%M:%S_300s
@DATE_%H:%M:%S_-1h
```

## Visualización de los datos

La información generada por WUX se visualizará en forma de módulos de la siguiente forma. El habilitar la vista en *modo jerárquico*, en la lista de módulos, ayudará a visualizar la información de forma mucho más clara:

Total de elementos 18

F.	P.	Tipo ▲▼	Nombre módulo ▲▼	Descripción	Estado ▲▼	Advertencia	Datos	Gráfico
○			coddns			N/A - N/A	1	
			L coddns_Global_Status	Test OK		N/A - N/A	1	
			L coddns_Global_Time	Test OK		N/A - N/A	6,6	
			L coddns_Phase 0: Wordpress_Status			N/A - N/A	1	
			L coddns_Phase 0: Wordpress_Time			N/A - N/A	4,7	
			L coddns_Phase 1: Application_Status			N/A - N/A	1	
			L coddns_Phase 1: Application_Time			N/A - N/A	1,9	
			L coddns_UX_Stats_DNS			N/A - N/A	4	
			L coddns_UX_Stats_TSSL			N/A - N/A	604	
			L coddns_UX_Stats_TST			N/A - N/A	829	
			L coddns_UX_Stats_TT			N/A - N/A	862	
			L coddns_UX_Stats_TTC			N/A - N/A	33	
			L coddns_UX_Stats_TTCP			N/A - N/A	51	
			L coddns_UX_Stats_TTR	Time global spent in retrieve https://coddns.es (selected it...		N/A - N/A	9,111	
			L coddns_UX_Stats_TTR_css	Time spent in retrieve css resources from https://coddns.es		N/A - N/A	2,649	
			L coddns_UX_Stats_TTR_image	Time spent in retrieve image resources from https://coddns.e...		N/A - N/A	2,230	
			L coddns_UX_Stats_TTR_js	Time spent in retrieve js resources from https://coddns.es		N/A - N/A	3,398	
			L coddns_UX_Stats_TTR_Main	Time spent in retrieve main HTML from https://coddns.es		N/A - N/A	834	

Dentro de esta jerarquía encontraremos los siguientes módulos:

- *módulo\_Global\_Status*: Indicará el estado global de la navegación completa.
  - En el caso de que exista una grabación da el estado de la grabación WUX.
  - En el caso que se habilite los **run performance Test** pero no se incluya la grabación, el estado del módulo Global Status es el de la comprobación que se realiza al obtener estos módulos.
- *módulo\_Global\_Time*: Indicará el tiempo global empleado en la navegación completa.
- *módulo\_Global\_Screenshot*: Contiene una imagen con el resultado del error de navegación, solo se generará en caso de error.
- *módulo\_Phase X: Nombre de fase\_Status*: Indicará el estado de la navegación durante la fase X.
- *módulo\_Phase X: Nombre de fase\_Time*: Indicará el tiempo empleado en la fase X.

Ejemplo: Vista de captura de error.

The screenshot displays the Pandora FMS console interface. On the left, there is a sidebar with navigation options like 'Full list of monitors' and 'List of modules'. The main area shows a 'Current data at 2017-09-08 01:45:59' header above a screenshot of a website. Below this, a table lists monitoring results with columns for Description, Status, Warn, Data, and Graph.

Description	Status	Warn	Data	Graph
[Acceso al portal]: Failed to execute clickAndWait at link=M...	Green	N/A - N/A	1	100%
Test OK	Green	N/A - N/A	1	100%
	Green	N/A - N/A	17,6	100%
	Green	N/A - N/A	1	100%
	Green	N/A - N/A	5,7	100%
	Green	N/A - N/A	1	100%
	Green	N/A - N/A	3,3	100%
	Green	N/A - N/A	1	100%
	Green	N/A - N/A	2,5	100%

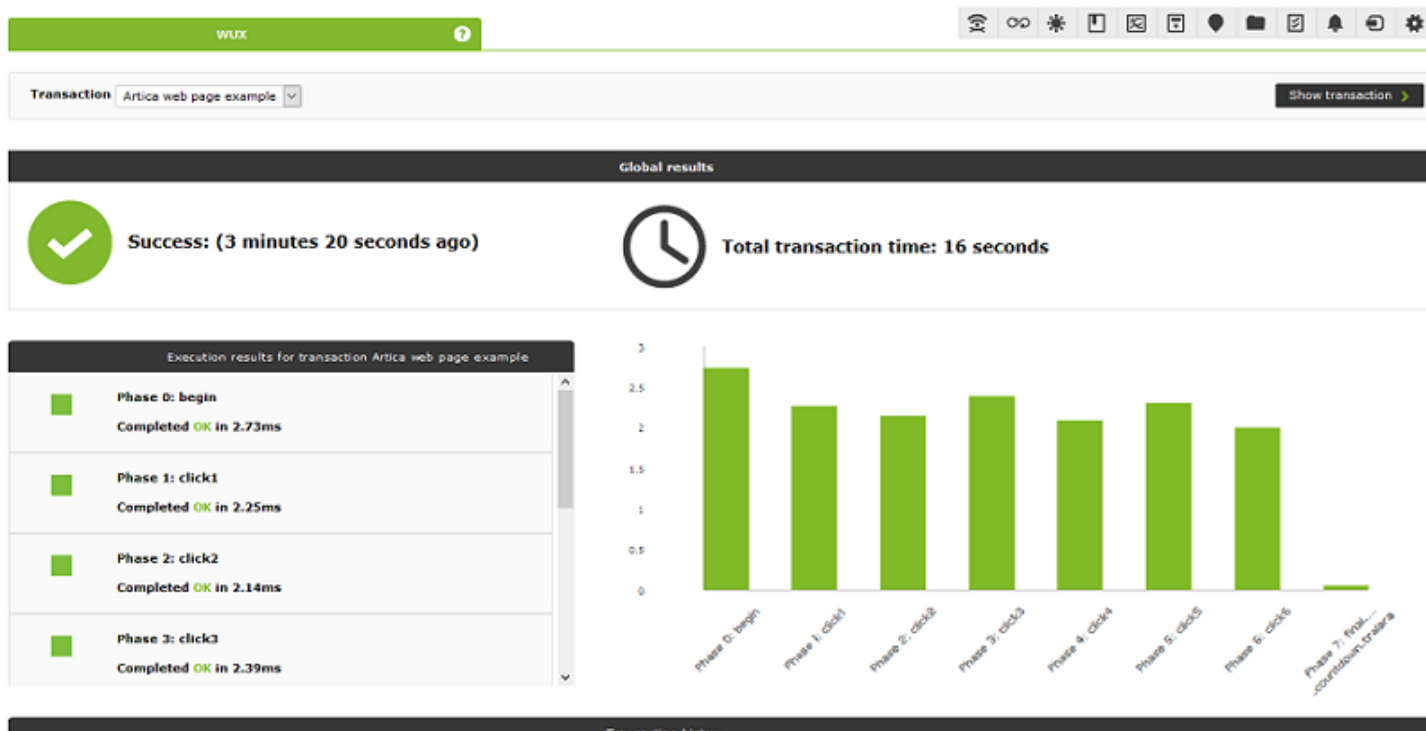
Atención: Si ha actualizado desde versiones anteriores a Pandora FMS 7.0NG 712 debe realizar un pequeño cambio.

Para poder almacenar las capturas de pantalla generadas por WUX Server, deberá lanzar las siguientes operaciones en su esquema de base de datos:

