7-12-2020

Ignacio Jara – Jonathan Navarro

Técnicas de calidad de software

Sección 001D

Yarella Francies Fernandez Burgos

# Índice:

1. Introducción
2. Plan de pruebas.
3. Casos de prueba.
4. Informe de Jenkins de casos de pruebas realizados con SonarQube.
5. Informe de resultados JMeter.
6. Informe resultados casos de prueba ejecutados y propuestas de mejoras.
7. Conclusión
8. Bibliografía

# Índice

[Introducción](#_Introducción)

[Plan de pruebas](#_Plan_de_pruebas)

[Casos de prueba](#_Casos_de_prueba)

[Informe de Jenkins de casos de pruebas realizados con SonarQube](#_Informe_de_Jenkins)

[Informe de resultados JMeter](#_Informe_de_resultados)

[Informe resultados casos de prueba ejecutados y propuestas de mejoras](#_Informe_resultados_casos)

[Conclusión](#_Conclusión)

[Bibliografía](#_Bibliografía)

# Introducción

El presente documento se refiere a las problemáticas encontradas en el proyecto “CarritoComprasWEB”.

Se mostrarán los distintos informes que se han realizado con el trascurso del tiempo, los cuales muestran errores que hay en el proyecto, principalmente el código, también se mostraran las propuestas para mejorar algunos módulos, los distintos tipos de prueba a realizar y algunos módulos que nosotros pudimos arreglar.

# Plan de pruebas

**Historial de versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 15/09/2020 | 1.0 | Jonathan Navarro | Duoc Software Solution | Versión inicial del archivo |
| 05/12/2020 | 2.0 | Jonathan Navarro | Duoc Software Solution | Modificación del archivo relacionado con el final del proyecto |

**Información del proyecto**

|  |  |
| --- | --- |
| **Empresa / Organización** | Duoc Software Solution |
| **Proyecto** | CarritoComprasWEB |
| **Fecha de preparación** | 15/09/2020 |
| **Cliente** | Computer sale online |
| **Patrocinador principal** | Duoc Software Solution |
| **Gerente / Líder de proyecto** | Jonathan Navarro B. |
| **Gerente / Líder de pruebas de software** | Ignacio Jara B. |

**Aprobaciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u organización** | **Fecha** | **Firma** |
| Jonathan Navarro | Líder de proyecto | Duoc Software Solution | 15/09/2020 | jnavarrob |
| Ignacio Jara | Líder de pruebas | Duoc Software Solution | 15/09/2020 | ijarab |
| Jonathan Navarro | Líder de proyecto | Duoc Software Solution | 05/12/2020 | jnavarrob |
| Ignacio Jara | Líder de pruebas | Duoc Software Solution | 05/12/2020 | ijarab |

**Resumen ejecutivo**

La empresa “Duoc Software Solution” nos ha contratado como equipo QA, solicitándonos crear y ejecutar un plan de pruebas para un sistema de compras Online.

Este documento contendrá el objetivo del plan de pruebas, el alcance de dichas pruebas, los criterios de aceptación o rechazo, los recursos que se utilizaran para realizar dichas pruebas, los tipos de pruebas a realizar, etc.

El documento tiene como objetivo adicional el mostrar el alcance que tendrán las pruebas y qué criterios se usarán para rechazar o aceptar el proyecto.

**Alcance de las pruebas**

**En este documento los módulos a probar son:**

**1**-Ingresar producto

**2**-Modificar producto

**3**-Eliminar producto

**4**-Ingresar empleado

**5**-Modificar empleado

**6**-Eliminar empleado

**7**-Realizar compra de producto

**8**-Buscar producto

**Nuevas funcionalidades a probar**

Las nuevas funcionalidades que se van a probar son las de agregar, eliminar y actualizar a los empleados y a los productos

**Pruebas de regresión**

1. Después de haber agregado una nueva tabla a la BD, conteniendo los datos de los empleados, ningún módulo anterior ha sido afectado de forma negativa.

Al insertar la tabla empleados, se habilitaron las funciones de listar, agregar y eliminar empleado, como también se habilitó la posibilidad de agregar y eliminar los productos.

1. Después de haber modificado el código para arreglar los casos de prueba RF-2, RF-4, RF-9 y RF-11, ningún otro modulo que anteriormente haya funcionado correctamente ha sido afectado de forma negativa Y/O positiva.

**Funcionalidades a no probar**

No aplica

**Enfoque de pruebas (estrategia)**

Las pruebas a ser realizadas serán de tipo Caja Negra y Caja Blanca, se probará la interfaz de la página web, sus distintos módulos, el código y también se probará la BD

**Criterios de aceptación o rechazo**

**Criterios de aceptación:**

Si el módulo a probar cumple con el 90% de su función con respecto a las necesidades del cliente será aprobado. Si un módulo se encuentra entre un 85% y un 89% se hará una reunión para ver si dicho módulo puede ser aceptado o rechazado dependiendo de su función en el programa

**Criterios de rechazo:**

Si el módulo a probar no cumple con el 84% de su función con respecto a las necesidades del cliente será rechazado, lo cual implica que el módulo tendrá que ser modificado y posteriormente puesto en prueba otra vez.

**Criterios de suspensión**

El QA debe tener un ambiente laboral de pruebas óptimo para la ejecución de estas, las herramientas necesarias para la ejecución del plan de prueba de este proyecto son:

-Un computador que cumpla con los requerimientos mínimos del software.

-Un computador con los softwares necesarios para la ejecución del software a probar.

-Datos necesarios para loguearse en el software y probar los módulos del sistema.

-Además en caso de que el QA se ausente por necesidad mayor (enfermedad, asunto personal, etc).

**Criterios de reanudación**

Las pruebas serán reanudadas cuando el/los problemas se hayan resuelto.

Las pruebas serán reanudadas cuando el QA tenga los recursos necesarios para hacer las pruebas.

**Entregables**

Para la primera iteración de este proyecto se entregarán los documentos del plan de pruebas y los casos de prueba.

Para la segunda iteración se entregarán el informe de resultado de Jenkins + SonarQube y el informe de resultado de JMeter.

Para la iteración final se entregarán los resultados de los casos de prueba ejecutados y las propuestas de mejora.

**Recursos**

**Requerimientos de entornos – Hardware**

Procesador: Intel i3 o superior

Memoria RAM: 4GB o superior

Almacenamiento: 15 GB o superior

**Requerimientos de entornos – Software**

Netbeans JEE (Java Enterprise Edition) 8 o superior + servidor GlassFish

Xampp versión 7.3 o superior

SonarQube

SonarScanner

JMeter

**Herramientas de pruebas requeridas**

Las herramientas que se usarán para realizar las pruebas son SonarQube, SonarScanner, Glassfish, Netbeans y JMeter

**Personal**

Líder del proyecto: Jonathan Navarro B.

Líder de pruebas: Ignacio Jara B.

