



Informe de Pruebas Funcionales Automatizadas

Sistema ZEE Perú

|  |  |
| --- | --- |
| Ministerios – Amazonia | Consultoría para la Implementación de una Solución para la Gestión Integral de Operaciones de las Zonas Económicas Especiales  Versión: 2.0 | Nivel de seguridad N3 – Confidencial. |

Índice

[1 | Registro de Cambios 4](#_Toc173834013)

[2 | Glosario de siglas y abreviaturas 5](#_Toc173834014)

[3 | Introducción 6](#_Toc173834015)

[3.1 Contenido del documento 6](#_Toc173834016)

[3.2 Cambios desde la última versión 6](#_Toc173834017)

[3.3 Referencias 6](#_Toc173834018)

[4 | Objetivo 7](#_Toc173834019)

[5 | Metodología 8](#_Toc173834020)

[6 | Estrategia por Ejecutar 9](#_Toc173834021)

[7 | Herramientas para las Pruebas funcionales 10](#_Toc173834022)

[7.1 Herramientas 10](#_Toc173834023)

[7.2 Supervisión del Sistema 10](#_Toc173834024)

[7.3 Testing con Selenium 11](#_Toc173834025)

[7.3.1 Pasos para la ejecucion de pruebas en Selenium IDE. 11](#_Toc173834026)

[7.3.2 Prueba de Login 14](#_Toc173834027)

[7.3.3 Prueba de Ingresos, Bandeja de Control Inicial. 14](#_Toc173834028)

[7.3.4 Prueba de Salida, Registro de Vehículo para Salida. 15](#_Toc173834029)

[7.3.5 Prueba de Aforo, Registro de Control de Inventario. 16](#_Toc173834030)

[7.3.6 Prueba de Balanza, Registro de Pesaje. 17](#_Toc173834031)

[7.3.7 Prueba de Contrato, Registro de Inversión de Contrato. 18](#_Toc173834032)

[7.3.8 Prueba de Tesoreria, Registro de Tipo de Cambio. 19](#_Toc173834033)

[7.3.9 Prueba de Mensajería, Crear Notificación 20](#_Toc173834034)

[7.3.10 Prueba Control Acceso : Bandeja Control Acceso 21](#_Toc173834035)

[8 | Resultados de las Pruebas 23](#_Toc173834036)

[9 | Conclusiones 24](#_Toc173834037)

[10 | Recomendaciones 25](#_Toc173834038)

**Índice de figuras**

[Imagen 1: Flujo de la metodología usada 8](#_Toc173833981)

[Imagen 2: Instalación de la extensión 11](#_Toc173833982)

[Imagen 3: Visualización de Selenium IDE en las extensiones del navegador 11](#_Toc173833983)

[Imagen 4: Selenium IDE, abrir archivo test 12](#_Toc173833984)

[Imagen 5:Selenium IDE, elegir archivo test 12](#_Toc173833985)

[Imagen 6: Selenium IDE, iniciar test 13](#_Toc173833986)

[Imagen 7:Selenium IDE test finalizado 13](#_Toc173833987)

[Imagen 8: Spect de Login 14](#_Toc173833988)

[Imagen 9: Log de testing de Login 14](#_Toc173833989)

[Imagen 10: Spect de Ingresos, bandeja de control inicial 14](#_Toc173833990)

[Imagen 11: Log de testing de Ingresos, bandeja de control inicial 15](#_Toc173833991)

[Imagen 12: Spect de Salida, registro de vehículo para salida 15](#_Toc173833992)

[Imagen 13: Log de testing de Salida, registro de vehículo para salida 15](#_Toc173833993)

[Imagen 14: finalización del script correctamente. 16](#_Toc173833994)

[Imagen 15: Spect de Aforo, registro de control de inventario 16](#_Toc173833995)

[Imagen 16: Log de testing de Aforo, registro de control de inventario 16](#_Toc173833996)

[Imagen 17: script ejecutado correctamente 17](#_Toc173833997)

[Imagen 18: Spect de Balanza, gestión de balanzas 17](#_Toc173833998)

[Imagen 19: Log de testing de Balanza, gestión de balanzas 17](#_Toc173833999)

[Imagen 20: script ejecutado correctamente 18](#_Toc173834000)

[Imagen 21:Spect de Contrato, registro de inversión de contrato 18](#_Toc173834001)

[Imagen 22: Log de testing de Contrato, registro de inversión de contrato 18](#_Toc173834002)

[Imagen 23: script ejecutado correctamente. 19](#_Toc173834003)

[Imagen 24: Spect de Tesorería, registro de tipo de cambio 19](#_Toc173834004)

[Imagen 25: Log de testing de Tesorería, registro de tipo de cambio 19](#_Toc173834005)

[Imagen 26: finalización del script tipo de cambio correctamente. 20](#_Toc173834006)

[Imagen 27 :Spect de Mensajería, crear notificación. 20](#_Toc173834007)

[Imagen 28: Log de testing de Mensajería, crear notificación. 20](#_Toc173834008)

[Imagen 29: finalización correcta del script 21](#_Toc173834009)

[Imagen 30: script de usuario, crear usuario 21](#_Toc173834010)

[Imagen 31: Log de testing de usuarios, crear usuario. 21](#_Toc173834011)

[Imagen 32: finalización del script para creación usuario. 22](#_Toc173834012)

# Registro de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Revisado | Resumen de los cambios producidos |
| v1.0 | 01/05/2023 | Marco Risco | Primera versión del documento |
| V2.0 | **16/01/2024** | Marco Risco | Se agregaron las pruebas funcionales con respecto a los flujos principales del sistema ZEE Perú. |
| V3.0 | 31/07/2024 | Abel Chingo Tello | Actualización de documento. |

# Glosario de siglas y abreviaturas

|  |  |
| --- | --- |
| **ZEE** | Zonas Económicas Especiales |
| **Sistema ZEE Perú** | Denominación de la solución para la gestión integral de operaciones de las Zonas Económicas Especiales. |

# Introducción

## Contenido del documento

El presente informe proporciona detalles sobre las pruebas funcionales automatizadas realizadas en el sistema ZEE Perú utilizando la herramienta Selenium. Estas pruebas tienen como objetivo verificar el correcto funcionamiento de las funcionalidades del sistema de manera automatizada, garantizando la calidad del software y acelerando el proceso de prueba. A continuación, se presenta un resumen de las pruebas realizadas y sus resultados.

## Cambios desde la última versión

Se realizaron cambios en el punto 8.3 Testing con Selenium, agregando pruebas funcionales automatizadas con respecto a las distintas funcionalidades del sistema ZEE Perú.

## Referencias

1. Términos de Referencia de la Solución para la Gestión Integral de Operaciones de las Zonas Económicas Especiales.
2. Propuesta para la Implementación de una Solución para la Gestión Integral de Operaciones de las Zonas Económicas Especiales.
3. Entregable 2. Documento de Análisis y Diseño del Sistema ZEE Perú
4. Entregable 3. Arquitectura Tecnológica
5. Arquitectura Patrón de Aplicaciones – Diseño Proyecto VUCE 2.0
6. Arquitectura VUCE 2.0 v2.1 (pptx)

# Objetivo

1. Automatizar las pruebas funcionales del sistema ZEE Perú utilizando la herramienta Selenium.
2. Verificar el correcto funcionamiento de las funcionalidades clave del sistema mediante pruebas automatizadas.
3. Asegurar la consistencia y la calidad de las pruebas funcionales en cada iteración del desarrollo.
4. Detectar y reportar de manera temprana posibles fallos o desviaciones en el comportamiento esperado del sistema.

# Metodología

Las pruebas funcionales automatizadas con Selenium se enfocaron en las siguientes áreas del sistema ZEE Perú.

En la siguiente gráfica se detalla la metodología estándar utilizada llevada a cabo las tareas relacionadas a las pruebas funcionales.

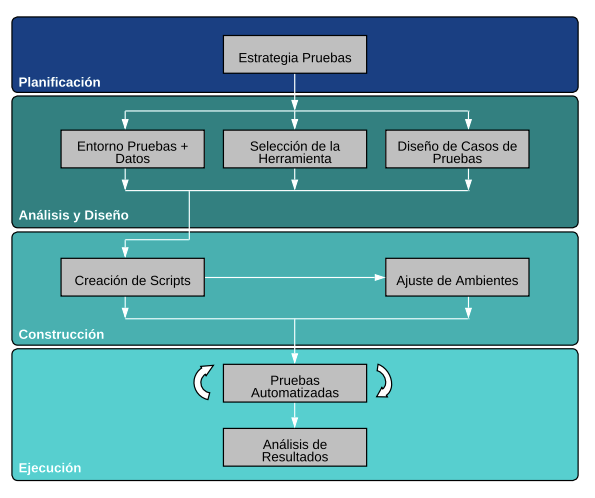


Imagen 1: Flujo de la metodología usada

Es Importante indicar que el bloque “Ajuste de Ambientes” no se realizó como parte de las actividades del presente informe.

# Estrategia por Ejecutar

A continuación, se describen las estrategias de pruebas automatizadas, definiendo las técnicas a usar, los criterios de éxito a considerar y el entorno de pruebas a utilizar, entre otros aspectos que se detallarán a continuación:

1. Se seleccionaron las funcionalidades clave del sistema para ser incluidas en las pruebas automatizadas.
2. Se utilizó Selenium como herramienta principal para la automatización de las pruebas.
3. Se ejecutaron las pruebas en el navegador web Google Chrome, para garantizar la compatibilidad.

# Herramientas para las Pruebas funcionales

## Herramientas

A continuación, se describen cada una de las herramientas de soporte a las pruebas funcionales atomizadas; las cuales se utilizarán para la ejecución de la prueba y la administración de sus resultados.

| **Nivel** | **Herramienta** | **Descripción** |
| --- | --- | --- |
| Pruebas funcionales automatizadas |  | El objetivo principal de la aplicación es medir:   * Automatizar las pruebas funcionales del sistema ZEE Perú utilizando la herramienta Selenium versión 3.17.2. * Verificar el correcto funcionamiento de las funcionalidades clave del sistema mediante pruebas automatizadas. |

Tabla 3. Herramientas de soporte

## Supervisión del Sistema

Se realizará el control y seguimiento del cumplimiento de las pruebas planificadas, para lo que se utilizarán las siguientes herramientas:

| **Herramienta** | **Aporte** |
| --- | --- |
|  | Las pruebas automatizadas se ejecutarán mediante Selenium, donde se monitorearán las respuestas obtenidas de dichas pruebas. |

Tabla 4. Herramientas de supervisión del sistema.

## Testing con Selenium

### Pasos para la ejecucion de pruebas en Selenium IDE.

1. Instalamos es nuestro navegador Chrome la extensión de Selenium IDE del siguiente enlace: <https://chromewebstore.google.com/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?hl=es&utm_source=ext_sidebar> y añadir al navegador.