```
alter table tagente_estado modify column datos mediumtext;
alter table tagente_datos_string modify column datos mediumtext;
```

Si no realiza estas acciones las capturas podrían verse cortadas.

Accediendo a la sección *WUX* del agente puede ver detalles adicionales de la transacción y sus fases:



Las estadísticas del sitio web se resumen en los siguientes conceptos:

- Stats\_TT: Tiempo total en obtener el sitio web.
- Stats\_TDNS: Tiempo total en resolver la dirección IP del objetivo.
- Stats\_TTCP: Tiempo empleado en conectar vía TCP.
- Stats\_TSSL: Tiempo empleado en establecer comunicación SSL.
- Stats\_TST : Tiempo transcurrido hasta que inició la transferencia de datos.
- Stats\_TTC: Tiempo transfiriendo datos; agrupará todos los tiempos de transferencia de recursos.
- Stats\_TTR: Tiempo empleado en transferir el contenido de la página.
- Stats\_TTR\_main: Tiempo empleado en transferir el código HTML.
- Stats\_TTR\_image: Tiempo empleado en transferir los recursos de tipo imagen (png | jpg | jpeg | bmp | tiff | gif | webp | svg).
- Stats\_TTR\_css: Tiempo empleado en transferir las hojas de estilo.
- Stats\_TTR\_js: Tiempo empleado en transferir los ficheros JavaScript.

## Asignar alertas a un módulo de análisis web

Las alertas asociadas a los módulos de análisis web siguen la misma dinámica de funcionamiento que todo el sistema de alertas en Pandora FMS.

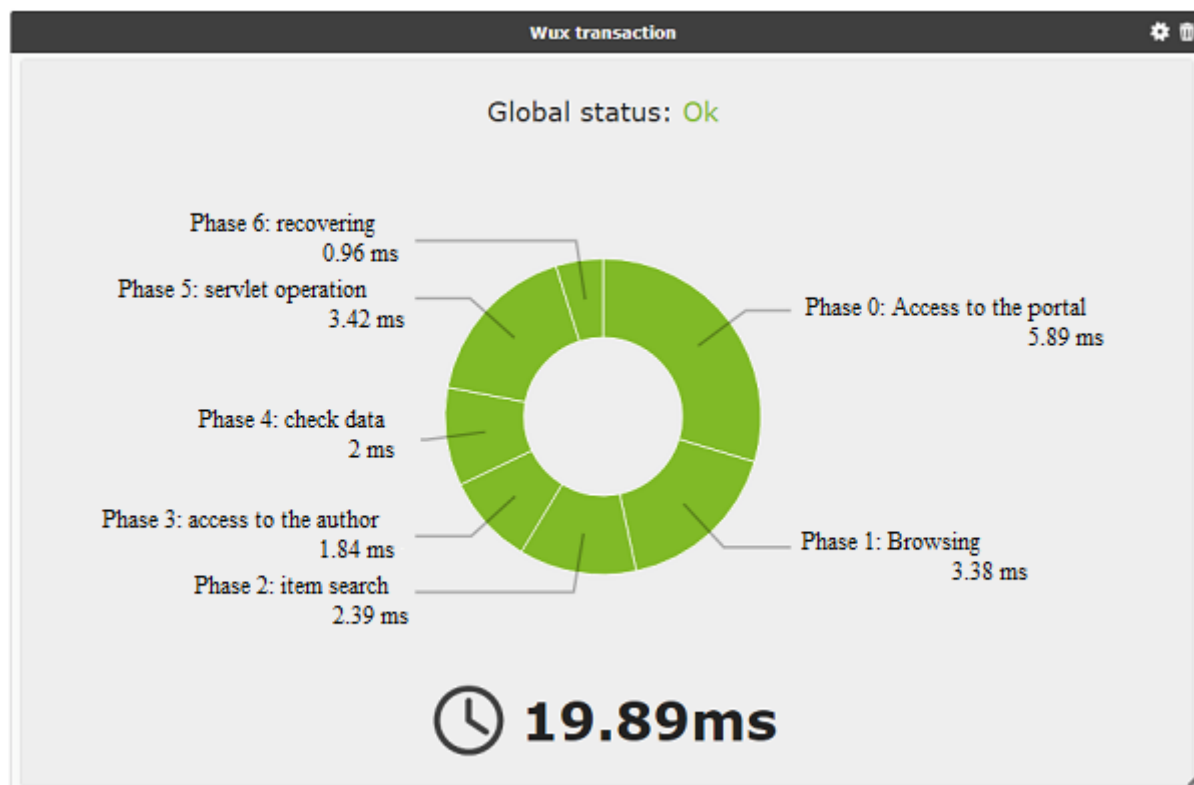
Por compatibilidad se recomienda asignar las plantillas de alertas sobre los subelementos autogenerados por el módulo de análisis web, como por ejemplo:

- El estado de la navegación global.
- Alertas sobre umbrales de tiempos.
- Alertas con plantilla de aviso puesta en "siempre" para los módulos captura de imagen de resultados.

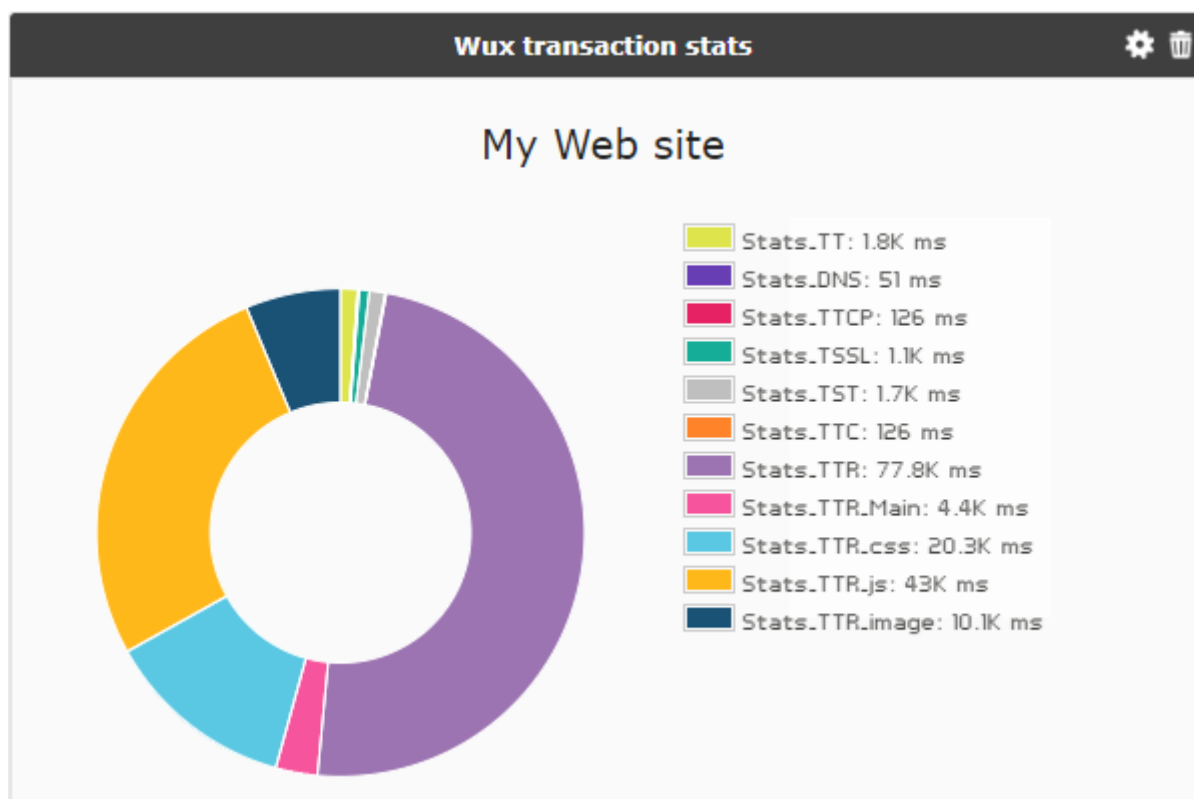
## Nuevos widgets para Dashboards

Para poder representar la información de la navegación de usuario, se han agregado dos nuevos tipos de widget:

Widget de estado de las fases de navegación



Widget de estadísticas del sitio web



## Experiencia de usuario de escritorio (PDR)

### Despliegue del sistema PDR

#### E

El sistema PDR únicamente funciona en sistemas Windows®, y una vez en ejecución por el Agente de Pandora FMS éste debe correr en modo proceso, ya que de correr en modo servicio no funcionará. Además, no será posible bloquear la sesión de escritorio, por lo que se recomienda su uso en máquinas virtuales.

Prerrequisitos:

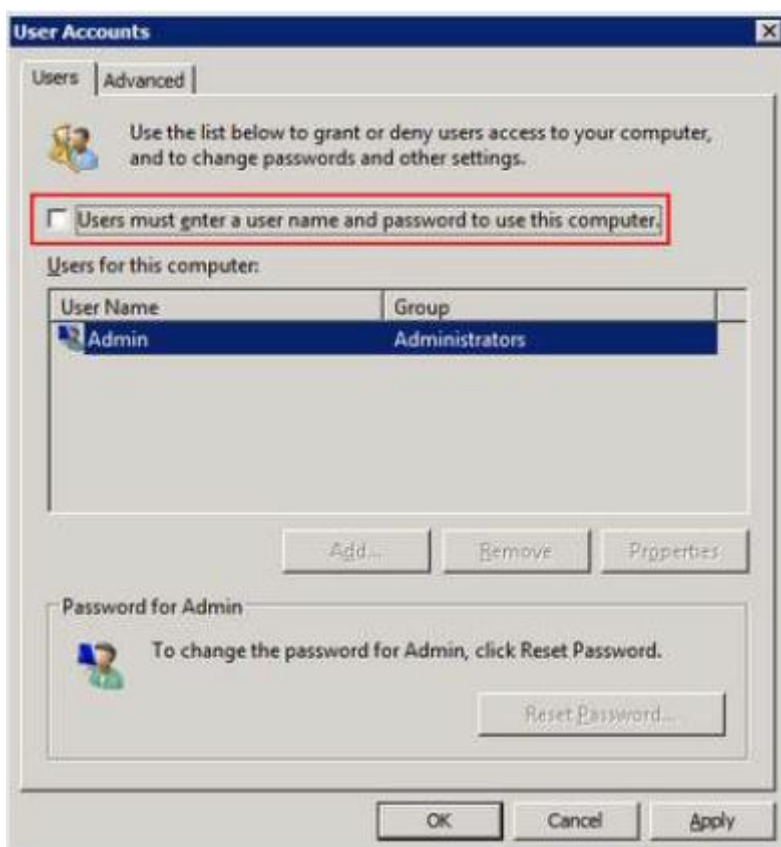
- Sistema Windows® en modo escritorio con auto arranque y auto inicio de sesión.

Para lograr dicha configuración ejecute:

Para versiones anteriores a Windows 10®:

```
control userpasswords2
```

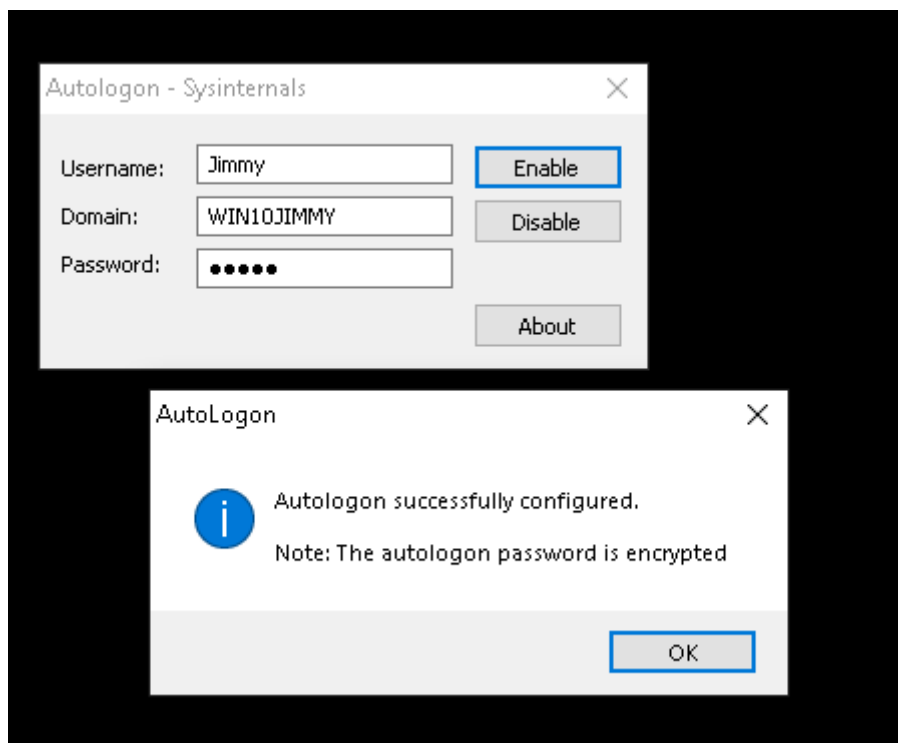
Y desmarque la casilla Users must enter a user name and password to use this computer:



Haga clic en el botón Ok e introduzca el nombre de usuario y su contraseña.

Para Windows 10®:

Una manera de habilitar el auto inicio en MS Windows®, versión 10, es utilizar Powertools de Sysinternals (disponible en la [sección de descargas](#) de la documentación de Microsoft®) y habilitar el usuario que realizará la monitorización, por ejemplo:



Para preparar el entorno cree el siguiente directorio:

```
C:\PDR
```

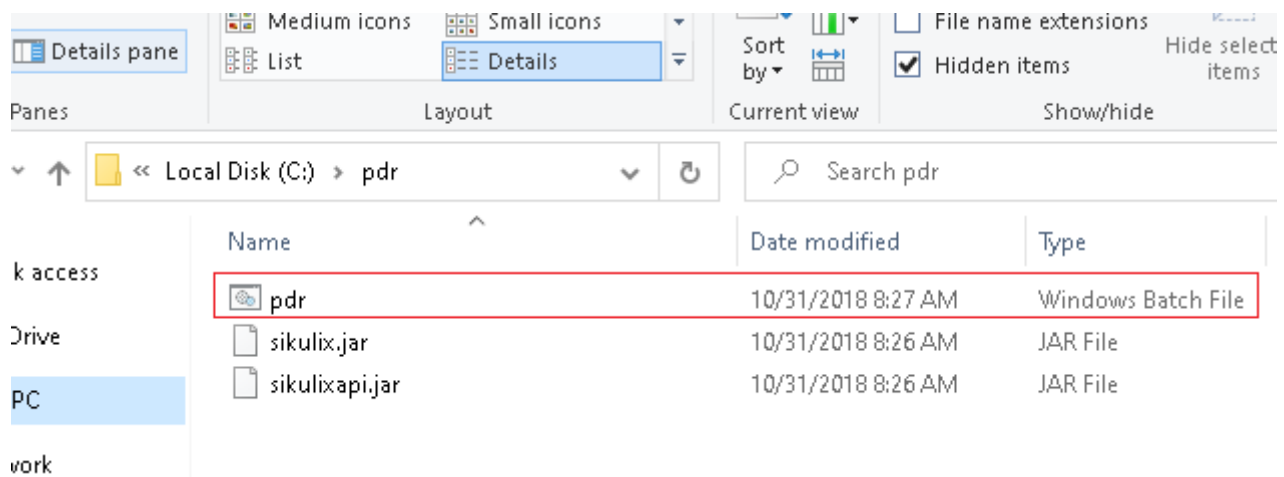
Y descomprima el fichero PDR.rar (proporcionado en el siguiente enlace <https://pandorafms.com/library/pdr-cmd-for-ux-monitoring/>) en dicha carpeta.

## Ejecución de sesiones PDR

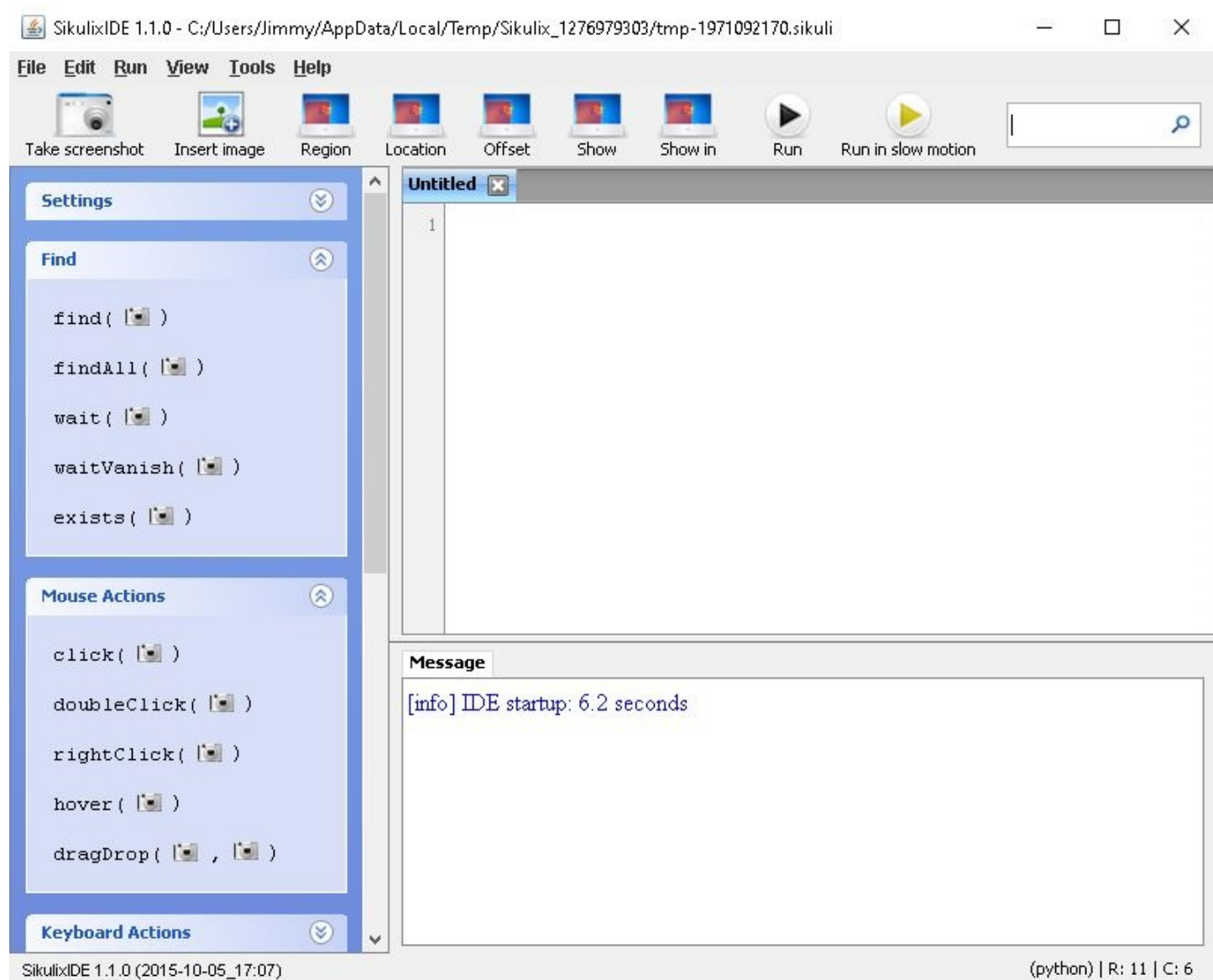
Para iniciar la grabadora, ejecute el siguiente script:

```
C:\PDR\pdr
```



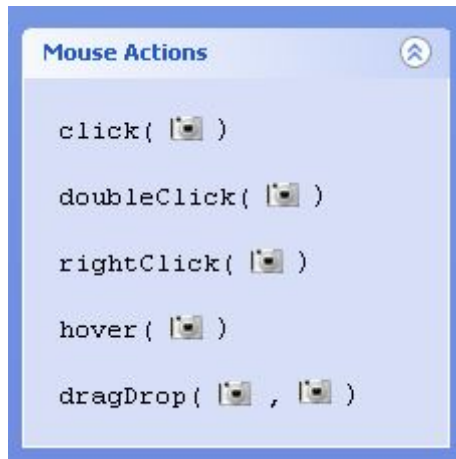


Tras el proceso de carga, entra en el entorno de grabación:

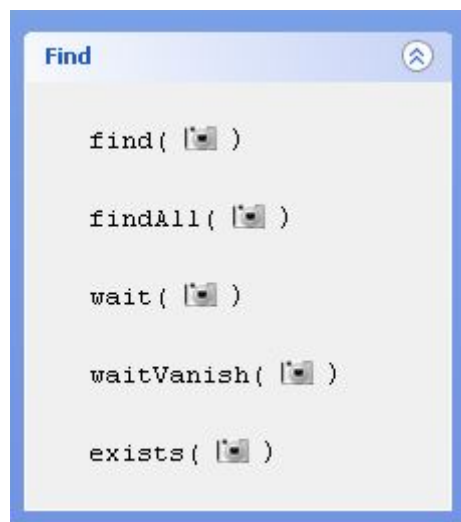


Seleccione las acciones que desea realizar y haga una captura de pantalla en la zona donde desea aplicarlas. A continuación puedes encontrar una lista con las más comunes.

Acciones generales:



Acciones de control de flujo:



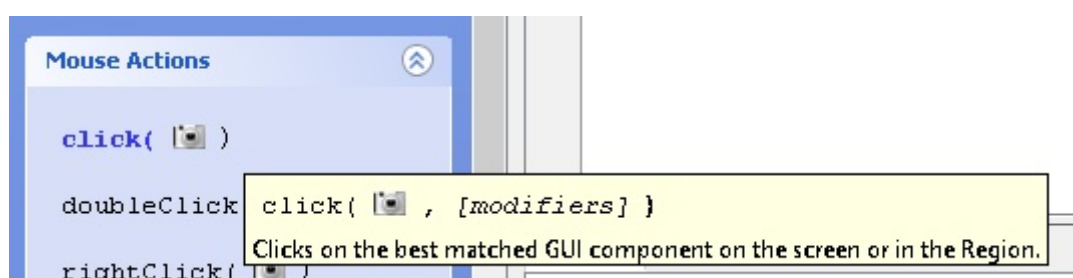
Más ayuda lingüística: <http://sikulix-2014.readthedocs.io/en/latest/index.html>

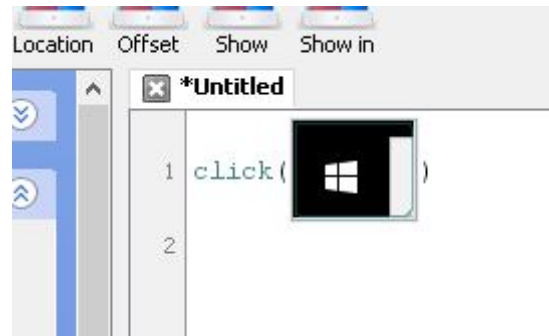
Al grabar el proyecto, se creará una carpeta que contendrá los siguientes elementos:

- Fichero .py con el código del script de automatización.
- Imágenes para controlar la navegación.

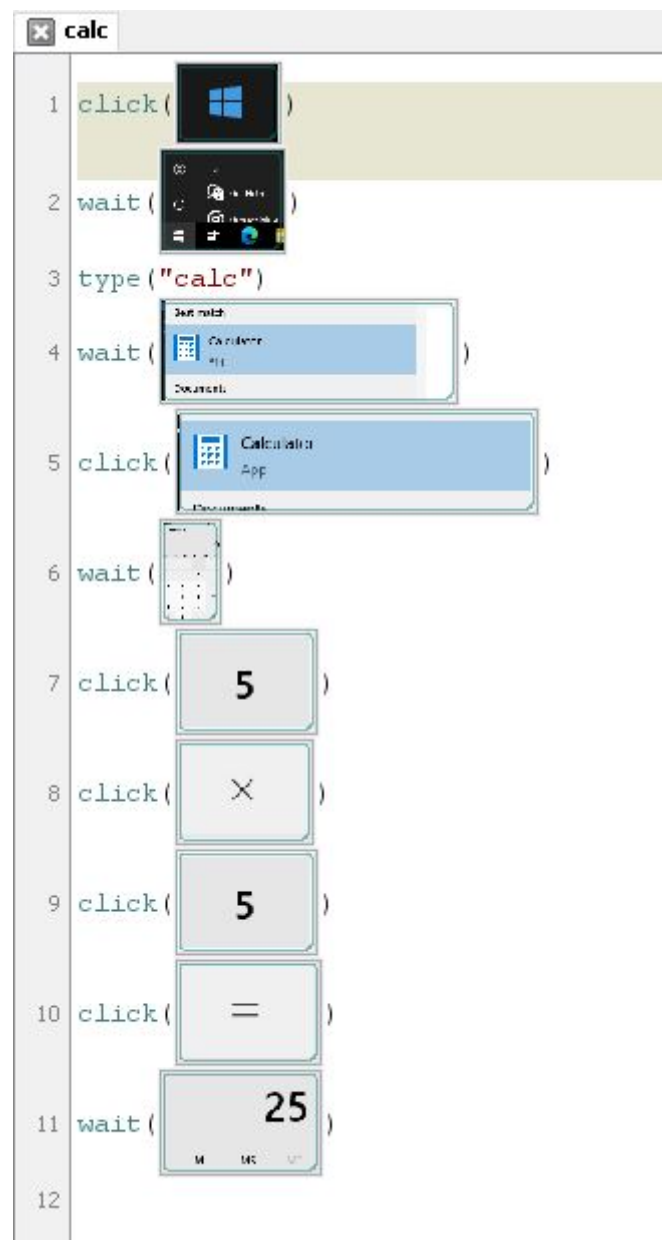
Un ejemplo sencillo de ejecución podría ser la supervisión de que su calculadora de Windows funciona correctamente. Mostraremos todo el proceso a través de capturas de pantalla.

1. Elija la acción “click” y el área donde desea aplicar la acción. La pantalla cambiará al modo “area selection”:





2. Introduzca la acción `type` y el texto "calc". A continuación, introduzca las acciones para esperar a que aparezca la zona de la "calculadora" y haga clic en ella. A continuación, espere a que aparezca la ventana de la calculadora para introducir las siguientes acciones de esta manera:



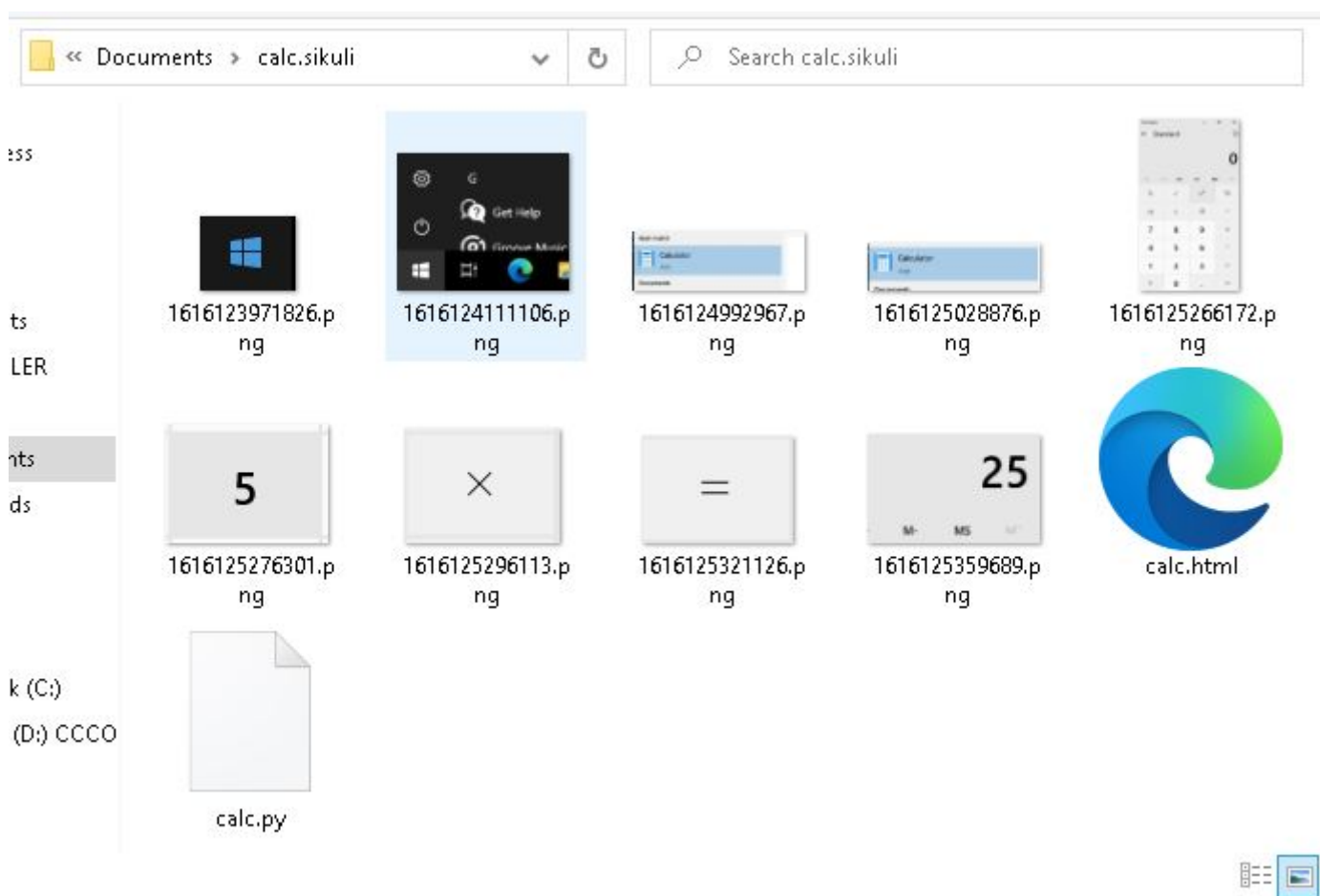
3. A continuación, haga clic en los botones correspondientes, seleccionando las zonas como en los pasos anteriores.

4. Por último, guarde el proceso y ejecútelo utilizando la función en el botón Run.

## Comentarios importantes:

- Si hace doble clic en cualquiera de las imágenes dentro del entorno de grabación, podrá ajustar los detalles de la captura de pantalla de control, como seleccionar el punto exacto en el que hacer clic.
- Es muy recomendable introducir acciones como *wait* después de cada clic para asegurarse de que la ejecución no se detendrá debido a un retraso del sistema operativo.
- La grabadora buscará el área que se vea igual que la captura de pantalla, por lo que se debe tener especial cuidado en caso de que algo se resalte cuando el botón se encuentre arriba (por ejemplo, los botones de la calculadora cambian a color naranja cuando se pasa el ratón sobre ellos).

En la carpeta donde guardó la grabación podrá ver los archivos de imagen y el archivo Python.



**Nota:** Puede personalizar la captura realizada desde el motor de grabación PDR utilizando el siguiente código:

```
## OPTIONAL ##capture file names
import shutil
file = "C:\probes\screenshot_prueba.sikuli.png"
focusWindow = App.focusedWindow()
regionImage = capture(focusWindow)
shutil.move(regionImage, file)
```

Esto creará una imagen de la ventana activa en el destino indicado por el archivo, con la que podrá hacer una captura de pantalla justo cuando lo desee.

Puede personalizar la captura de pantalla utilizando coordenadas. Para ello, especifique la siguiente sintaxis:

```
capture(x, y, w, h)
```

Donde:

X: posición horizontal del rectángulo para tomar una captura de pantalla.

Y: posición vertical del rectángulo para tomar una captura de pantalla.

W: Ancho de la pantalla.

H: Altura de la pantalla.

### **Grabar una sesión transaccional con Pandora FMS UX PDR**

No es necesario grabar una sesión específica; simplemente grabe las sesiones que necesite en *scripts* diferentes. Será Pandora FMS UX quien gestione la devolución de resultados para organizarla como una transacción compleja. En el siguiente punto veremos cómo hacer la llamada de la ejecución para que funcione correctamente.

Si va a realizar varias grabaciones para crear un proceso transaccional debe tener cuidado de que al realizar la grabación los elementos que busca estén presentes. Se recomienda la ejecución manual de las grabaciones que quiera utilizar en un único proceso transaccional de forma seguida, para asegurar de que las acciones transcurren según lo esperado.

En el siguiente ejemplo se ha realizado un proceso de grabación que constituirá la segunda fase del proceso empleado en el ejemplo anterior. Vamos a tomar el resultado devuelto por la calculadora y pegarlo en un bloc de notas. La grabación presenta el siguiente aspecto:

```

1 rightClick( 25 )
2 wait( Copy )
3 click( Copy )
4
5 click( Windows )
6 wait( Start Menu )
7 type( "notepad" )
8 wait( Notepad App )
9 click( Notepad App )
10 click( Edit )
11 wait( Edit Menu )
12 click( Paste Ctrl+V )

```

Y hemos incluido otra secuencia que consistirá en guardar el archivo de texto en una ubicación determinada, sobrescribiendo el anterior. Esto otorga gran flexibilidad, ya que abre la posibilidad de monitorizar paralelamente estos ficheros reflejando la información que necesites en todo tipo de variedad de chequeos de aplicaciones pesadas de escritorio. Esta tercera secuencia consiste en los siguientes pasos:

```

13 click( Paste Ctrl+V )
14 click( File Menu )
15 click( Save As )
16 wait( Save As Dialog )
17 type( "cal_latest" )
18 click( Save )
19

```

## Ejecución de sesiones PDR

### Ejecución estándar

Para lanzar sesiones pregrabadas de PDR, indique que el modo de trabajo es la ruta al fichero `pdr.cmd` desplegado en el punto correspondiente, el argumento de dicho fichero `-r`, el archivo que contiene las directrices de la sesión (`-script`), el directorio donde almacenar las capturas de pantalla (`-folder`) terminando en `'\'`, el cual es opcional, donde guardar las capturas en la carpeta donde está ubicado el `pdr`. También se puede introducir el número de reintentos consecutivos en caso de fallo (parámetro opcional).

En la ejecución siguiente también se personaliza la captura de pantalla para recoger únicamente la ventana activa:

```
pandora_ux_x64 -exe C:\PDR\pdr -args -r -script C:\pandora_ux\calculadora.sikuli
-folder C:\pandora_ux\ -ss_config active -retries 3
```

Se devolverán los siguientes módulos:

- `UX_Time_nombre_proyecto`.
- `UX_Status_nombre_proyecto`.
- `UX_Control_Snapshot_nombre_proyecto` (solo en la primera ejecución).

Si hay alguna fase con error, se creará también el módulo siguiente:

- `UX_Snapshot_nombre_proyecto`.

Y mostrará una imagen de la ventana activa (con `-ss_config active`) del momento del error.

Ejemplo de salida con ejecución correcta:

```
<module>
  <name><![CDATA[UX_Status_calculator.sikuli]]></name>
  <type>generic_proc</type>
  <data><![CDATA[1]]></data>
  <description><![CDATA[C:\pandora_ux\calculator.sikuli execution completed
Control snapshot rebuild ]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
</module>
<module>
  <name><![CDATA[UX_Time_calculator.sikuli]]></name>
  <type>generic_data</type>
  <data><![CDATA[20.204]]></data>
  <description><![CDATA[C:\pandora_ux\calculator.sikuli execution completed
Control snapshot rebuilt ]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
```

```

    <module_parent>UX_Status_calculator.sikuli</module_parent>
</module>
<module>
  <name><![CDATA[UX_Control_Snapshot_calculator.sikuli]]></name>
  <type>async_string</type>
  <data><![CDATA[data:image/png;base64,
IBCAIAAAA0CnfhAAAAAXNSR.../4x79e/7757f8H2C00s1C73yMAAAAASUVORK5CYII=]]></data>
  <description><![CDATA[Control image rebuilt]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_calculator.sikuli</module_parent>
</module>

```

Ejemplo de salida con ejecución fallida:

```

<module>
  <name><![CDATA[UX_Status_std.html]]></name>
  <type>generic_proc</type>
  <data><![CDATA[0]]></data>
  <description><![CDATA[Failed to execute verifyText]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
</module>
<module>
  <name><![CDATA[UX_Time_std.html]]></name>
  <type>generic_data</type>
  <data><![CDATA[15.463]]></data>
  <description><![CDATA[Failed to execute verifyText]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_std.html</module_parent>
</module>
<module>
  <name><![CDATA[UX_Snapshot_std.html]]></name>
  <type>async_string</type>
  <data><![CDATA[data:image/png;base64,
iVBORw0KGgoAAAANSUgAA...JRU5ErkJggg==]]></data>
  <description><![CDATA[Image (last error)]]></description>
  <tags>UX</tags>
  <module_group>UX</module_group>
  <module_parent>UX_Status_std.html</module_parent>
</module>

```

Si todo ha resultado correcto puede añadir la línea de ejecución al agente de Pandora FMS instalado en la máquina Windows® como *plugin* de Agente. La línea a añadir al fichero de configuración del Agente tendrá un aspecto similar a este (en una sola línea):

```

module_plugin C:\Users\pfms\Documents\Product\UX-Trans\ux\pandora_ux_x64.exe -
exe C:\PDR\pdr.bat -args -r -script C:\PDR\calc.sikuli -folder C:\PDR\ -
ss_config active -checkpoint -post "taskkill /F /IM calc.exe"

```



Como puede verse, esta ejecución tiene algunas opciones más. El parámetro `-checkpoint` sirve para mostrar una captura del resultado final aunque no exista fallo; `-post` ejecutará acciones una vez terminada la reproducción de la sesión. En este caso, vamos a cerrar el proceso de calculadora que ha iniciado la grabación, para evitar que el sistema termine fallando por exceso de ventanas abiertas o procesos similares.

Ahora que el Agente tiene la línea de ejecución `module_plugin` lista, debe lanzarlo en modo proceso, ejecutando lo siguiente desde línea de comando como administrador:

```
"%ProgramFiles%\pandora_agent\PandoraAgent.exe" --process
```

Al lanzarlo podrá ver cómo se ejecutan las acciones de forma automática conforme a la grabación. Desde este momento el proceso de agente no debe cerrarse, y tampoco se debe iniciar sesión en la máquina por escritorio remoto, o las ejecuciones podrían interrumpirse. La máquina debe quedar sin manipular, por esta razón es por la que se recomienda su uso en máquinas virtuales.

Si ya se tienen tests de automatización, se pueden incluir en Pandora FMS con la siguiente ejecución.

```
pandora_ux.64 -exe <exe of the automation system> -args <system arguments> -script <test file path>
```

- `<exe of the automation system>`: exe (ejecutable) del sistema de automatización.
- `<system arguments>`: Argumentos del sistema.
- `<test file path>`: Ruta ficheros de prueba.

## Ejecución basada en transacción

Si tenemos grabados diferentes procesos con PDR y hemos probado que funcionan al reproducirlos de forma continuada, haremos la ejecución del siguiente modo:

```
C:\Users\pfms\Documents\Product\UX-Trans\ux\pandora_ux_x64.exe -exe  
C:\PDR\pdr.cmd -args -r -t calculadora_trans -script  
C:\PDR\calc.sikuli,C:\PDR\savecalc.sikuli,C:\PDR\savefile.sikuli -folder C:\PDR\  
-ss_config active
```

Como puede observarse, simplemente indicaremos la ruta del nuevo `script` en el parámetro `-script` separada por una coma del script anterior y utilice el parámetro `-t` con el nombre de la transacción que englobará las diferentes fases. Si tuviera un proceso con más fases seguiría la misma lógica; por ejemplo:

```
pandora_ux_x64.exe -exe C:\PDR\pdr.cmd -args -r -t proceso_transaccional -script  
C:\PDR\script1,C:\PDR\script2,C:\PDR\script3,C:\PDR\script4 -folder C:\PDR\  
-ss_config active
```

La línea a añadir al fichero de configuración del Agente, para este caso, será la siguiente:

```
module_plugin C:\Users\pfms\Documents\Product\UX-Trans\ux\pandora_ux_x64.exe -
exe C:\PDR\pdr.cmd -args -r -t calculadora_trans -script
C:\PDR\calc.sikuli,C:\PDR\savecalc.sikuli,C:\PDR\savefile.sikuli -folder C:\PDR\
-ss_config active -checkpoint -post "taskkill /F /IM calc.exe"
```

Gracias al parámetro `-checkpoint` puede ver capturas del resultado final de cada fase en la consola de Pandora FMS.

Se devolverán los siguientes módulos por fase:

- UX\_Time\_nombre\_proyecto.fase\_orden.
- UX\_Status\_nombre\_proyecto.fase\_orden.

Si hay alguna fase con error, se creará también el módulo siguiente:

- UX\_Snapshot\_nombre\_proyecto.fase\_orden.

También mostrará una imagen de la web en el momento del error, en caso de que se produzca.

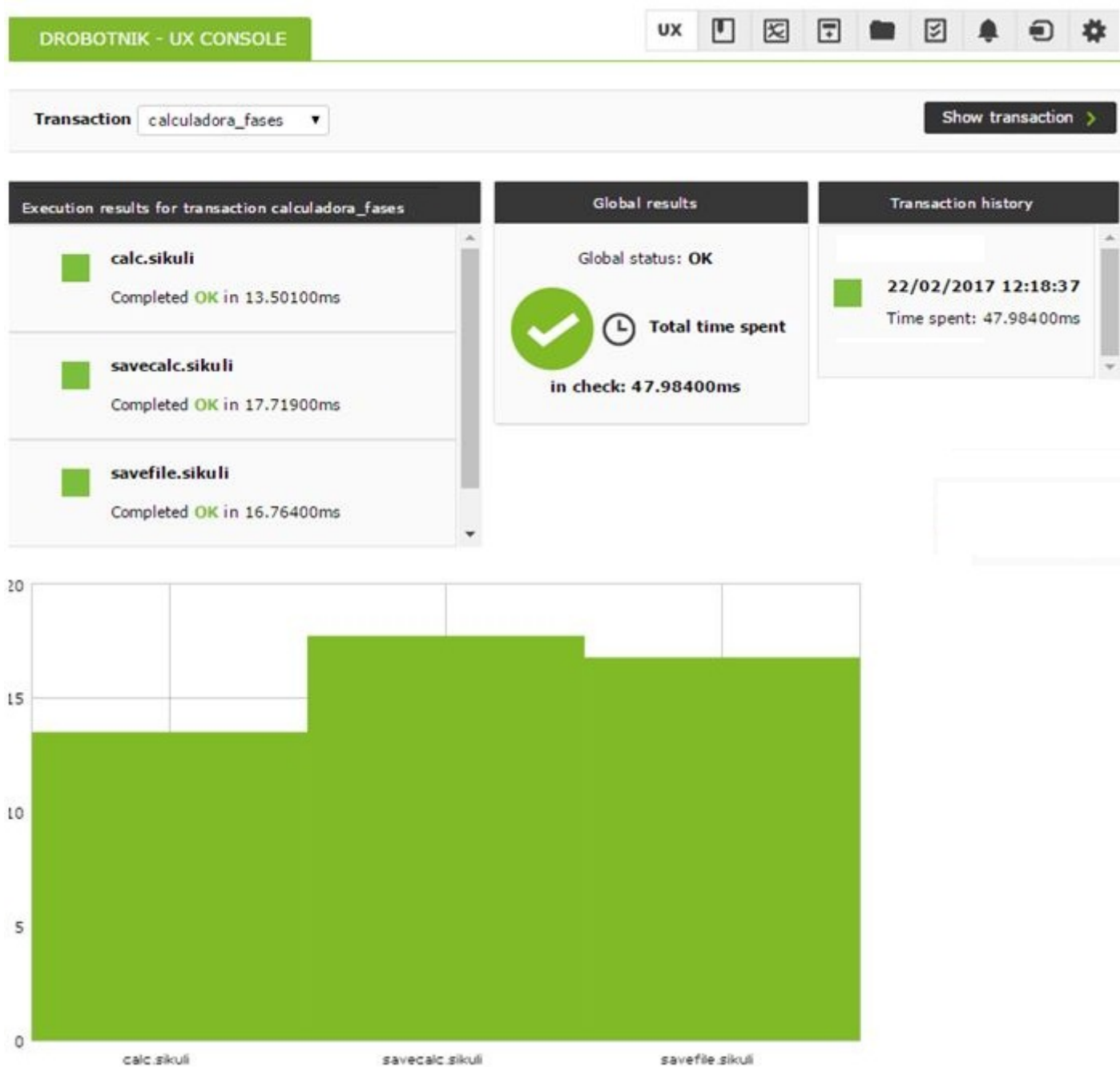
También se devolverán los módulos de resumen globales identificados con los siguientes nombres:

- UX\_Global\_Time\_nombre\_proyecto.
- UX\_Global\_Status\_nombre\_proyecto.
- UX\_Global\_Snapshot\_nombre\_proyecto.

Cuando el agente se encuentre corriendo en modo proceso y los módulos sean recuperados por Pandora FMS los veremos en la consola. De nuevo, mediante la vista en *modo jerárquico* podremos mostrar con claridad la relación entre los módulos mostrados, delimitando claramente las diferentes fases:

		UX_Global_Status_calculadora_fases	Test OK		N/A - N/A	1		101	5 seconds
		UX_Global_Time_calculadora_fases	Test OK		N/A - N/A	48		101	5 seconds
		UX_Status_calc.sikuli_0	C:\PDR\calc.sikuli execution completed Control snapshot rebu...		N/A - N/A	1		101	5 seconds
		UX_Time_calc.sikuli_0	C:\PDR\calc.sikuli execution completed Control snapshot rebu...		N/A - N/A	13.5		101	5 seconds
		UX_Control_Snapshot_calc.sikuli_0	Control image rebuilt		N/A - N/A			101	5 seconds
		UX_Status_savecalc.sikuli_1	C:\PDR\savecalc.sikuli execution completed Control snapshot ...		N/A - N/A	1		101	5 seconds
		UX_Time_savecalc.sikuli_1	C:\PDR\savecalc.sikuli execution completed Control snapshot ...		N/A - N/A	17.7		101	5 seconds
		UX_Control_Snapshot_savecalc.sikuli_1	Control image rebuilt		N/A - N/A			101	5 seconds
		UX_Status_savefile.sikuli_2	C:\PDR\savefile.sikuli execution completed Control snapshot ...		N/A - N/A	1		101	5 seconds
		UX_Time_savefile.sikuli_2	C:\PDR\savefile.sikuli execution completed Control snapshot ...		N/A - N/A	16.8		101	5 seconds
		UX_Control_Snapshot_savefile.sikuli_2	Control image rebuilt		N/A - N/A			101	5 seconds


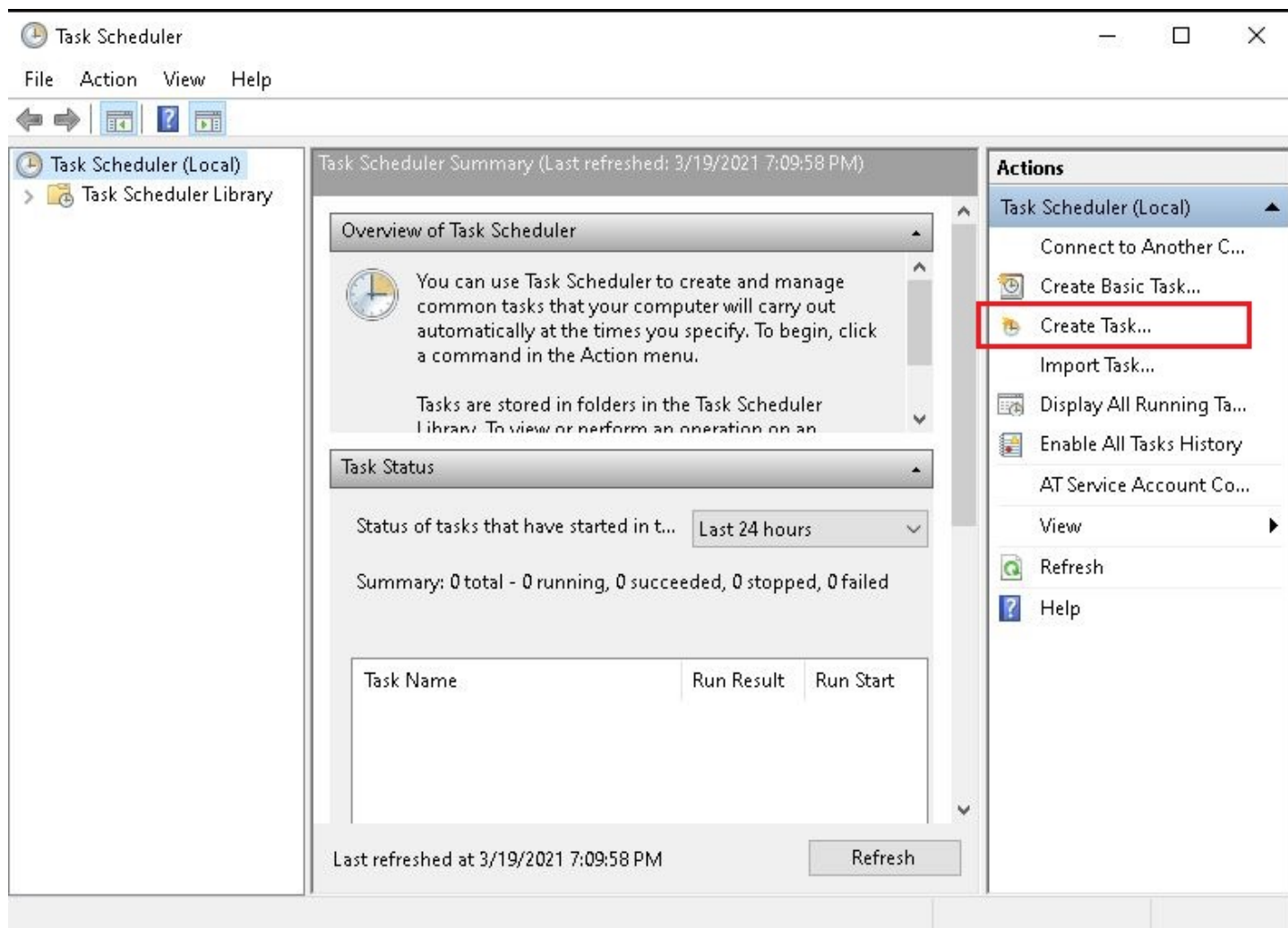
Y en el visor de transacciones podemos ver el detalle de las fases y la gráfica con los tiempos:



Las capturas de error solo se mostrarán cuando el cliente UX (PWR) y el servidor PWR estén corriendo en la misma máquina. En caso contrario, el directorio de entrega de las imágenes por parte del servidor PWR deberá ser accesible por el cliente para poder mostrar la imagen en Pandora FMS.

Será sobre estos Módulos sobre los que puede crear alertas y visualizar el histórico para construir gráficas y mostrar la evolución en el tiempo de los sistemas a monitorizar.

Se recomienda la creación de una tarea programada que arranque el agente de Pandora FMS en modo proceso cuando el equipo se inicie y así no se interrumpirán las ejecuciones aunque el sistema tenga un reinicio forzoso. Al estar en modo inicio de sesión automático sin contraseña, el agente siempre se ejecutará aunque se reinicie la máquina.