**Entrenamiento**

Para el entrenamiento del personal se utilizará un manual de usuario el cual se le entregará al jefe de la empresa, este manual contendrá toda la información básica sobre el uso del programa

**Planificación y organización**

**Procedimientos para las pruebas**

Se utilizarán pruebas del tipo caja negra y caja blanca

**Matriz de responsabilidades**

|  |  |  |
| --- | --- | --- |
| **Tareas/Roles** | **Líder de proyecto** | **Líder de pruebas** |
| Creación del plan de pruebas | R | A |
| Casos de prueba (Antes de edición) | R | A |
| Casos de prueba (Post edición) | R | A |
| Pruebas de caja negra y blanca | A | R |
| Informe de pruebas | R | A |
| Informe de resultados de prueba en Jenkins + SonarQube | A | R |
| Informe de resultado de pruebas con JMeter | A | R |
| Informe resultado de casos de prueba ejecutados | A | R |
| Propuestas de mejora | A | R |

**Cronograma**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Actividades | Semana 1 | Semana 2 | Semana 3 | Semana 4 | Semana 5 | Semana 6 | Semana 7 | Semana 8 |
| Creación del plan de pruebas |  |  |  |  |  |  |  |  |
| Casos de prueba (antes de edición) |  |  |  |  |  |  |  |  |
| Casos de prueba (Post edición) |  |  |  |  |  |  |  |  |
| Pruebas de caja negra y blanca |  |  |  |  |  |  |  |  |
| Informe de pruebas |  |  |  |  |  |  |  |  |
| Informe de resultados de prueba en Jenkins + SonarQube |  |  |  |  |  |  |  |  |
| Informe de resultado de pruebas con JMeter |  |  |  |  |  |  |  |  |
| Informe resultado de casos de prueba ejecutados |  |  |  |  |  |  |  |  |
| Propuestas de mejora |  |  |  |  |  |  |  |  |

**Premisas**

No aplica

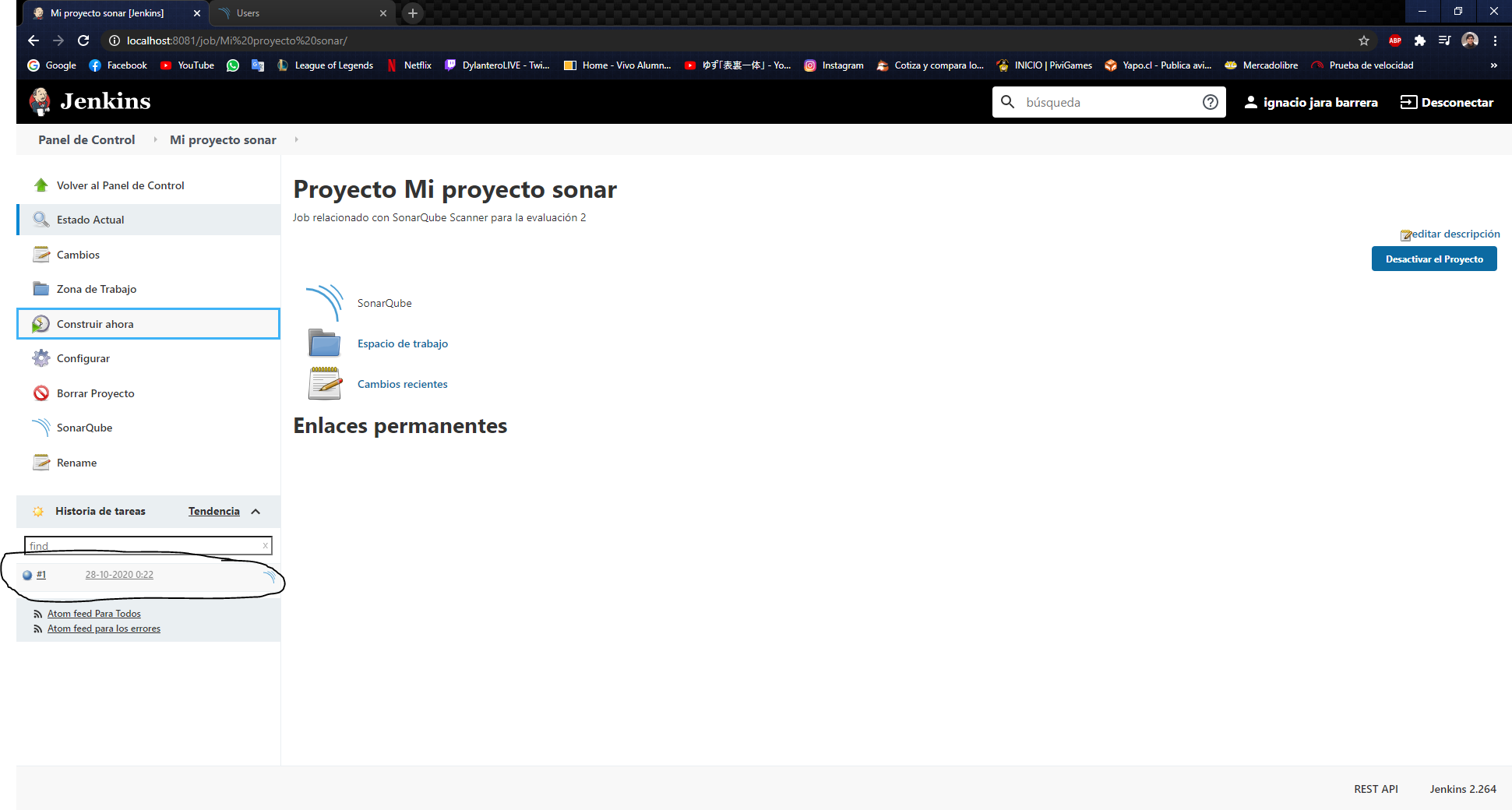
**Dependencias y Riesgos**

Los riesgos presentes en este proyecto son principalmente al momento de modificar el código para reparar una falla X que cause fallas en otros módulos que anteriormente funcionaban, en caso de haber cualquier fallo causado por la modificación del código se tiene un respaldo en limpio del proyecto el cual puede ser restaurado en cualquier momento.

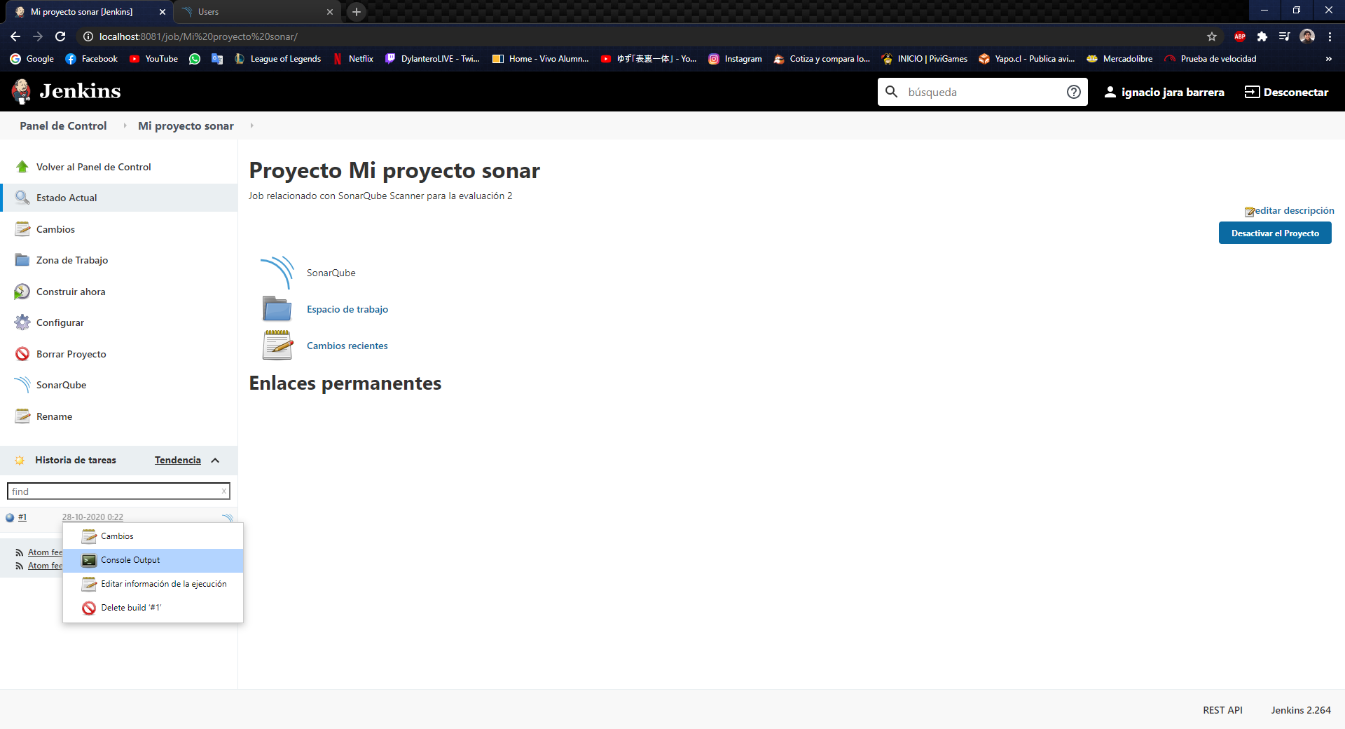
# Casos de prueba

[Planilla Casos de prueba](../../../Casos%20de%20Prueba_revision.xlsx)