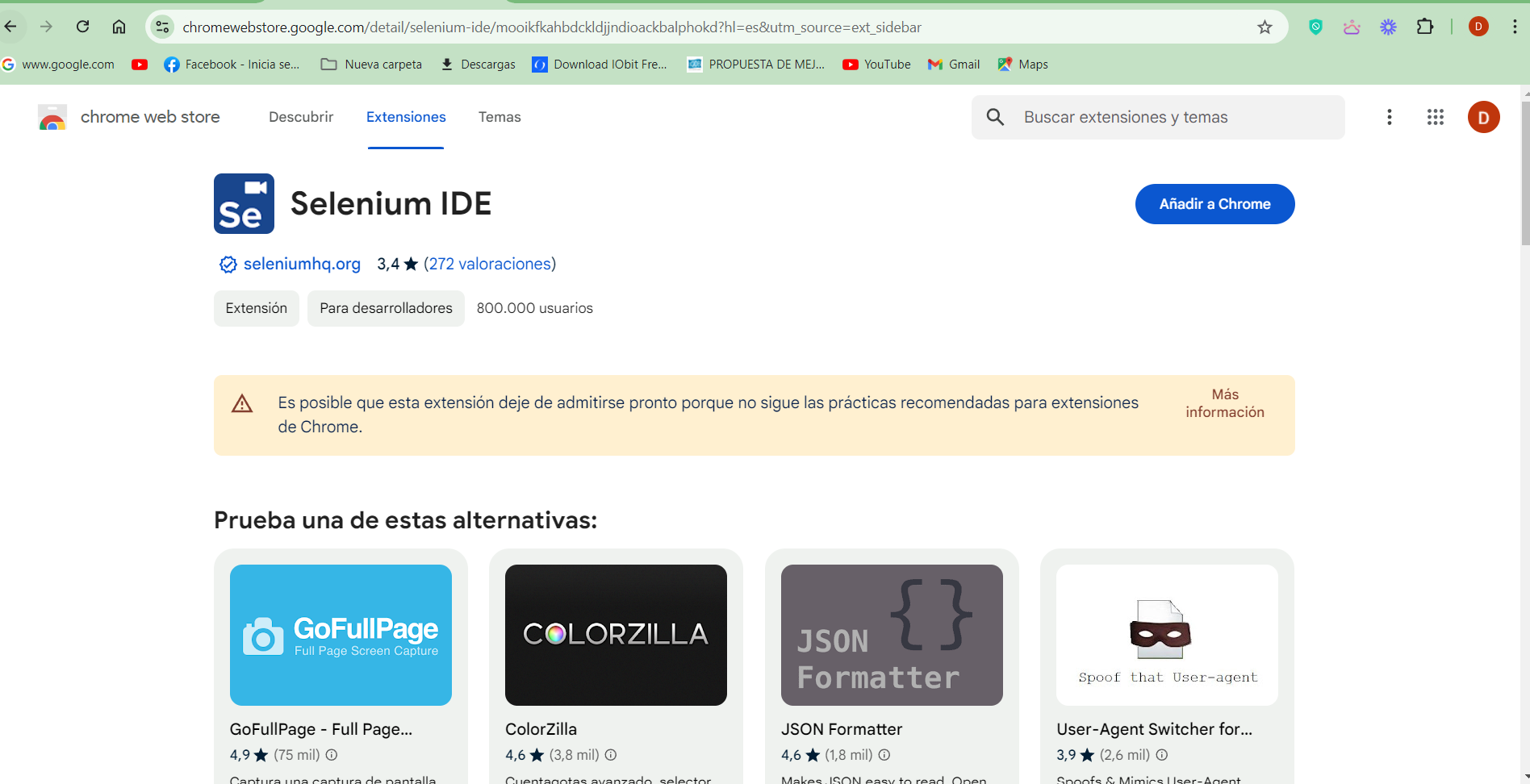


Imagen 2: Instalación de la extensión

1. Abrimos desde las extensiones Selenium IDE.

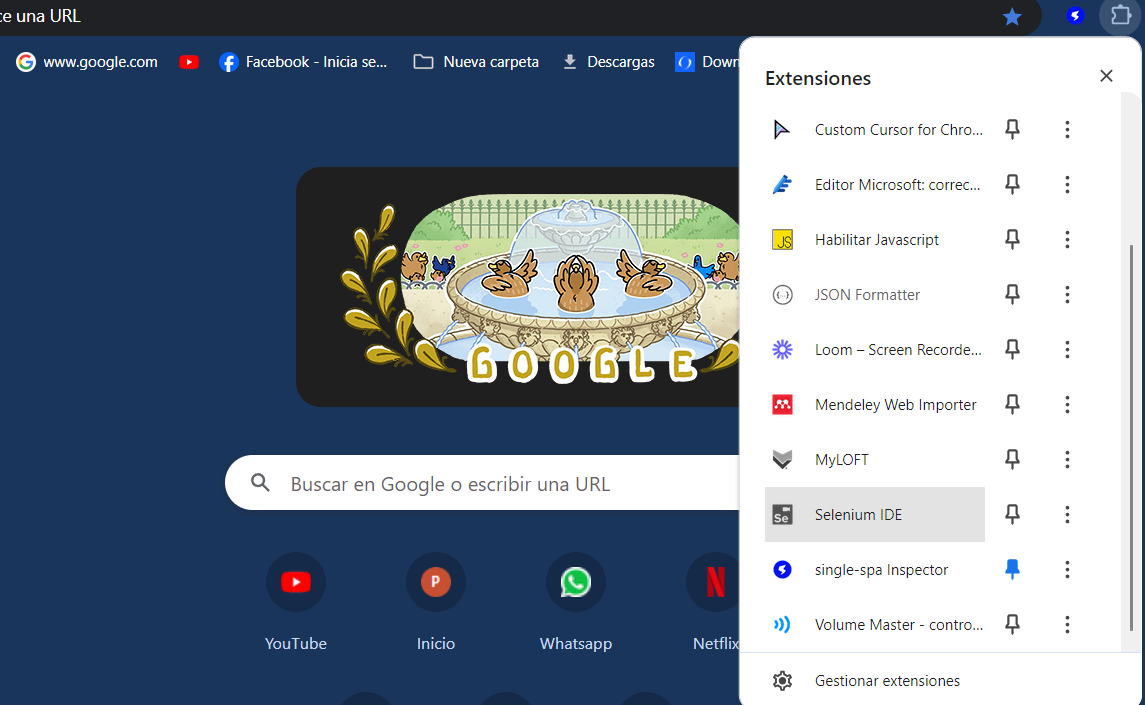


Imagen 3: Visualización de Selenium IDE en las extensiones del navegador

1. Abrimos Selenium IDE y elegimos la opción ‘Open an existing project’.

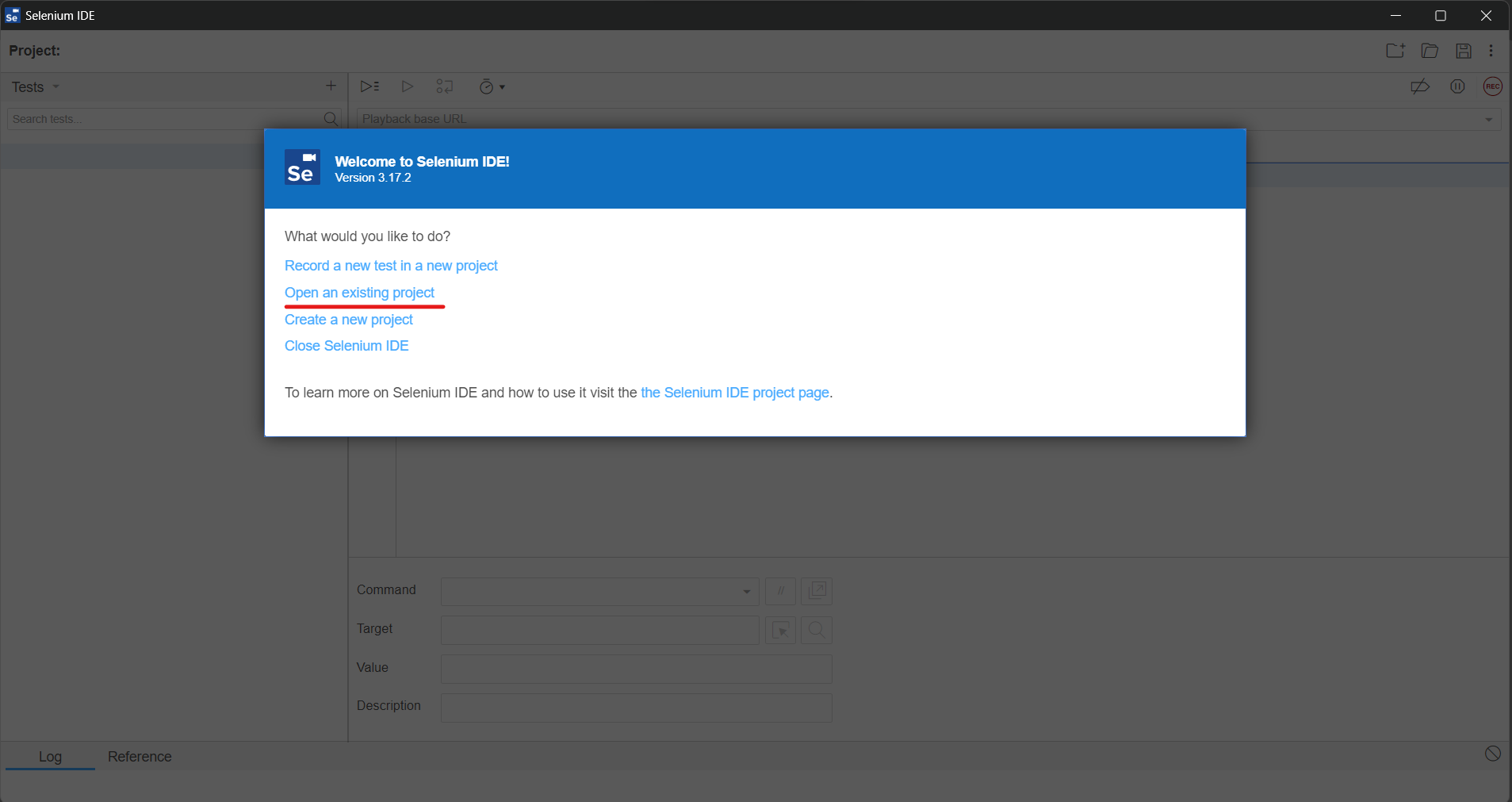


Imagen 4: Selenium IDE, abrir archivo test

1. Elegimos el archivo el cual contiene los test de Selenium.

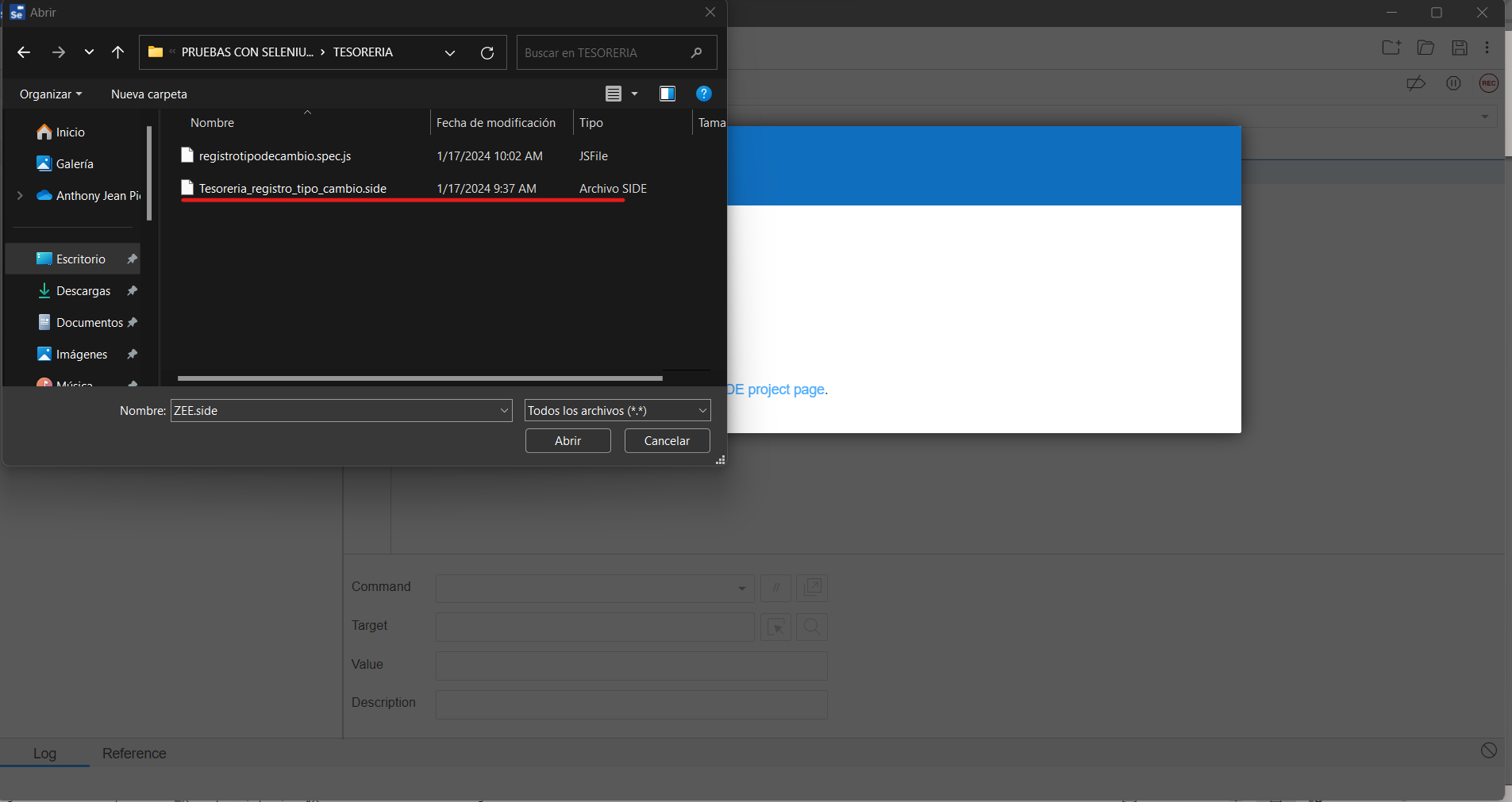


Imagen 5:Selenium IDE, elegir archivo test