 Task Scheduler

Task Scheduler

File Action View Help

Task Scheduler (Local)

Task Scheduler Library

Task Scheduler Summary (Last refreshed: 3/19/2021 7:09:58 PM)

Overview of Task Scheduler

You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the times you specify. To begin, click a command in the Action menu.

Tasks are stored in folders in the Task Scheduler Library. To view or perform an operation on an

Task Status

Status of tasks that have started in t... Last 24 hours

Summary: 0 total - 0 running, 0 succeeded, 0 stopped, 0 failed

Task Name	Run Result	Run Start
-----------	------------	-----------

Last refreshed at 3/19/2021 7:09:58 PM Refresh

Actions

Task Scheduler (Local)

- Connect to Another C...
- Create Basic Task...
- Create Task...
- Import Task...
- Display All Running Ta...
- Enable All Tasks History
- AT Service Account Co...
- View
- Refresh
- Help

Create Task

General Triggers Actions Conditions Settings

Name: Pandora FMS Agent

Location: \

Author: WIN10JIMMY\Jimmy

Description: Starting Pandora FMS Agent, process mode.

Security options

When running the task, use the following user account:

WIN10JIMMY\Jimmy Change User or Group...

Run only when user is logged on

Run whether user is logged on or not

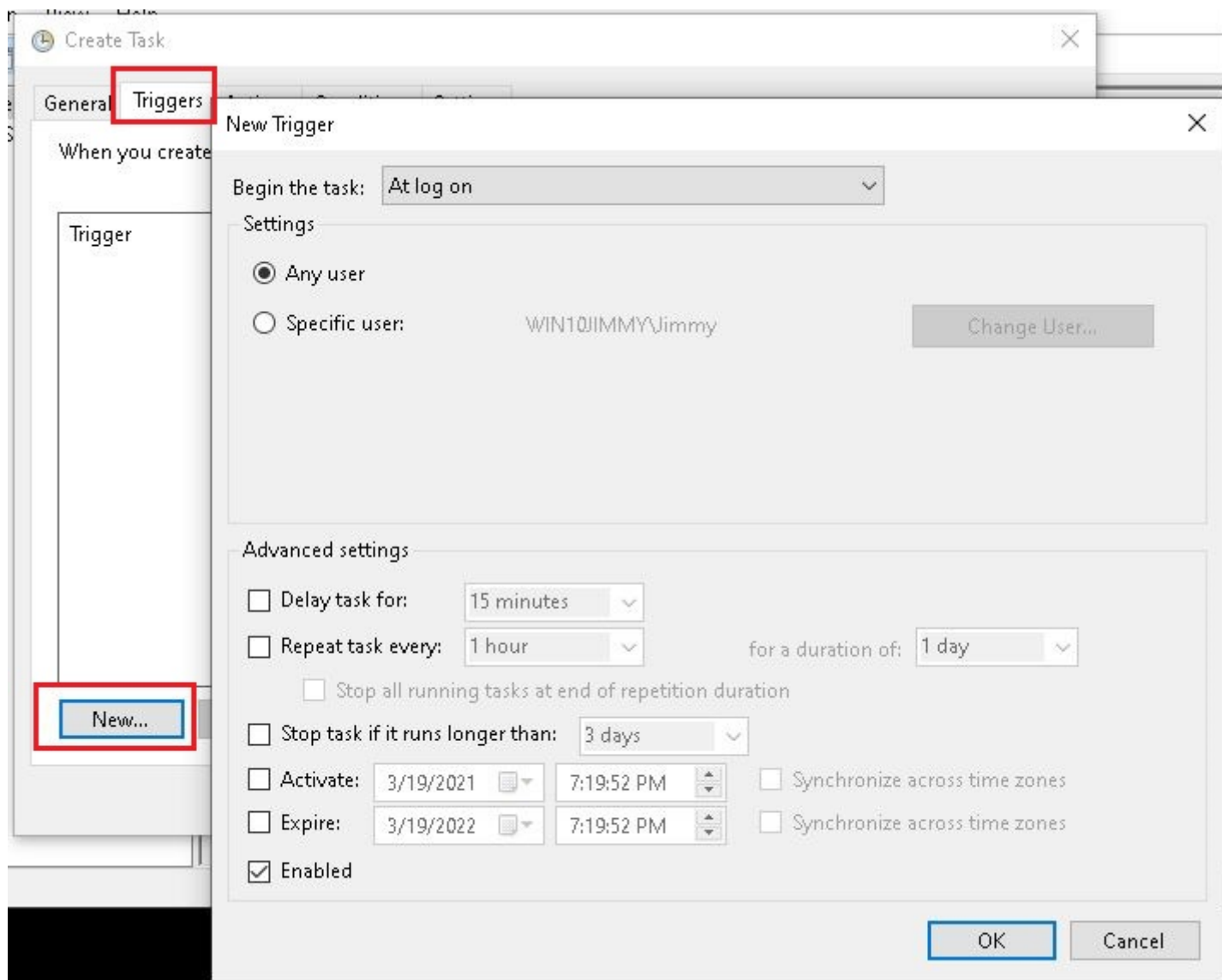
Do not store password. The task will only have access to local computer resources.

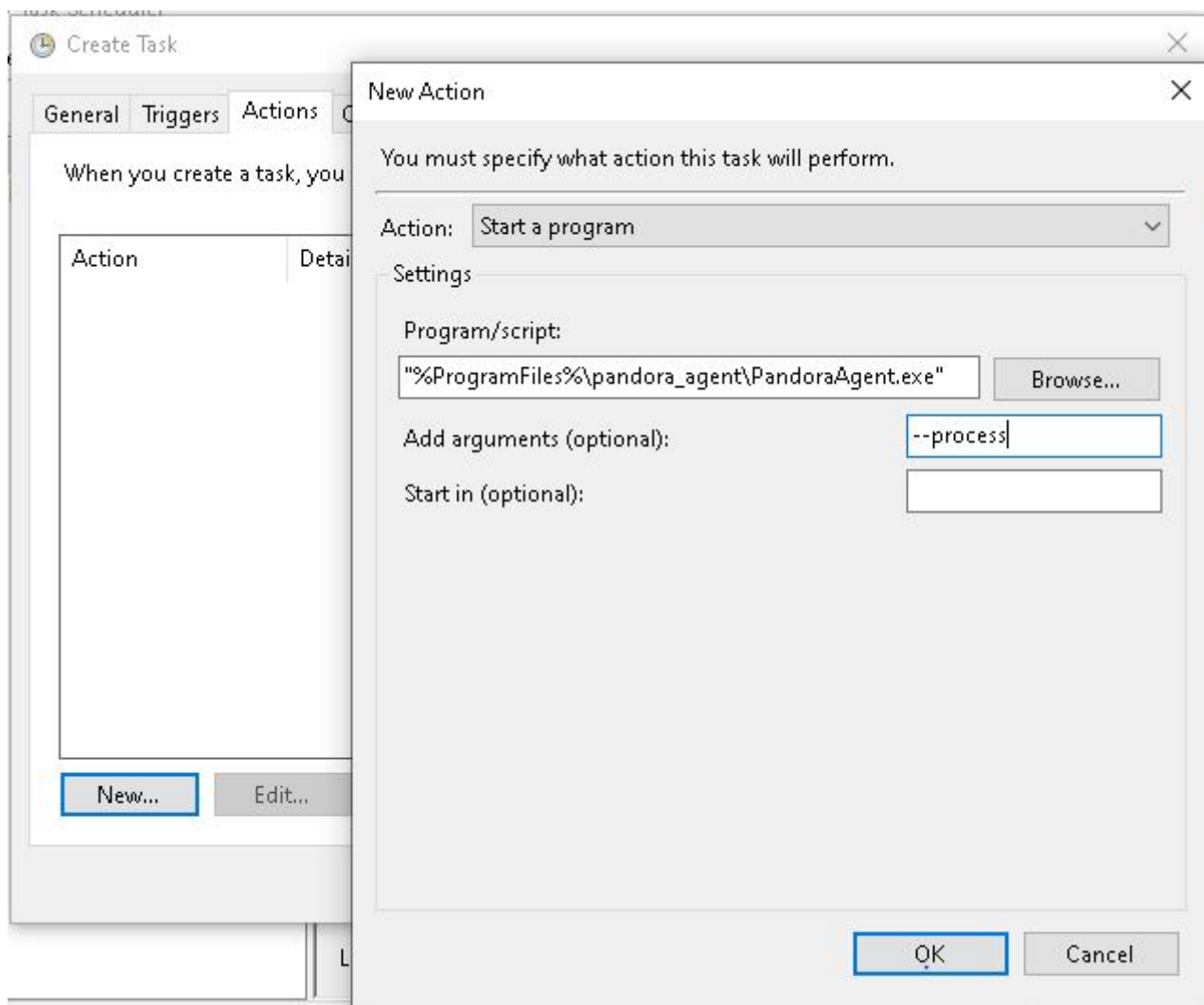
Run with highest privileges

Hidden

Configure for: Windows® 7, Windows Server™ 2008 R2

OK Cancel





Create Task

General Triggers Actions Conditions Settings

Specify the conditions that, along with the trigger, determine whether the task should run. The task will not run if any condition specified here is not true.

Idle

Start the task only if the computer is idle for: 10 minutes

Wait for idle for: 1 hour

Stop if the computer ceases to be idle

Restart if the idle state resumes

Power

Start the task only if the computer is on AC power

Stop if the computer switches to battery power

Wake the computer to run this task

Network

Start only if the following network connection is available:

Any connection

OK Cancel

Create Task

General Triggers Actions Conditions Settings

Specify additional settings that affect the behavior of the task.

Allow task to be run on demand

Run task as soon as possible after a scheduled start is missed

If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

Stop the task if it runs longer than: 3 days

If the running task does not end when requested, force it to stop

If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

Do not start a new instance

OK Cancel



De este modo garantizaremos que en este sistema Windows siempre estará corriendo el Agente de Pandora FMS en modo proceso aún cuando la máquina se reinicie, pudiendo enviar siempre la información recogida por la sonda PDR.

Sistemas con varios escritorios pueden resultar problemáticos, por lo que se recomienda siempre utilizar la configuración anteriormente descrita, en una máquina en modo escritorio con autoinicio de sesión y un único escritorio.

[Volver al Índice de Documentación Pandora FMS](#)