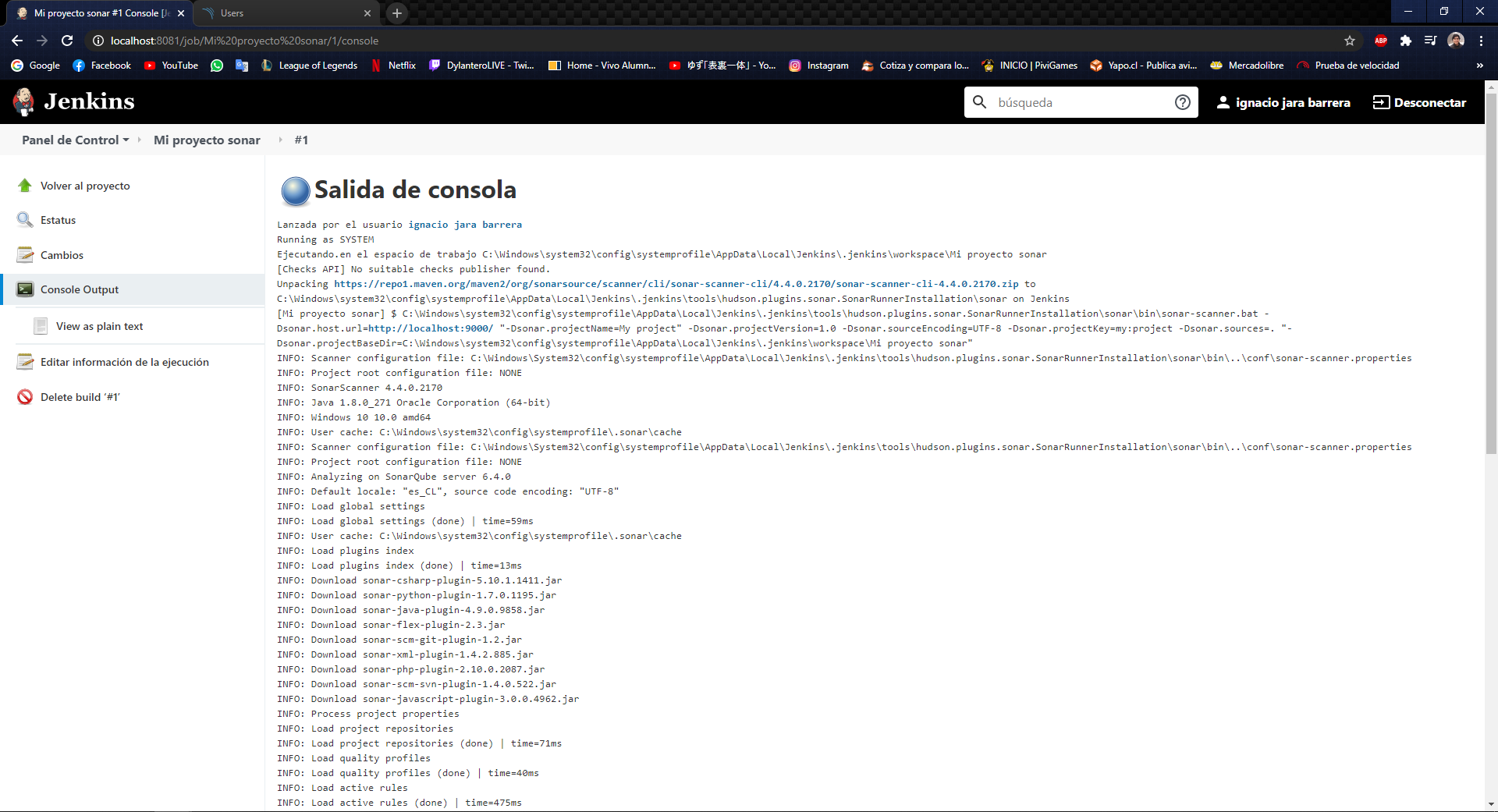
# Informe de Jenkins de casos de pruebas realizados con SonarQube

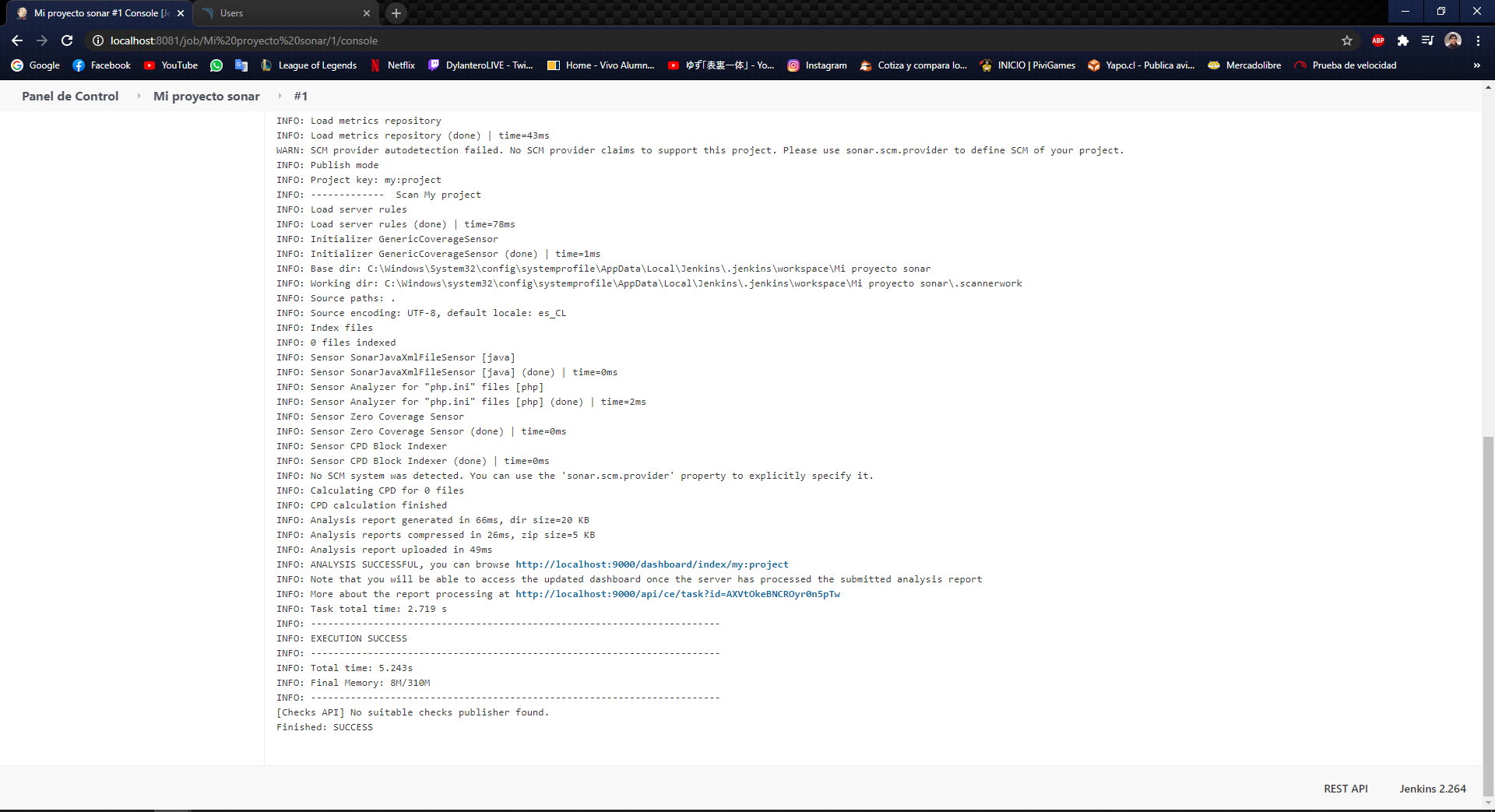


Paso 1: Ejecutamos una tarea presionando la opción “Construir ahora”, la cual quedara registrada en el historial de tareas.