1. Elegimos la opción de Run current test, y esperamos que se ejecute el test.

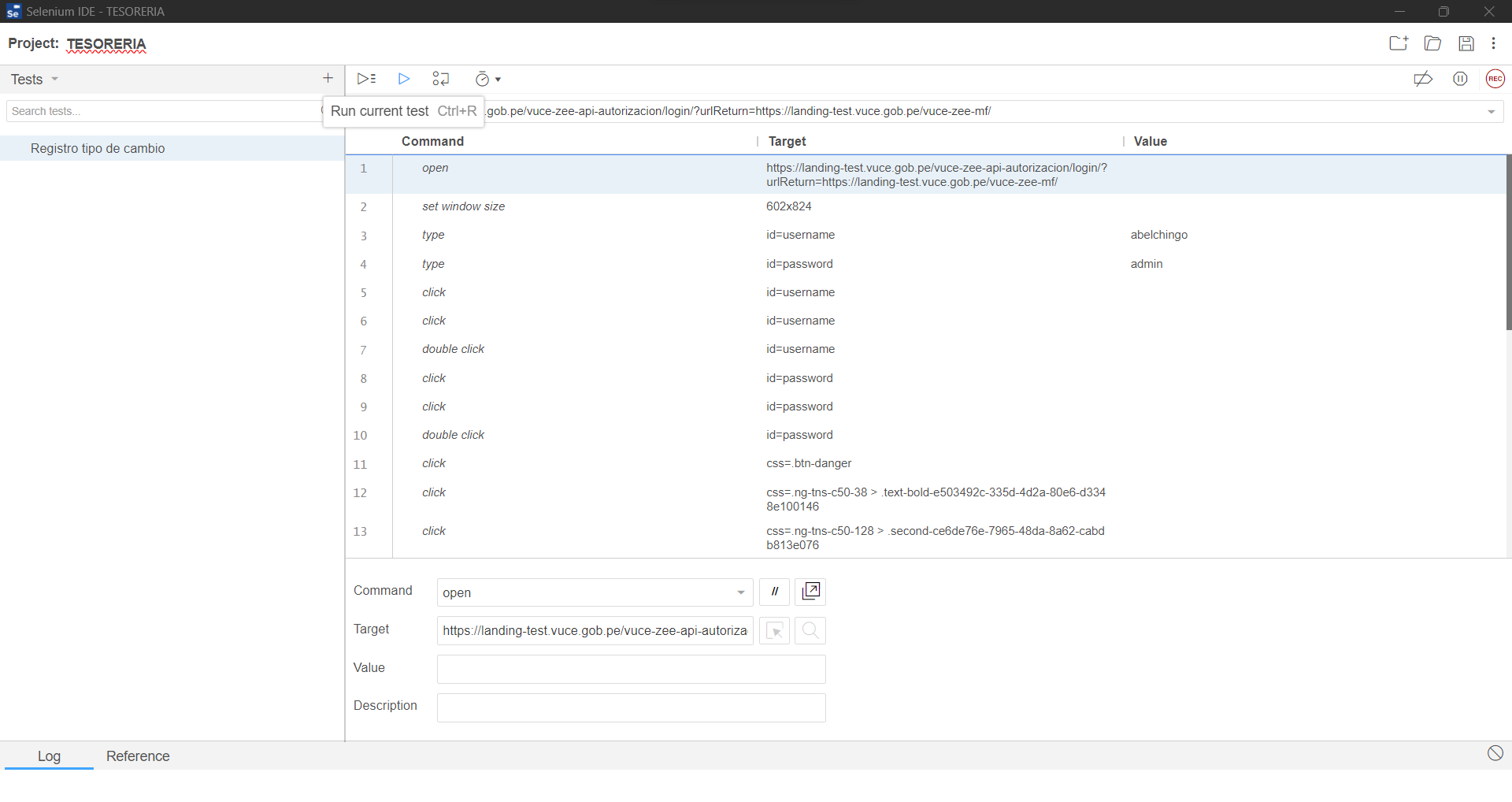


Imagen 6: Selenium IDE, iniciar test

1. Al fin mostrara que el test se ejecutó correctamente.

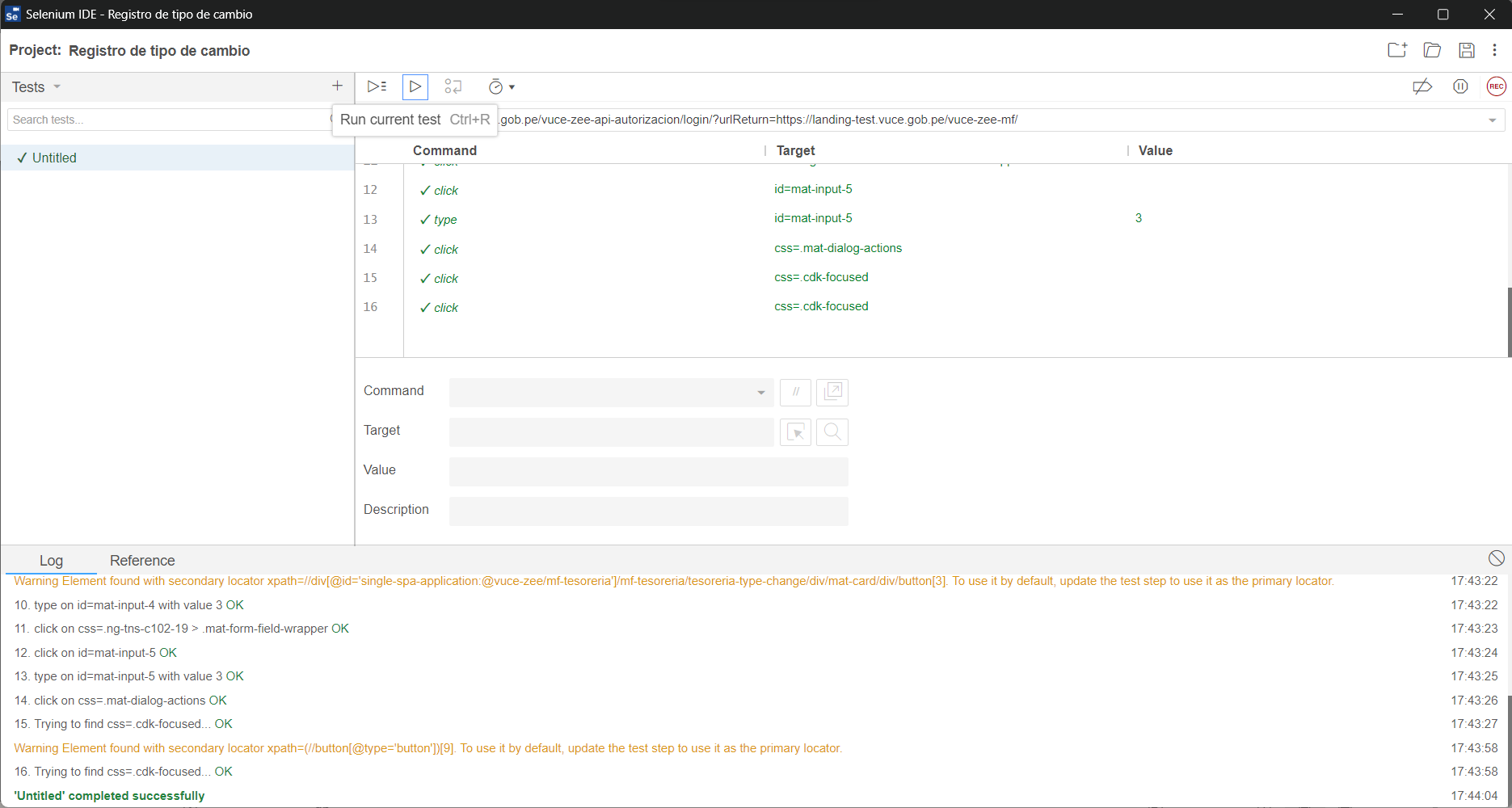


Imagen 7:Selenium IDE test finalizado

### Prueba de Login

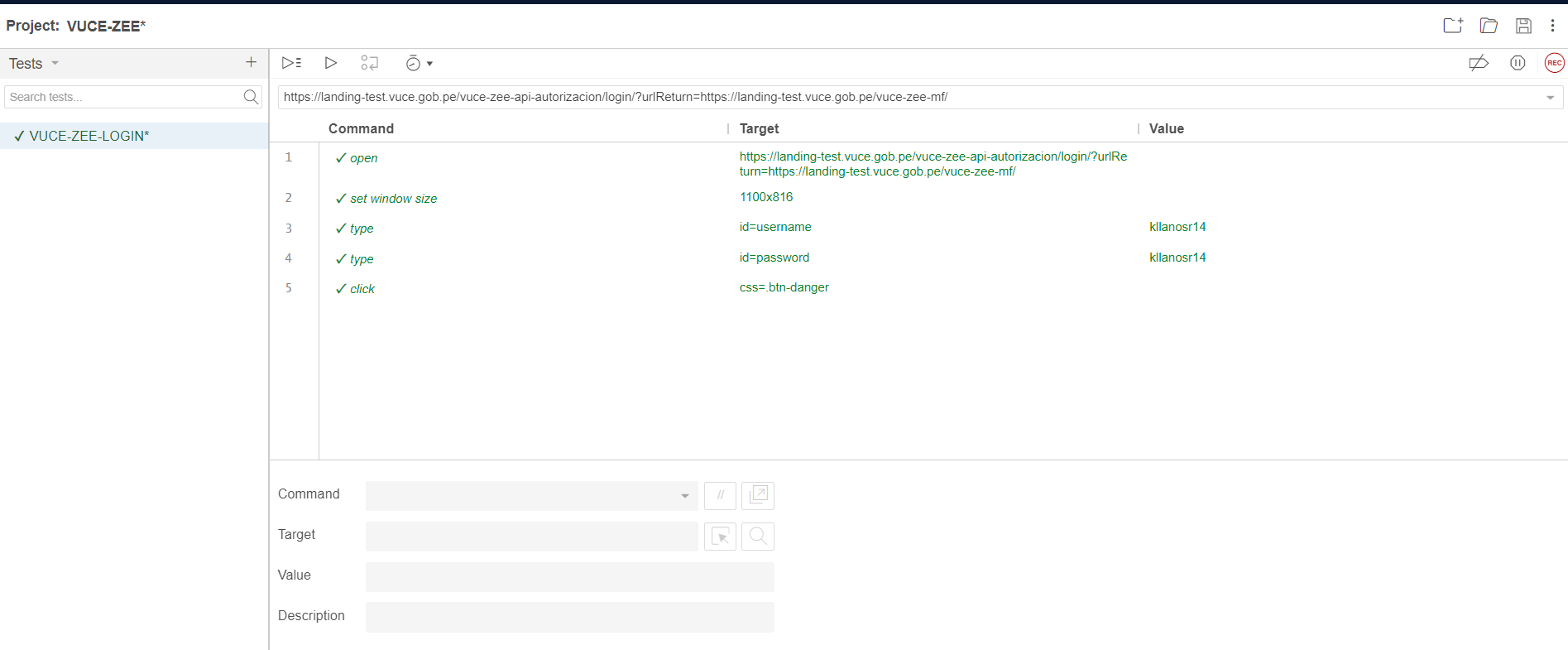


Imagen 8: Spect de Login

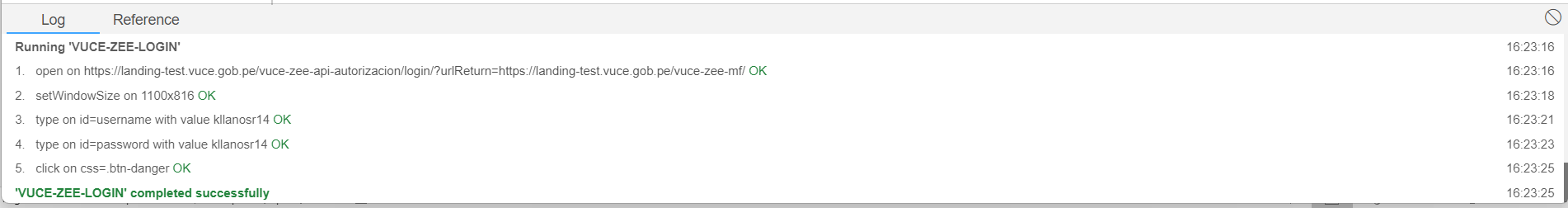


Imagen 9: Log de testing de Login

### Prueba de Ingresos, Bandeja de Control Inicial.

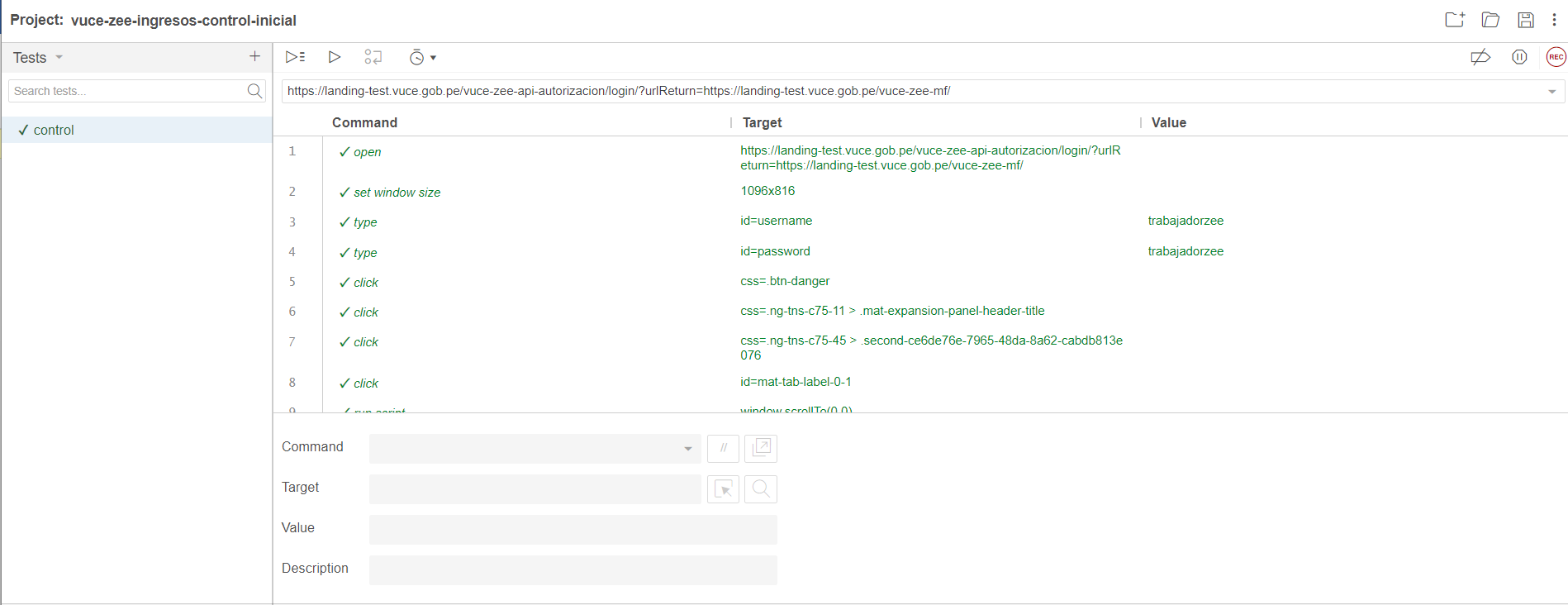


Imagen 10: Spect de Ingresos, bandeja de control inicial

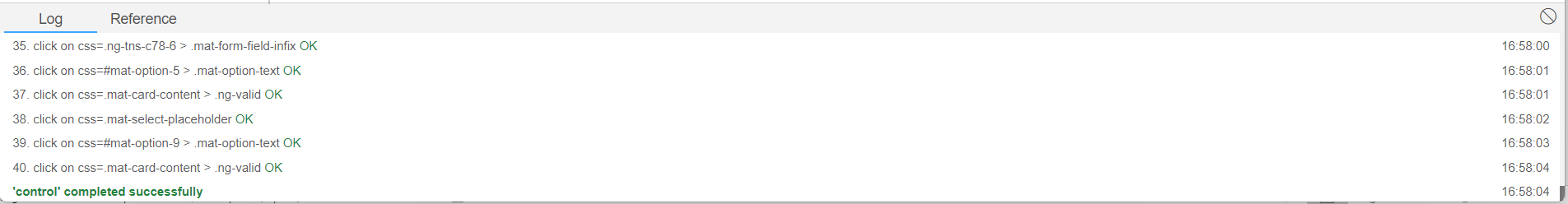


Imagen 11: Log de testing de Ingresos, bandeja de control inicial

### Prueba de Salida, Registro de Vehículo para Salida.

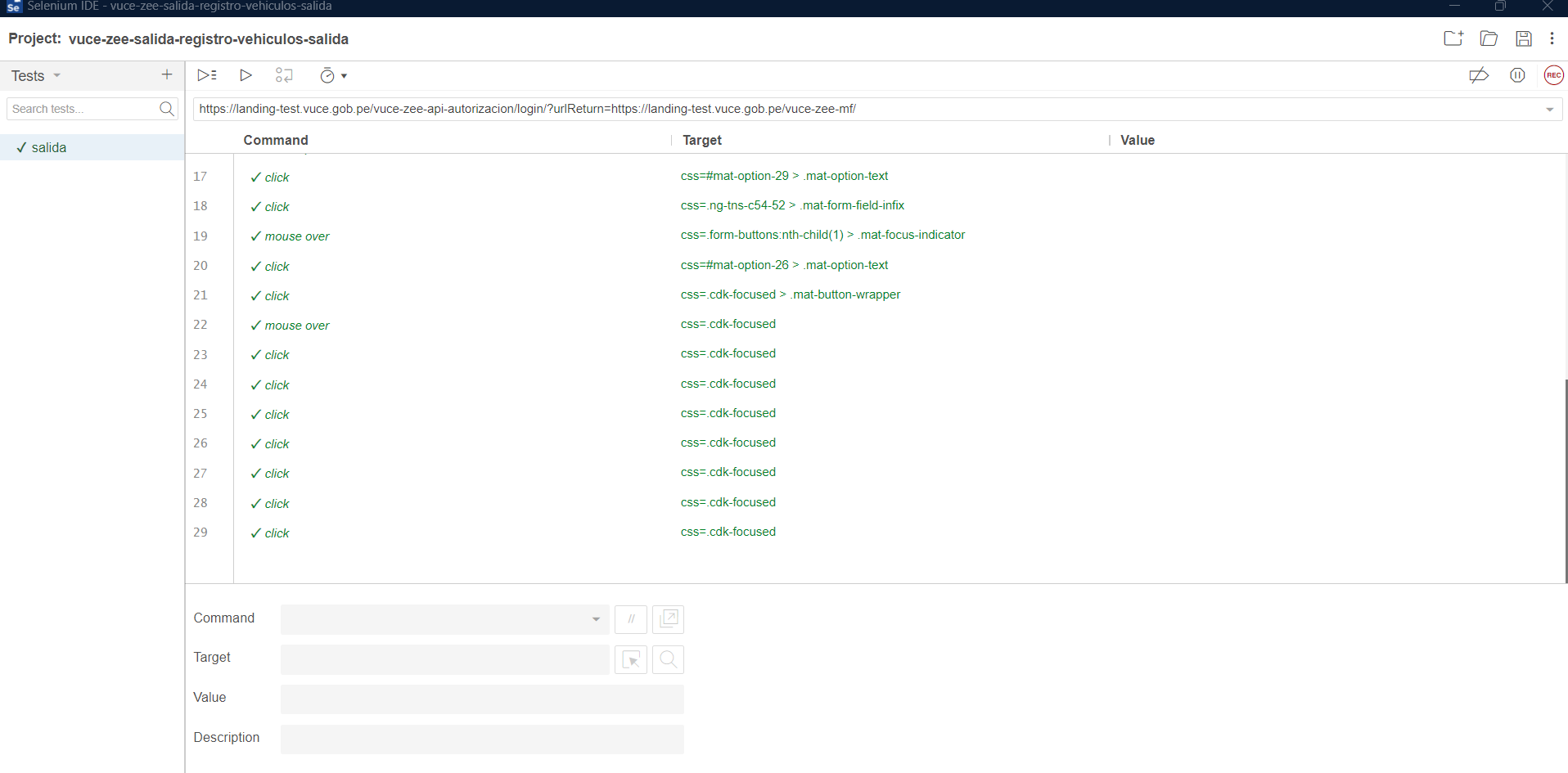


Imagen 12: Spect de Salida, registro de vehículo para salida

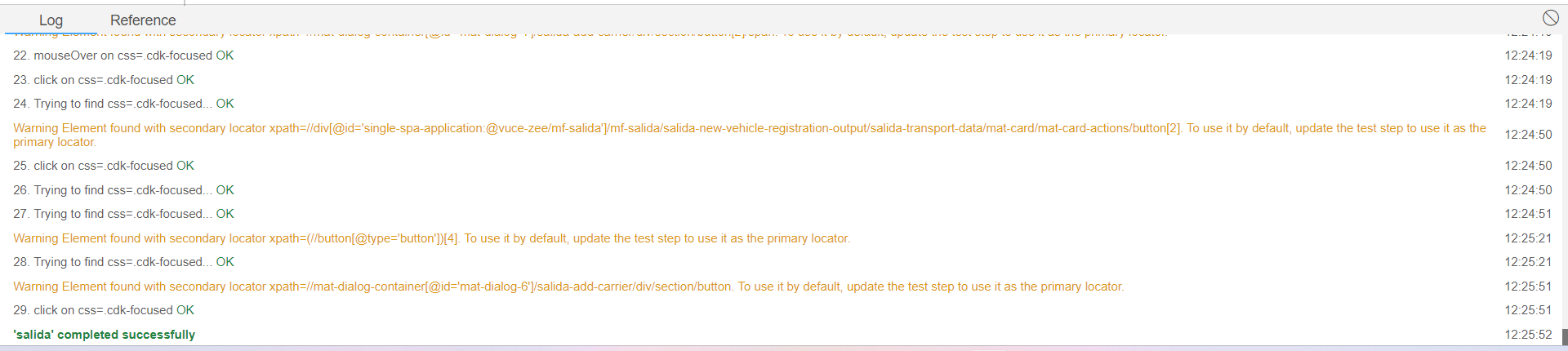


Imagen 13: Log de testing de Salida, registro de vehículo para salida

* En la siguiente imagen se comprueba que el script ejecutado fue correcto.

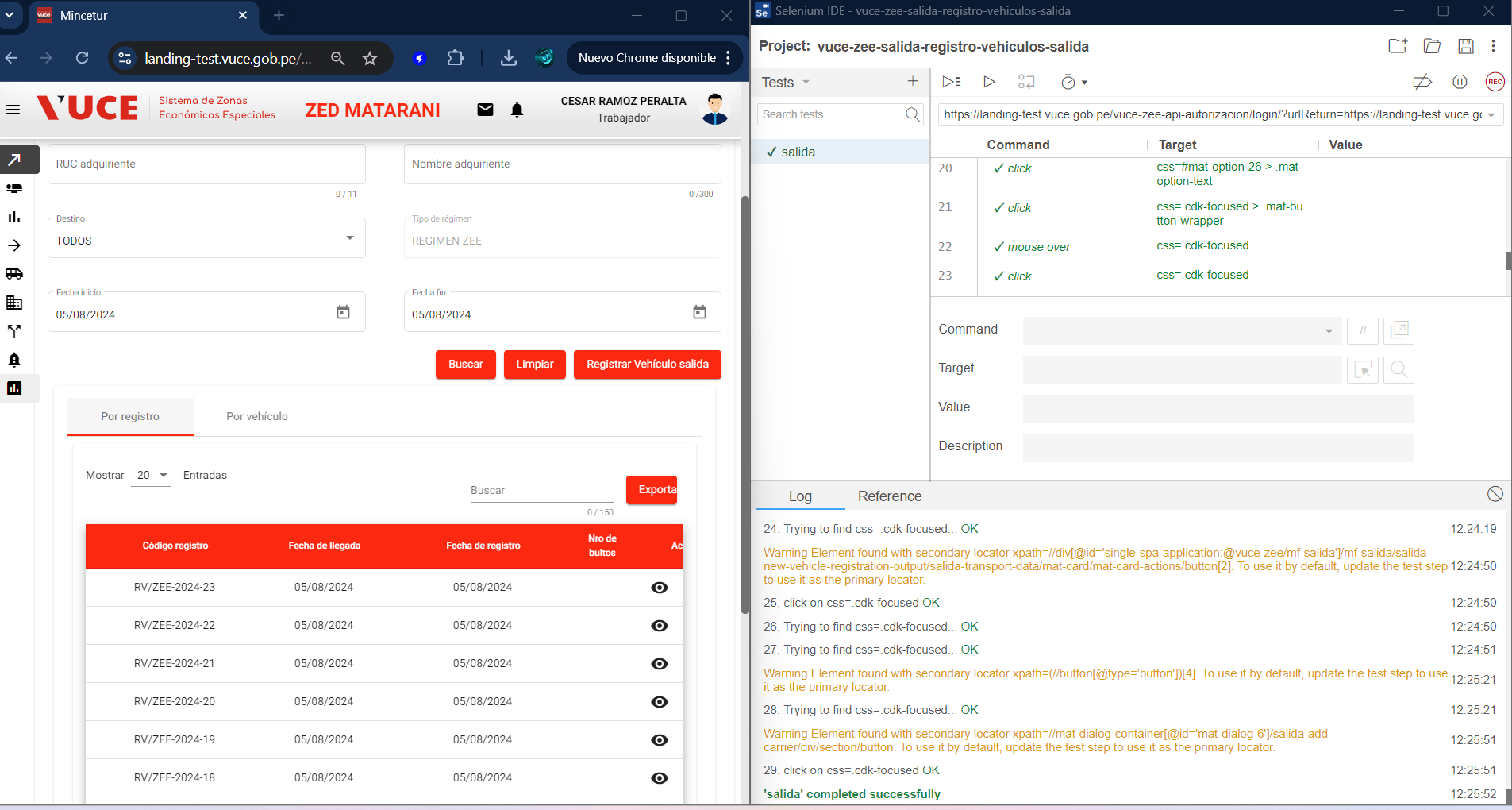


Imagen 14: finalización del script correctamente.