Paso 2: Seleccionamos la tarea ejecutada y seleccionamos la opción “Console Output”



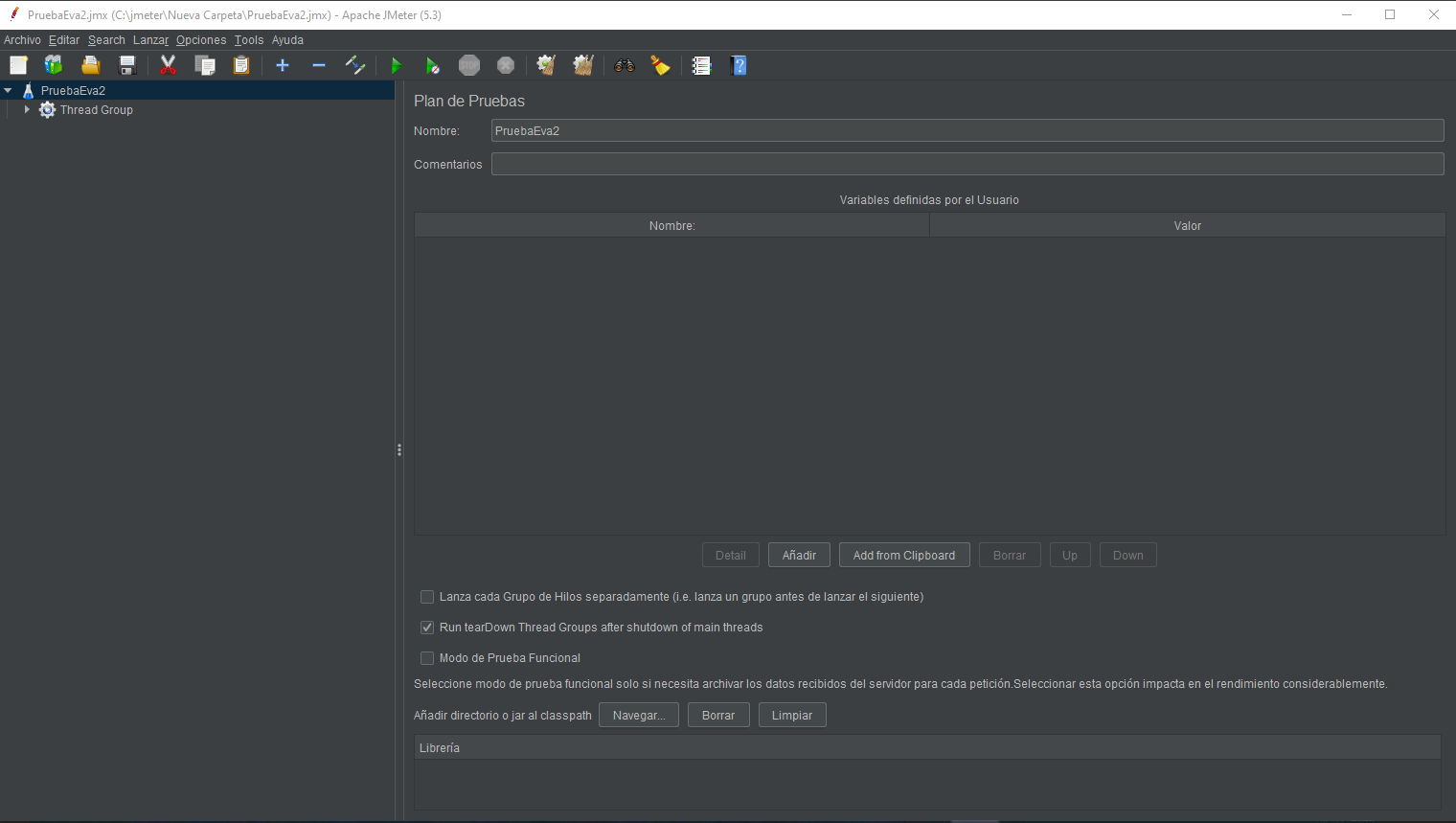


Paso 3: Al seleccionar la opción anterior se desplegará un informe de la ejecución de la tarea en formato de console output, la cual fue ejecutada correctamente como muestran las imágenes.