### Prueba de Aforo, Registro de Control de Inventario.

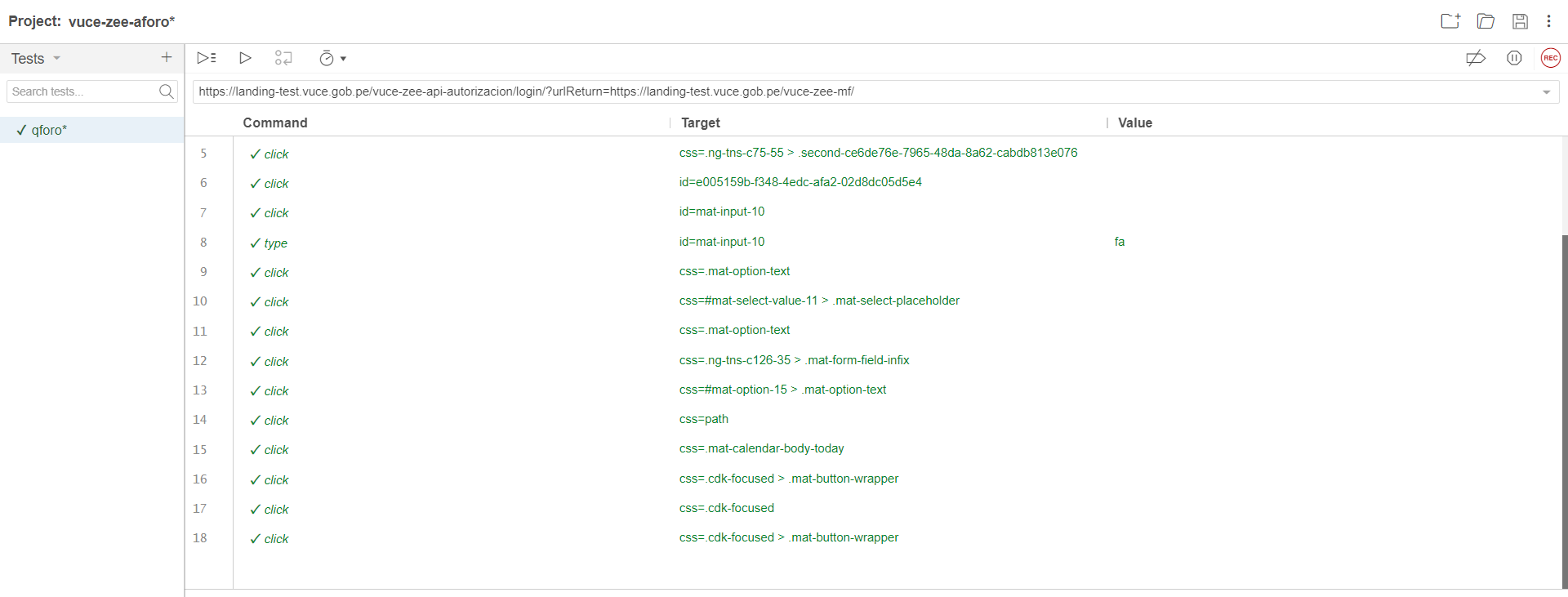


Imagen 15: Spect de Aforo, registro de control de inventario

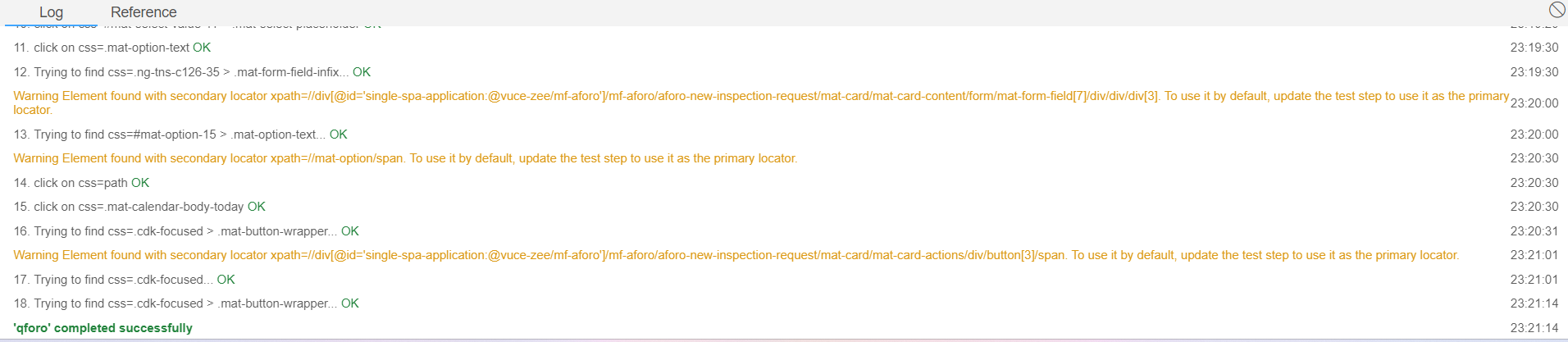


Imagen 16: Log de testing de Aforo, registro de control de inventario

* La siguiente imagen comprueba que el script ejecutado es correcto.

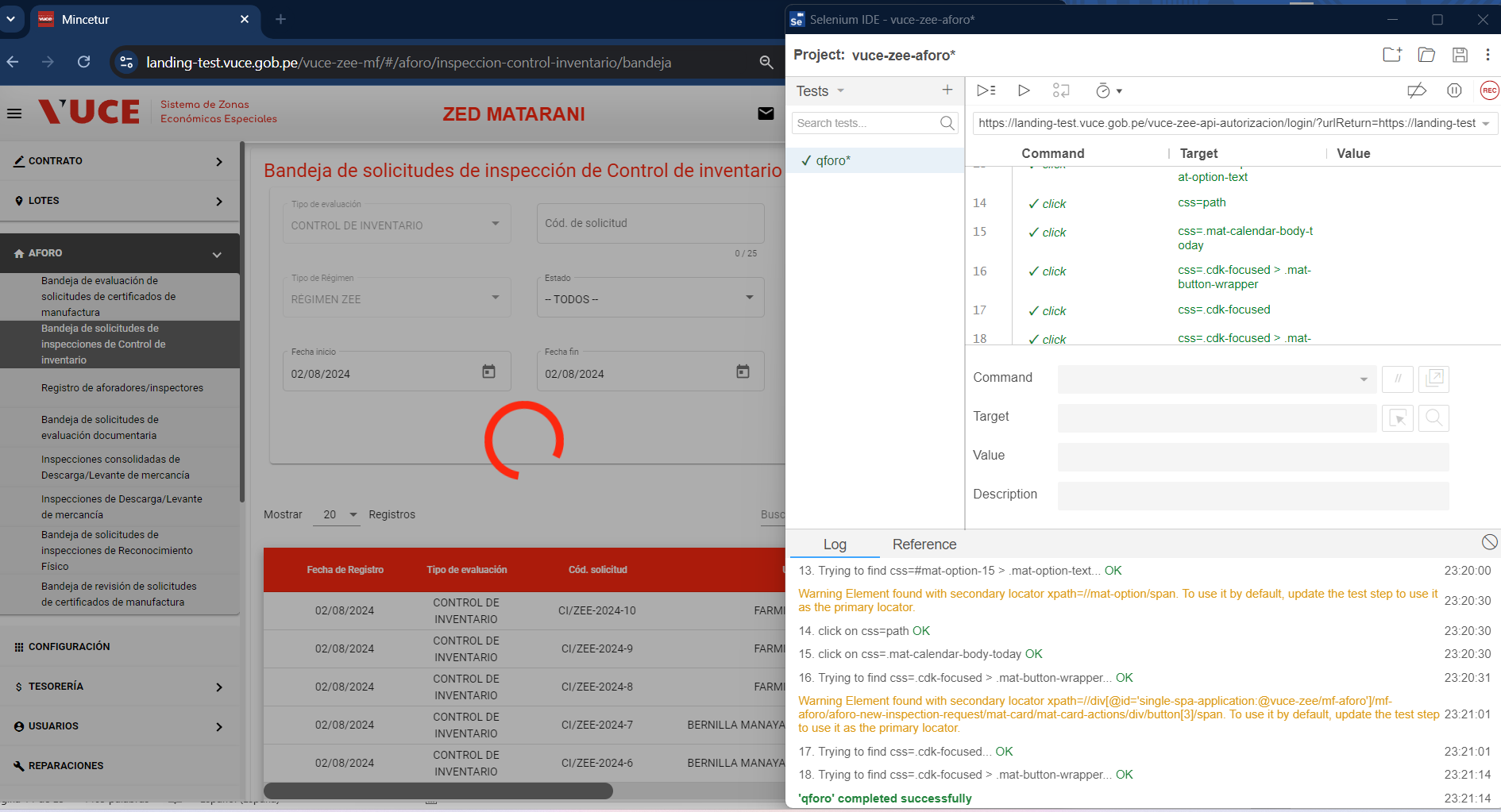


Imagen 17: script ejecutado correctamente

### Prueba de Balanza, Registro de Pesaje.

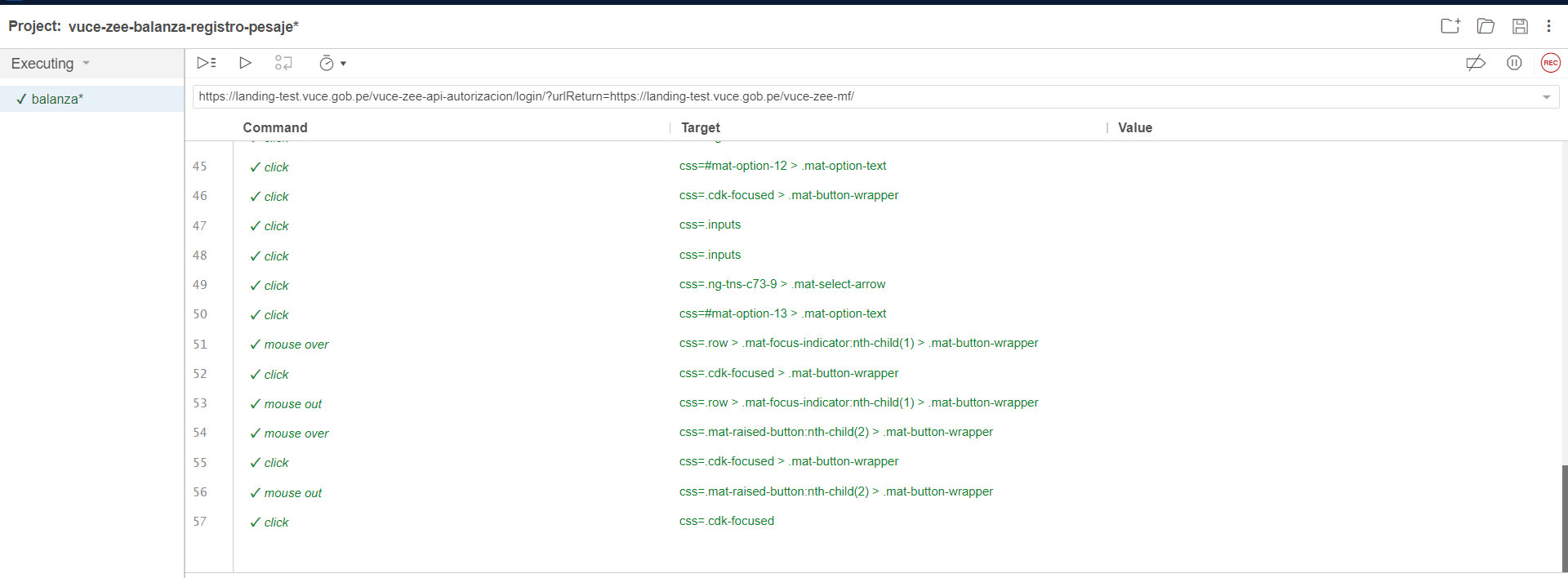


Imagen 18: Spect de Balanza, gestión de balanzas

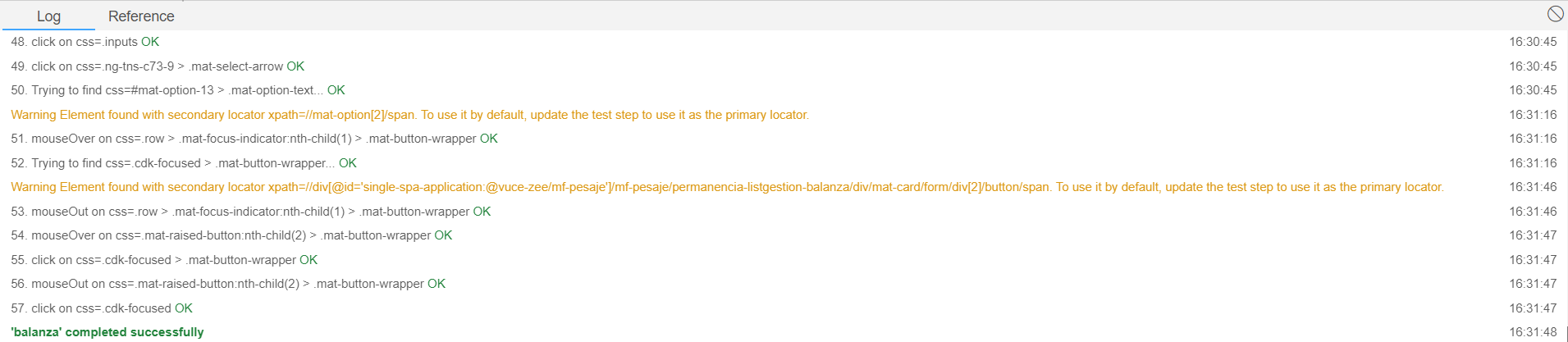


Imagen 19: Log de testing de Balanza, gestión de balanzas

* La siguiente imagen comprueba que el flujo del script es correcto.