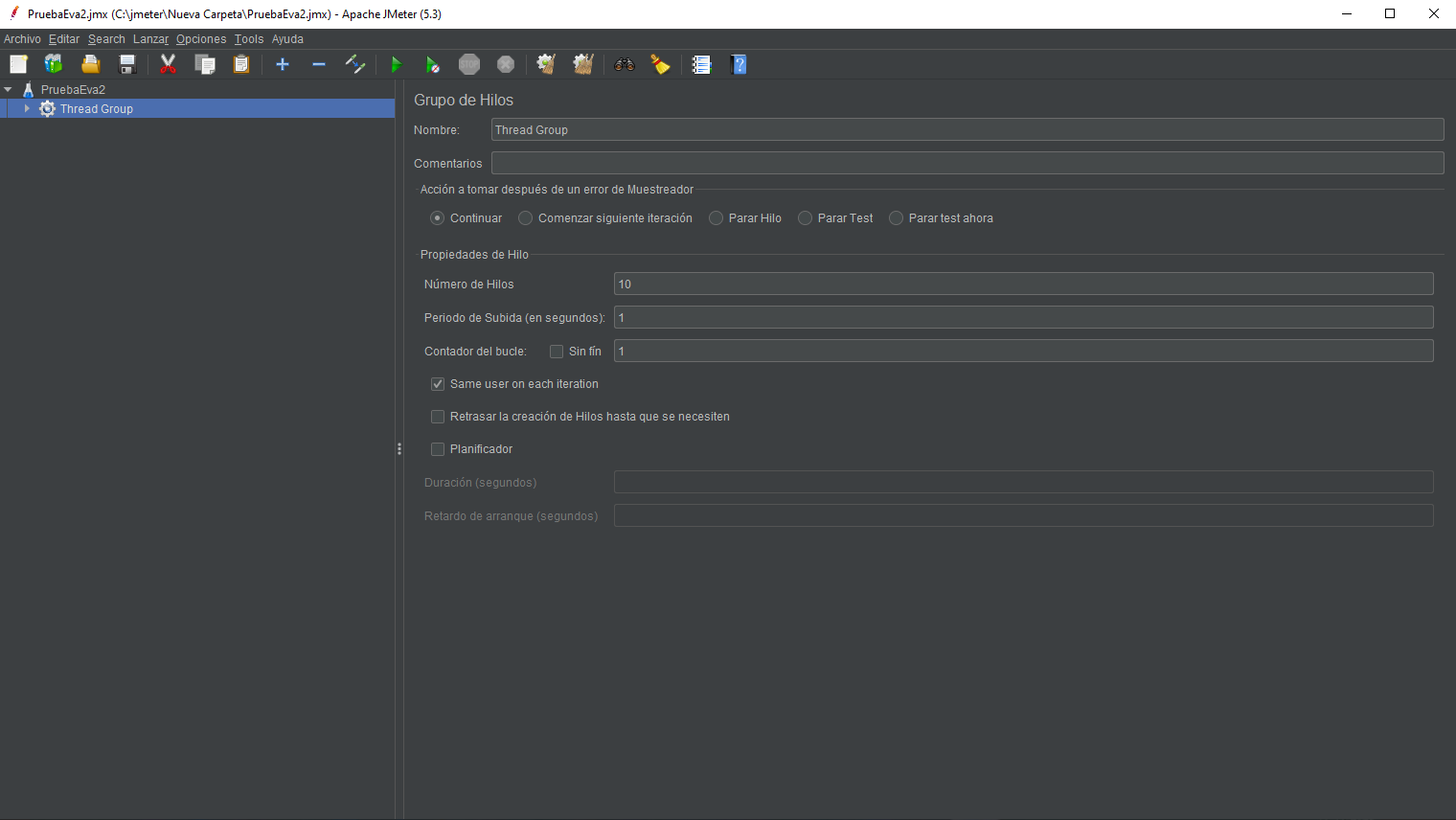
# Informe de resultados JMeter

[1-Comp](#_1-Instalación_de_Jenkins)ortamiento de JMETER en Carrito de compras con 10 usuarios simultáneos y tiempo de demora

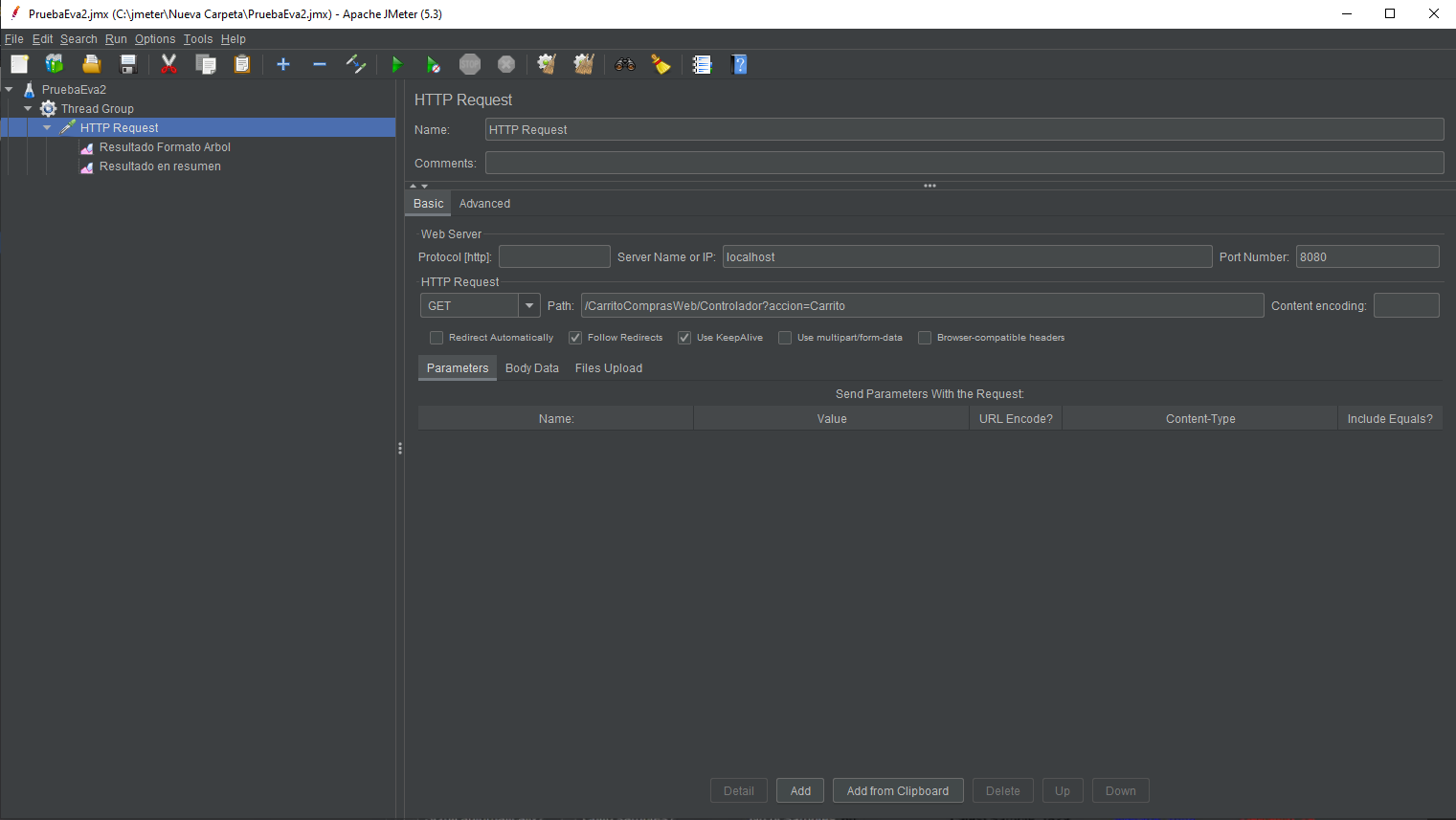
-Configuraciones iniciales para el test de carrito de compras



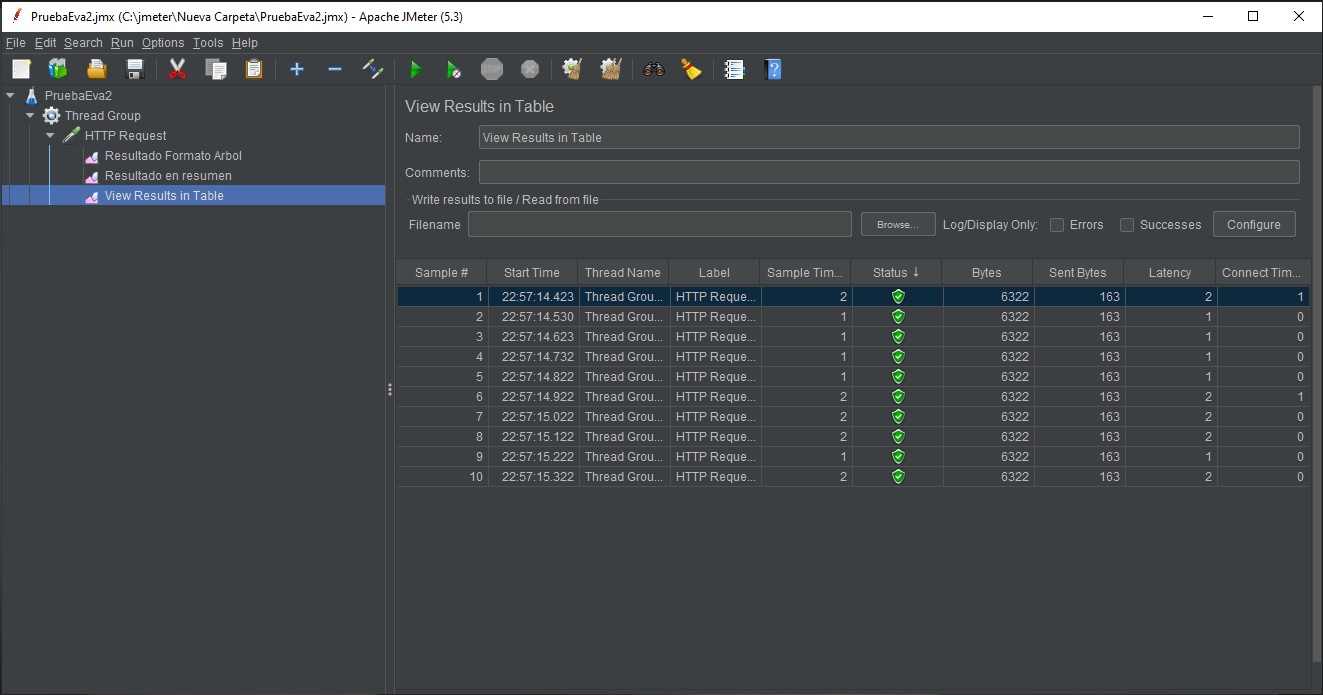
-Usuarios simultaneos:10



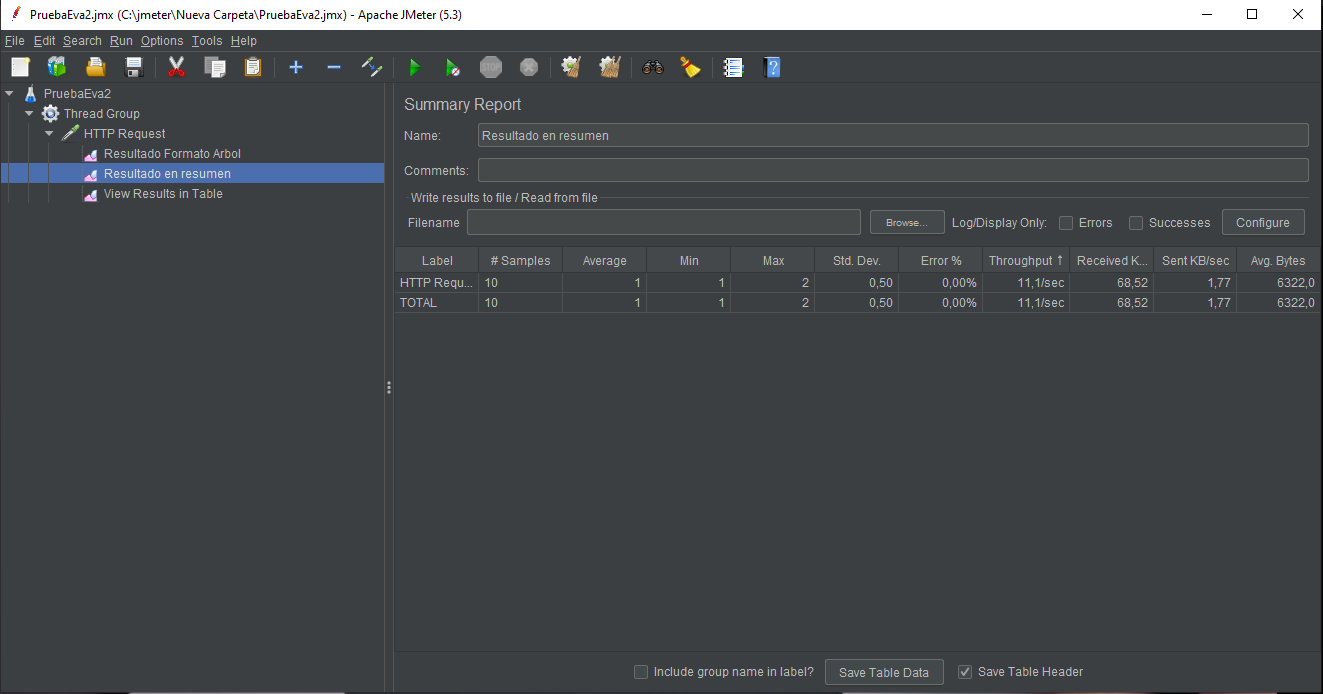
-Para el informe, se creó un test de tipo HTTP Request, el link utilizado es el del proyecto semestral “Carrito de compras”: http://localhost:8080/CarritoComprasWeb/Controlador?accion=Carrito



-Resultados de prueba de tipo tablas

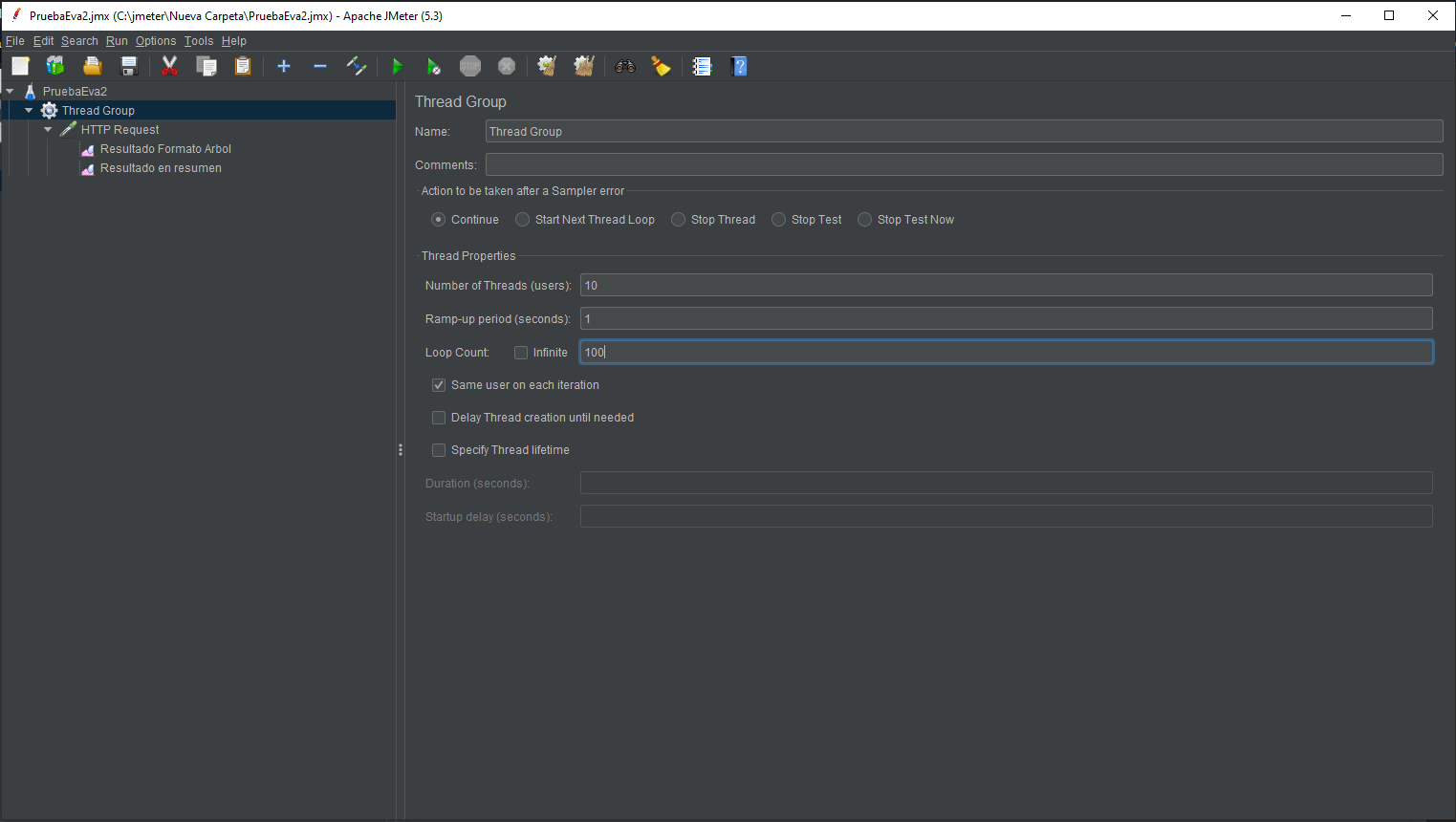


-Resultados de prueba de tipo resumen, la prueba finalizo en un tiempo de 1.89Segundos.