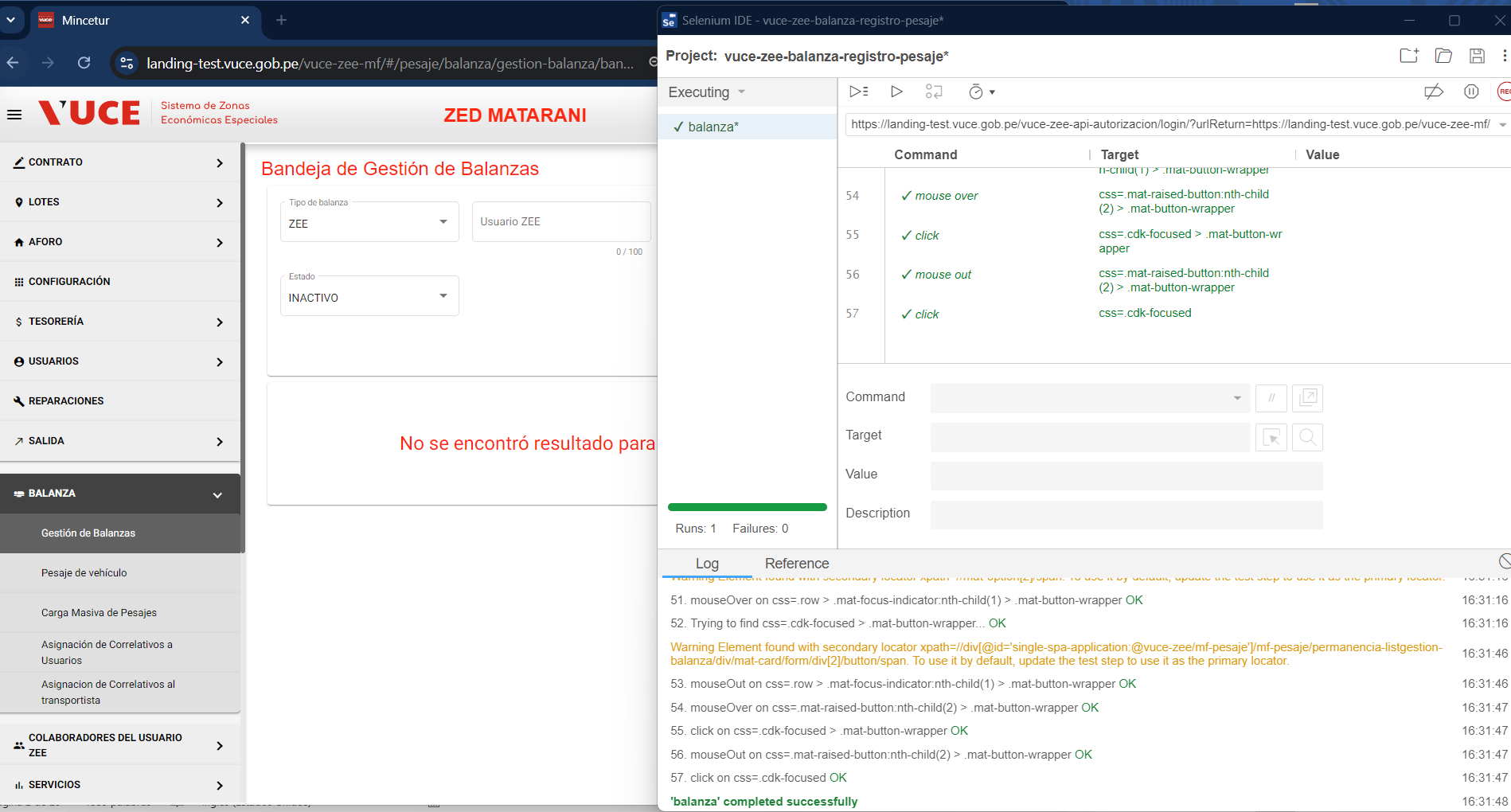


Imagen 20: script ejecutado correctamente

### Prueba de Contrato, Registro de Inversión de Contrato.

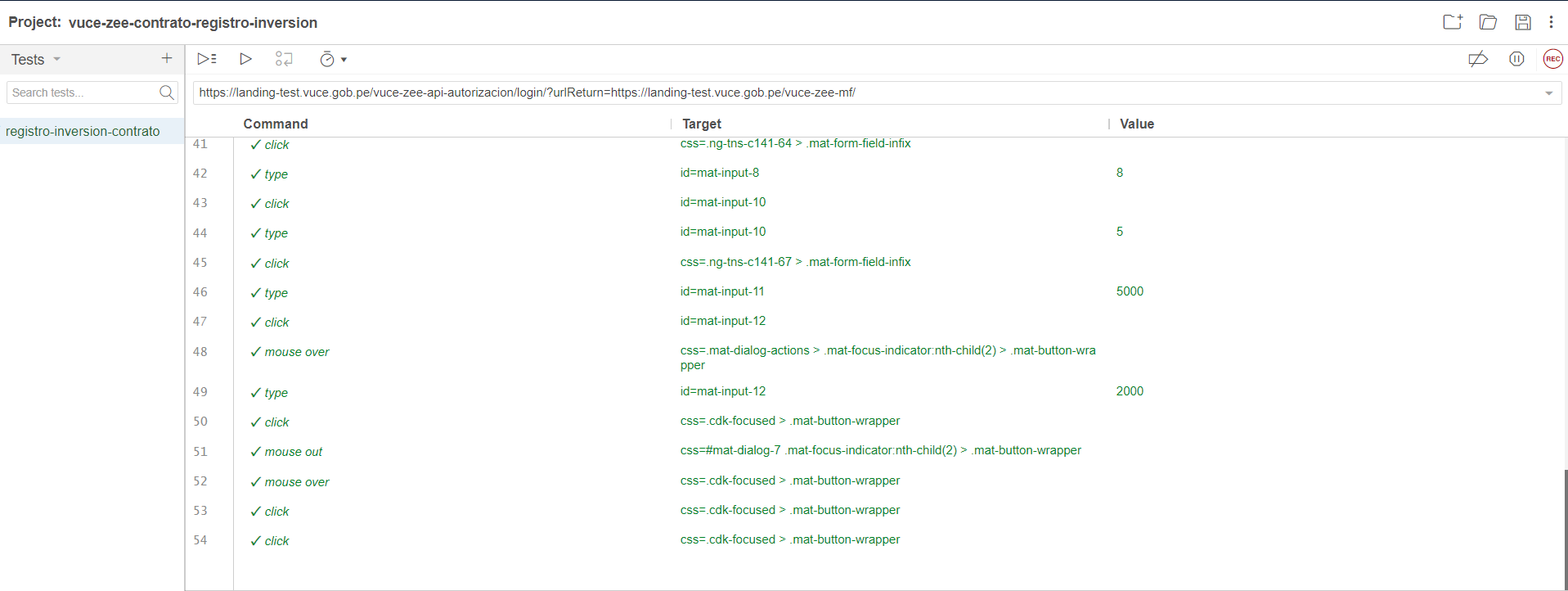


Imagen 21:Spect de Contrato, registro de inversión de contrato

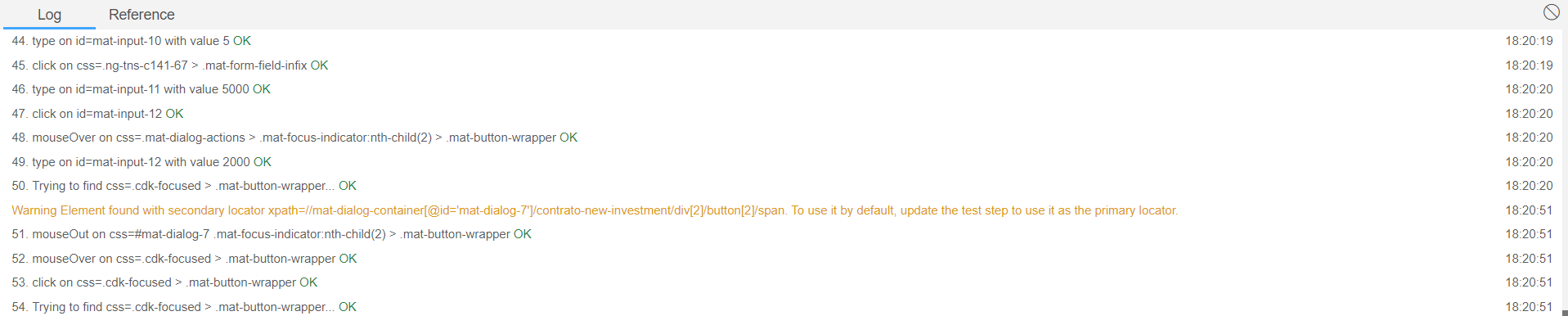


Imagen 22: Log de testing de Contrato, registro de inversión de contrato

Es necesario que ejecute el siguiente script en base de datos, para poder volver a ejecutar el script de Selenium sin ningún problema.

delete from vuce\_zee.vecr\_inve where vein\_inve\_id\_cont\_fk='8c0e8201-6de1-435e-993b-8e736807eb64' and vein\_inve\_month=2 and vein\_inve\_year=2024;

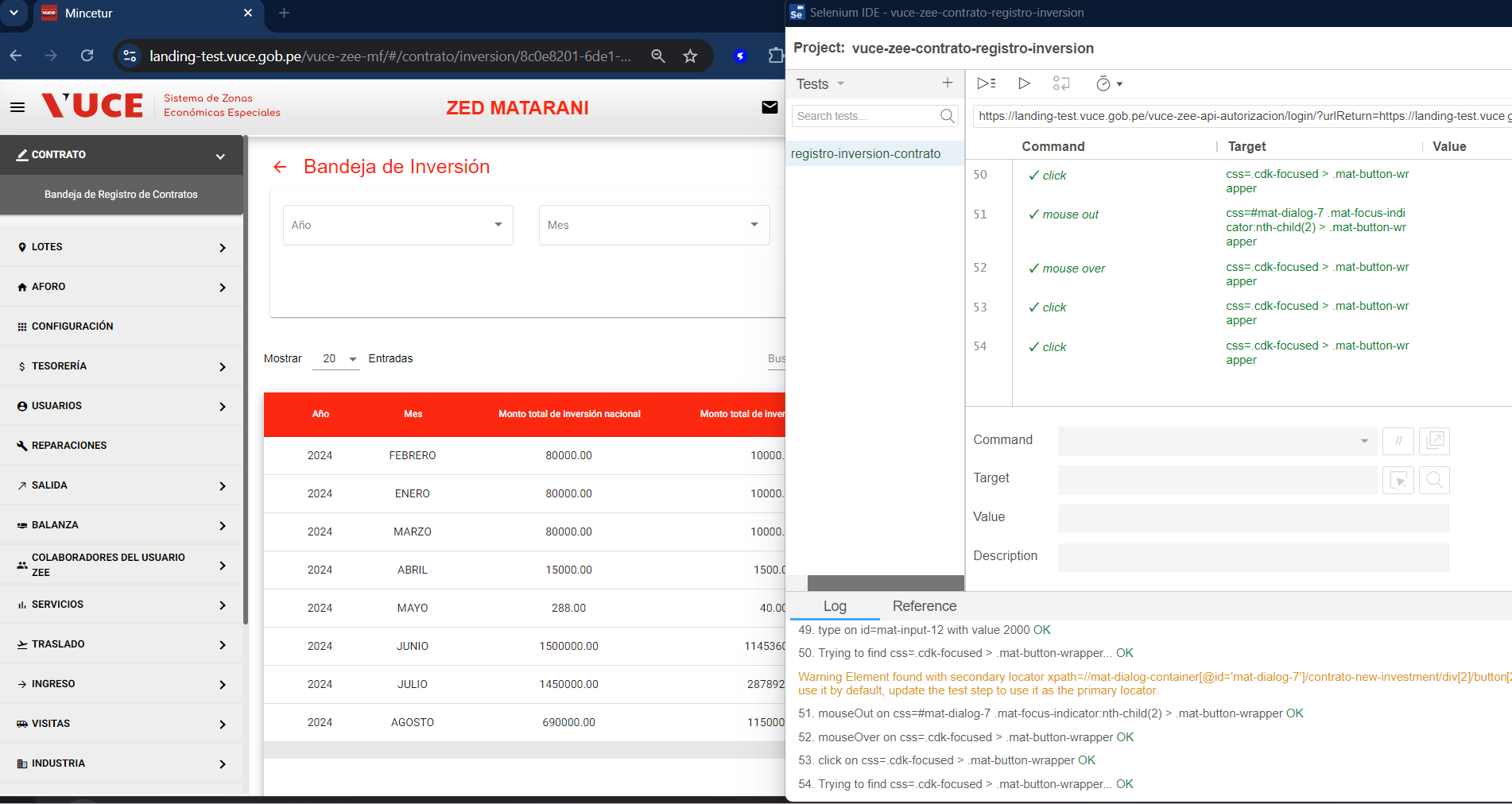


Imagen 23: script ejecutado correctamente.

### Prueba de Tesoreria, Registro de Tipo de Cambio.

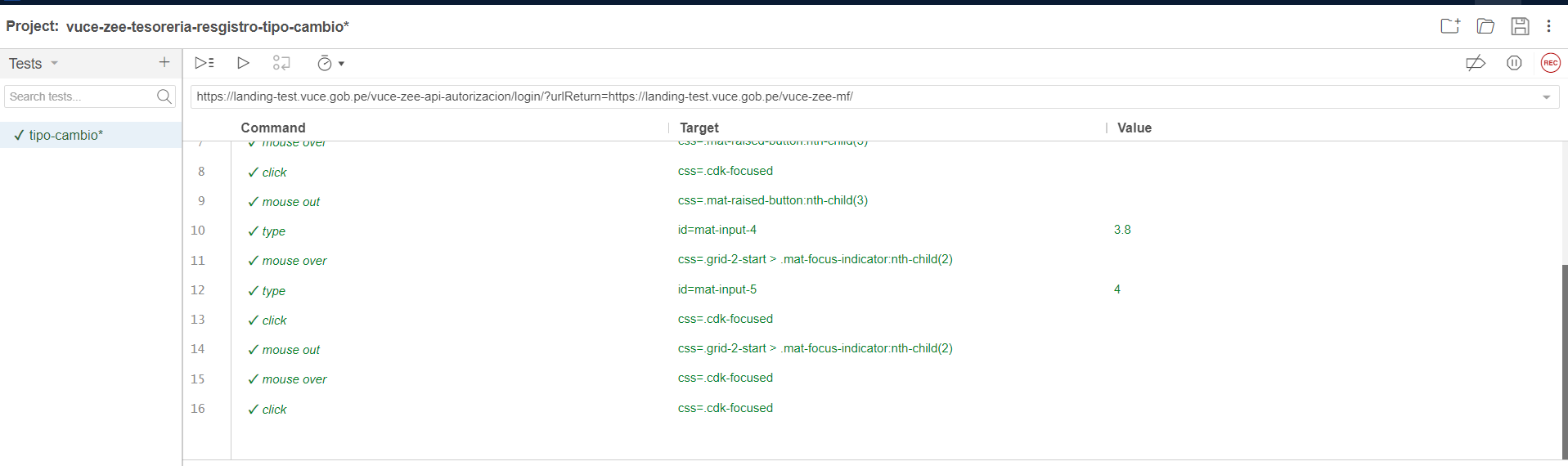


Imagen 24: Spect de Tesorería, registro de tipo de cambio

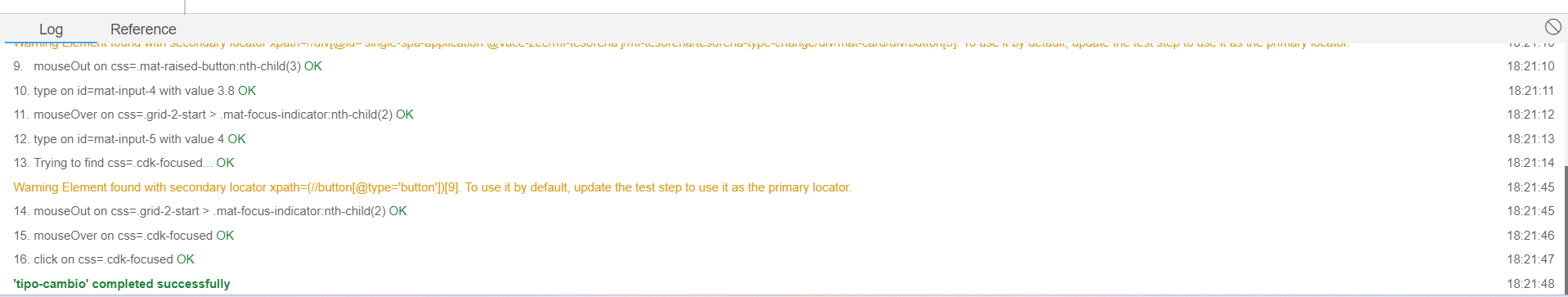


Imagen 25: Log de testing de Tesorería, registro de tipo de cambio

* La siguiente imagen comprueba que el script fue ejecutado correctamente.

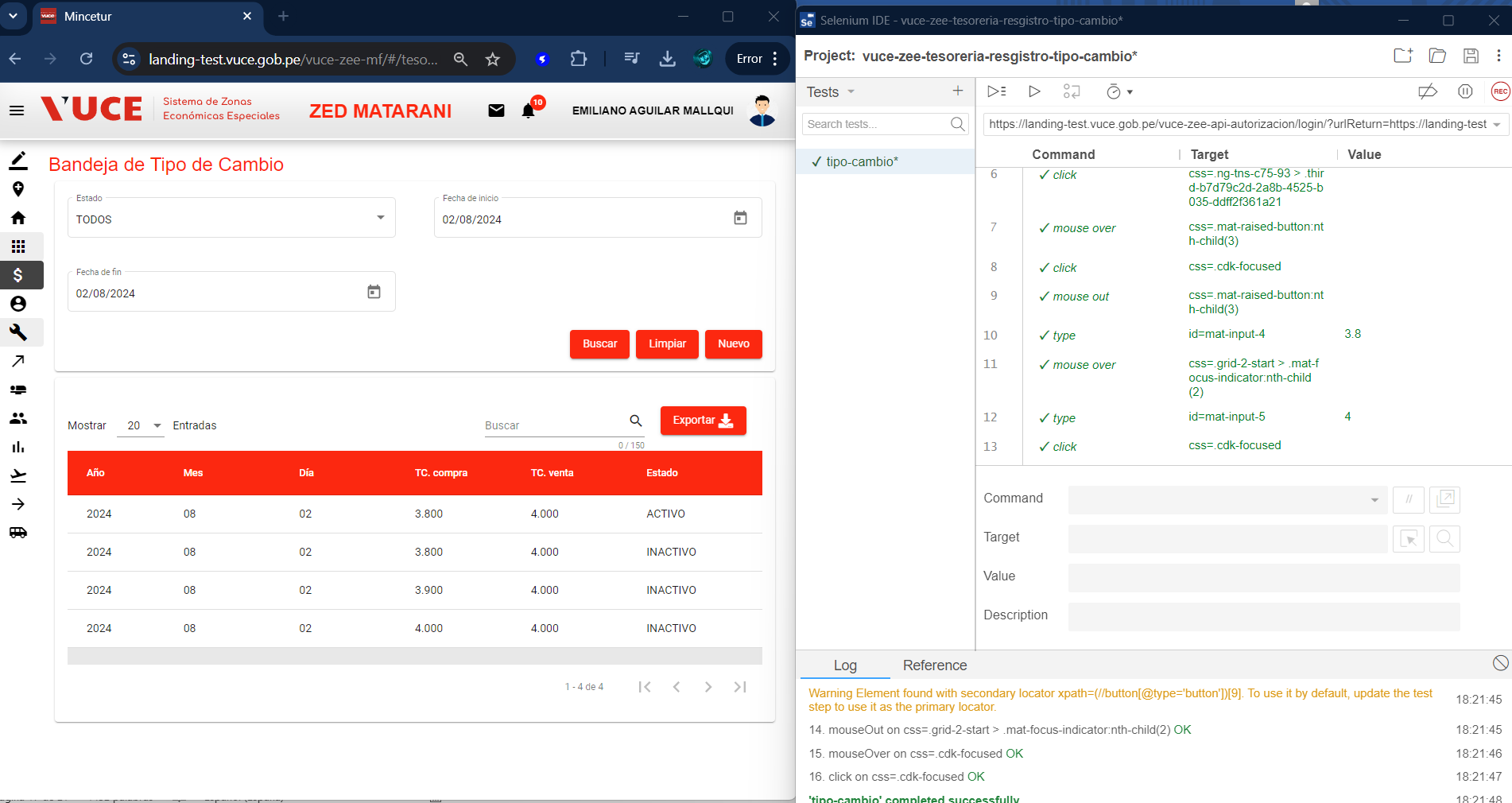


Imagen 26: finalización del script tipo de cambio correctamente.

### Prueba de Mensajería, Crear Notificación

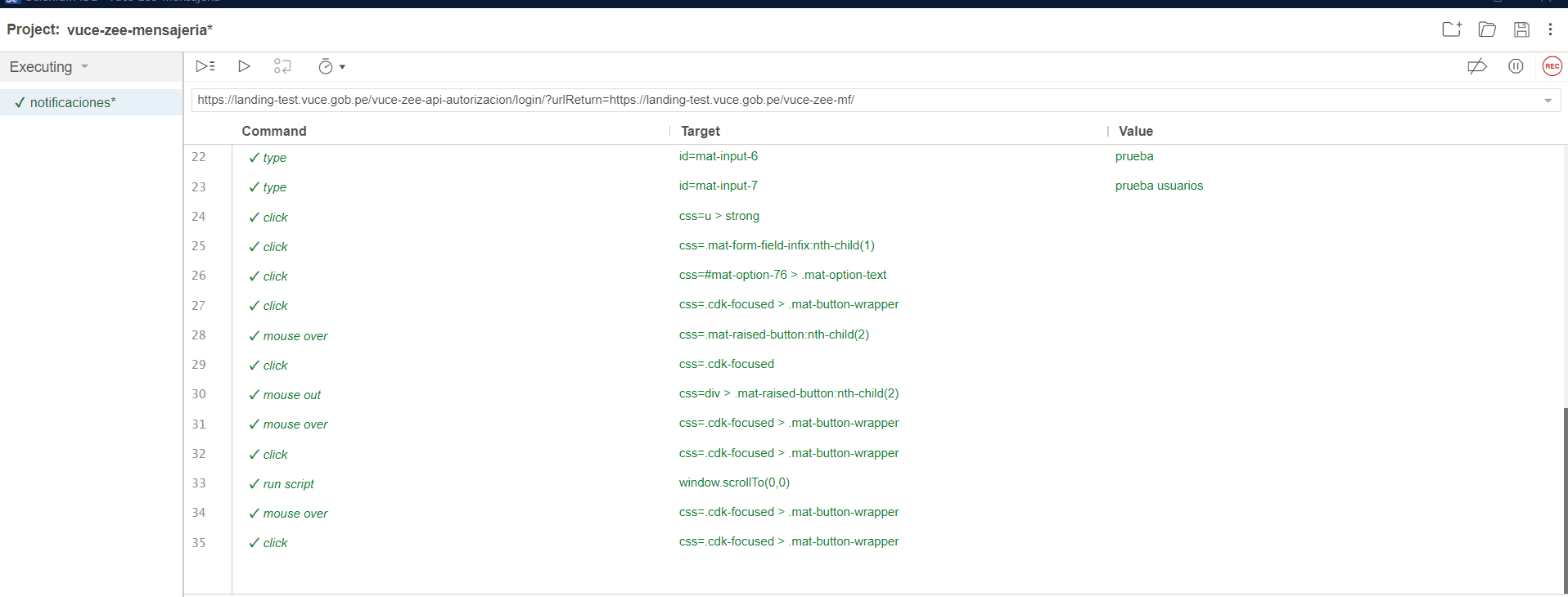


Imagen 27 :Spect de Mensajería, crear notificación.

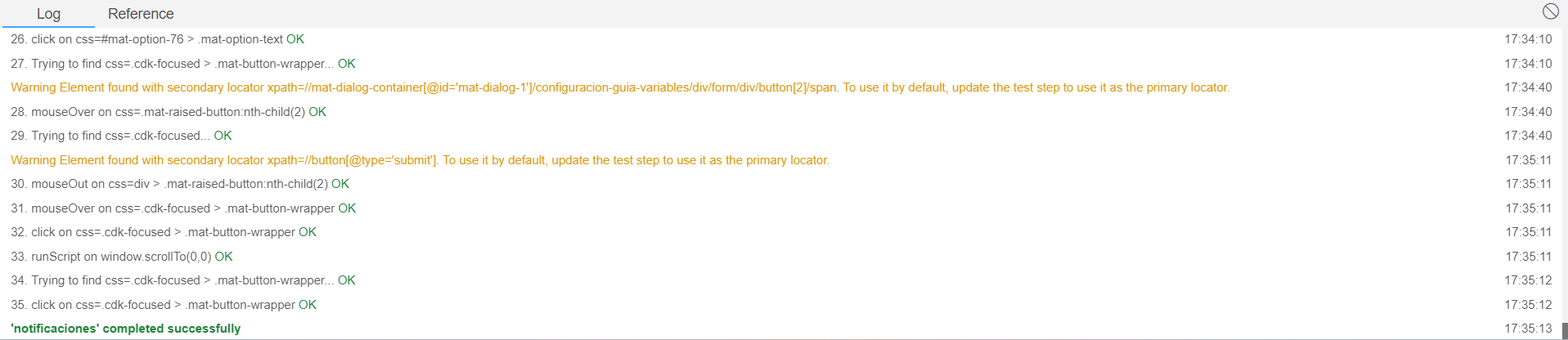


Imagen 28: Log de testing de Mensajería, crear notificación.

* La siguiente imagen muestra la ejecución del script correctamente.