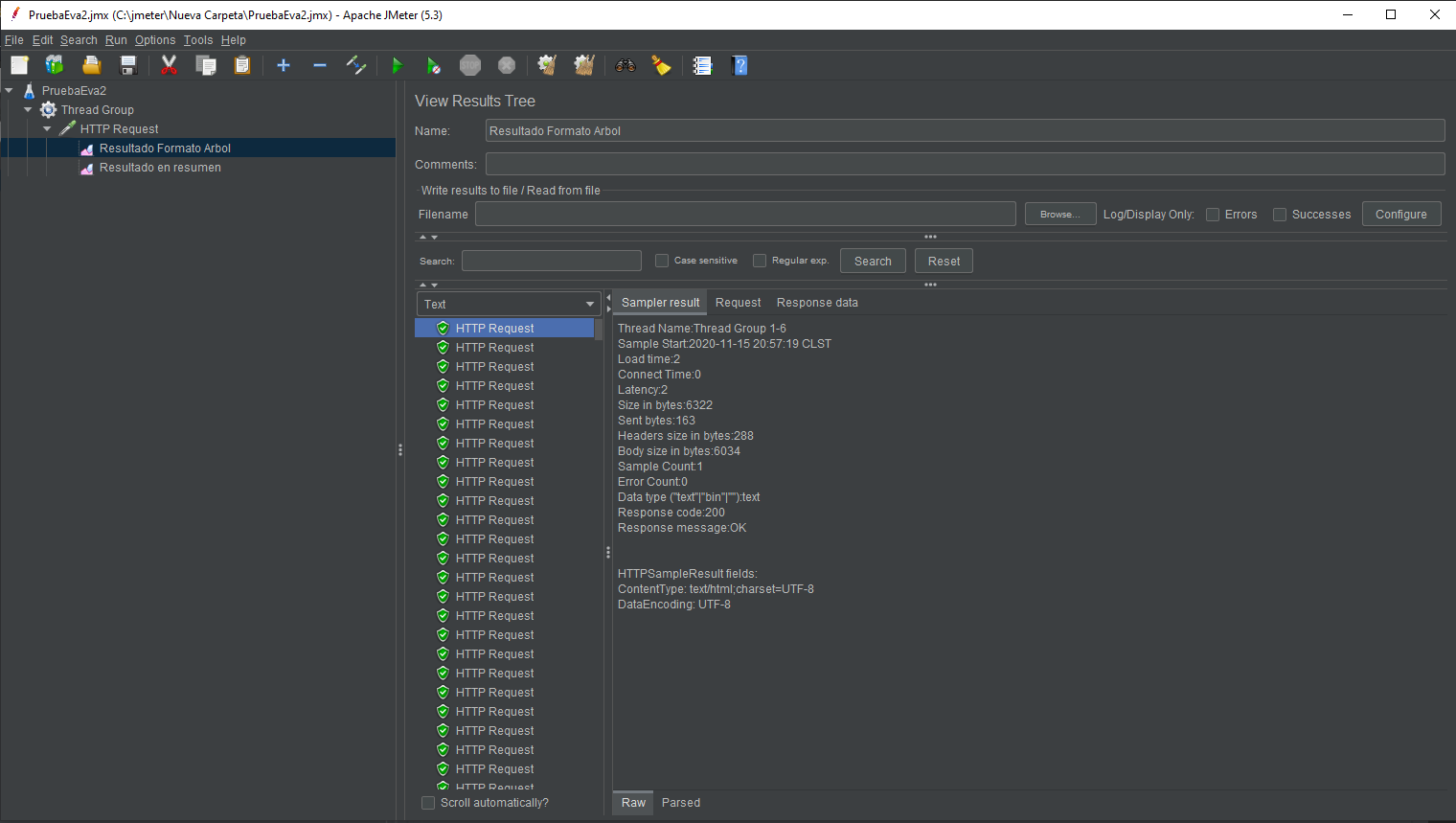


[2-](#_3-Integración_de_SonarQube)Comportamiento de JMETER con 10 usuarios simultáneos y un loop de 100 repeticiones