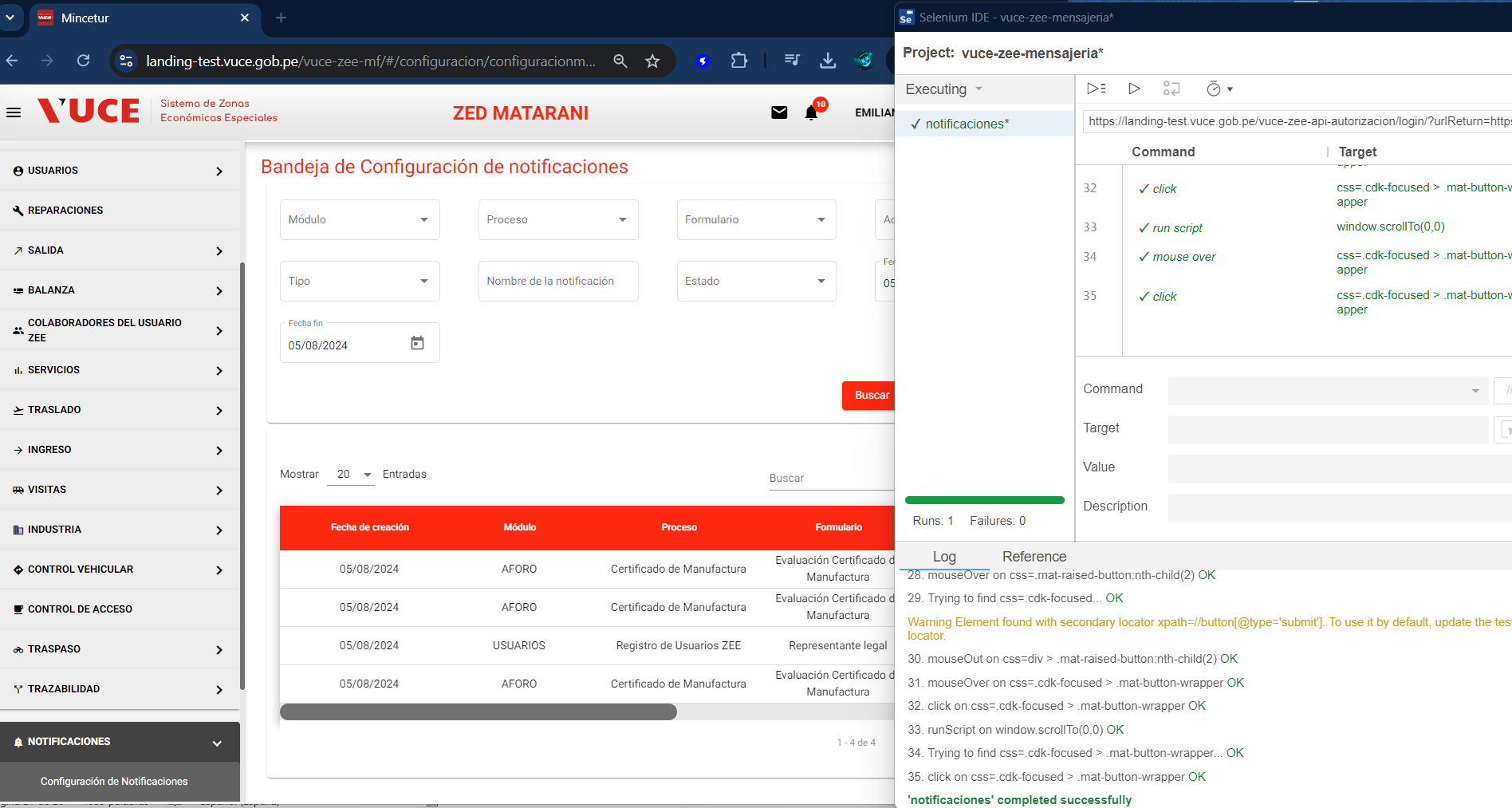


Imagen 29: finalización correcta del script

### Prueba Control Acceso : Bandeja Control Acceso

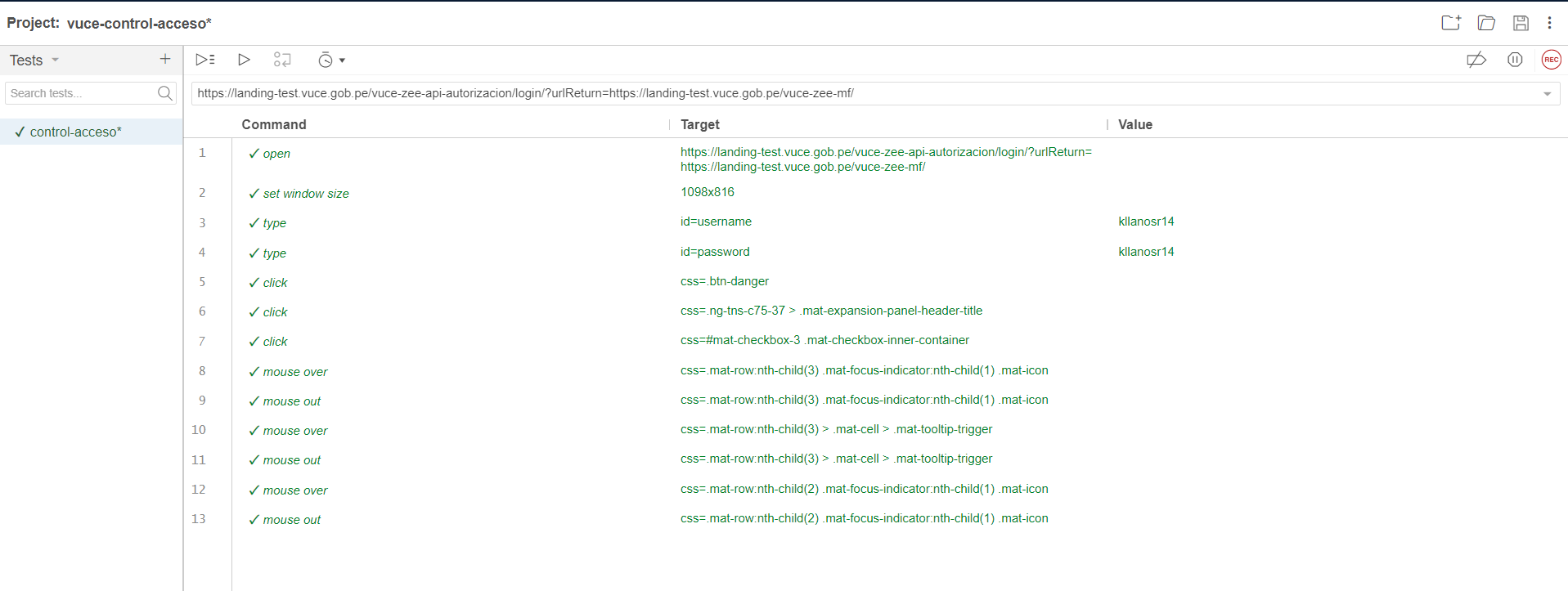


Imagen 30: script de usuario, crear usuario

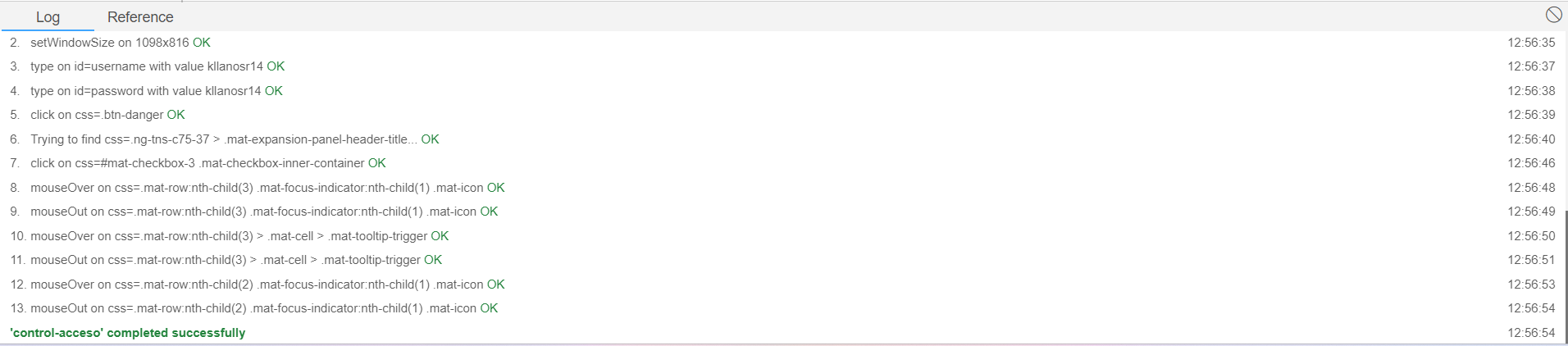


Imagen 31: Log de testing de usuarios, crear usuario.

* La siguiente imagen comprueba el correcto flujo para la creación de usuario.

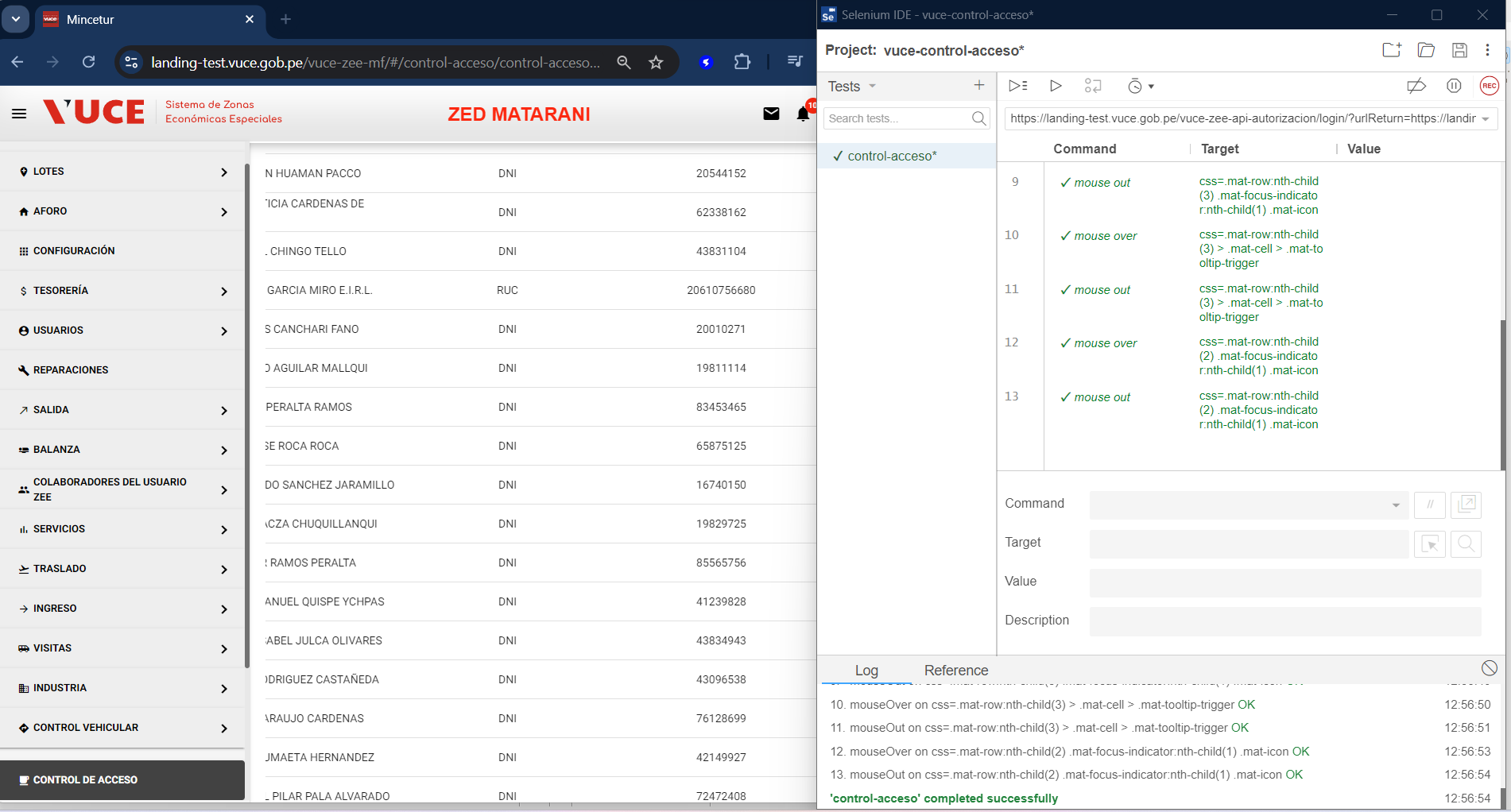


Imagen 32: finalización del script para creación usuario.

# Resultados de las Pruebas

A continuación, se describen los principales resultados del presente informe:

1. Se seleccionaron las funcionalidades clave del sistema para ser incluidas en las pruebas automatizadas.
2. Se diseñaron casos de prueba que cubrieran diferentes escenarios y flujos de trabajo del sistema.
3. Se utilizó Selenium como herramienta principal para la automatización de las pruebas.
4. Fueron ejecutadas 9 pruebas entre ella tenemos el módulo de Autenticación, Ingresos, Salida, Aforo, Balanza, Contrato, Tesorería, Mensajería, Control de Acceso para la validación del sistema, teniendo una correcta funcionalidad del sistema.
5. Se emplearon lenguajes de programación como JavaScript para desarrollar los scripts de pruebas.
6. Se ejecutaron las pruebas en el navegador Chrome para garantizar la compatibilidad.

# Conclusiones

A continuación, se describen las principales conclusiones del presente informe:

1. Las pruebas funcionales automatizadas con Selenium en el sistema ZEE Perú han demostrado ser efectivas en la verificación del correcto funcionamiento de las funcionalidades del sistema. La automatización de las pruebas ha permitido acelerar el proceso de prueba y garantizar la consistencia y calidad de las pruebas en cada iteración del desarrollo.
2. Las pruebas funcionales automatizadas con Selenium ha mejorado significativamente la cobertura de pruebas, permitiendo identificar y corregir errores de manera temprana en el ciclo de desarrollo.

# Recomendaciones

A continuación, se describen las principales recomendaciones del presente informe:

1. Mantener y actualizar regularmente los scripts de prueba automatizados a medida que el sistema evoluciona.
2. Ampliar la cobertura de las pruebas automatizadas para incluir un mayor número de funcionalidades y escenarios.
3. Realizar un mantenimiento adecuado de los scripts de prueba automatizados para asegurar su correcto funcionamiento a lo largo del tiempo.
4. Se recomienda que cuando el script este ejecutando no interferir en el uso del navegador ya que esto puede afectar el flujo y no ejecutarse correctamente.