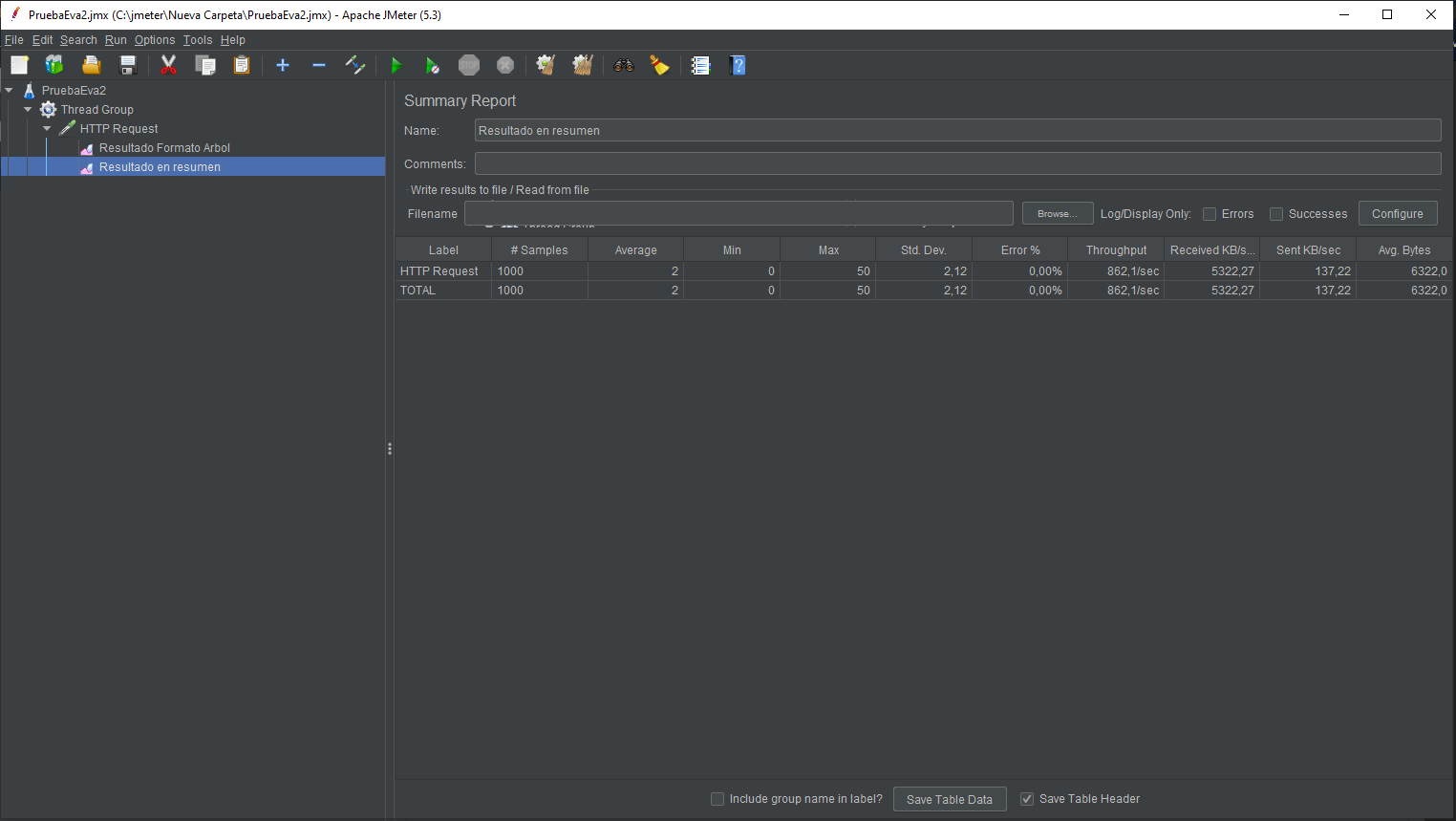
-Configuración inicial de prueba



-Resultado de prueba en tipo árbol

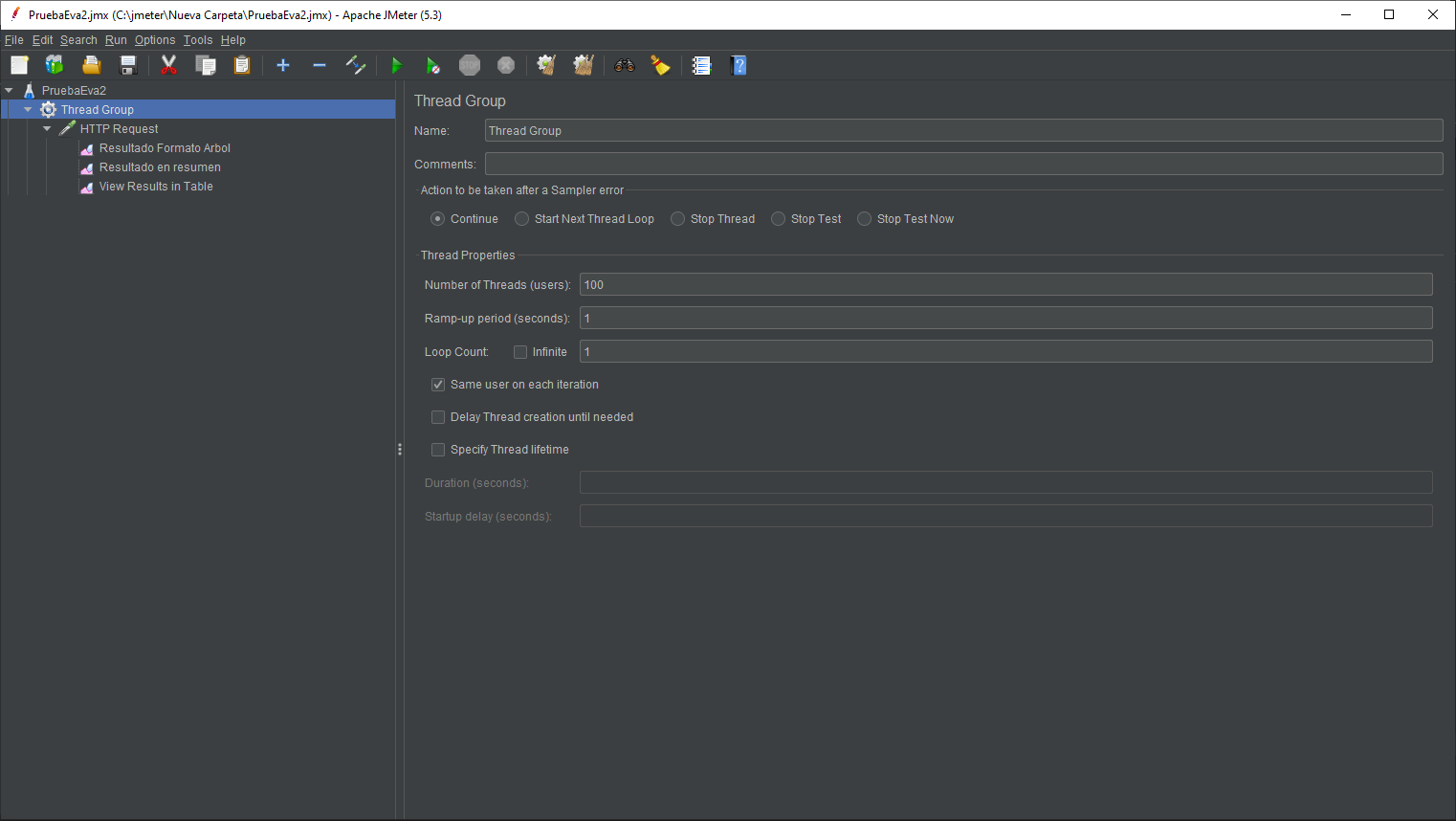


-Resultado de prueba en tipo resumen, la prueba fue exitosa y finalizo en un tiempo de 2.5 segundos

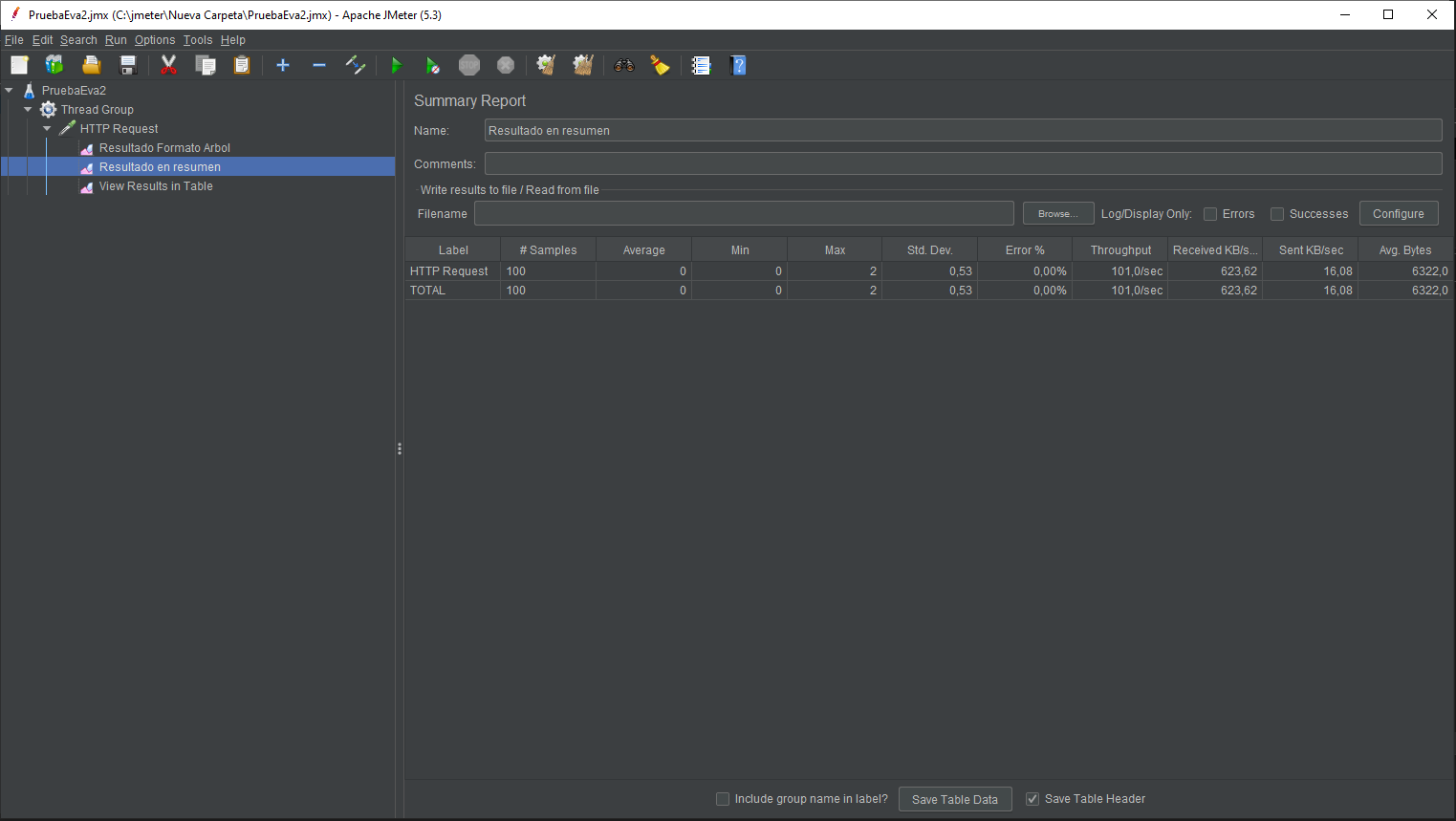


[3-](#_4-Informe_de_resultados)Comprobando límites de usuarios simultáneos que soporta Carrito de compras

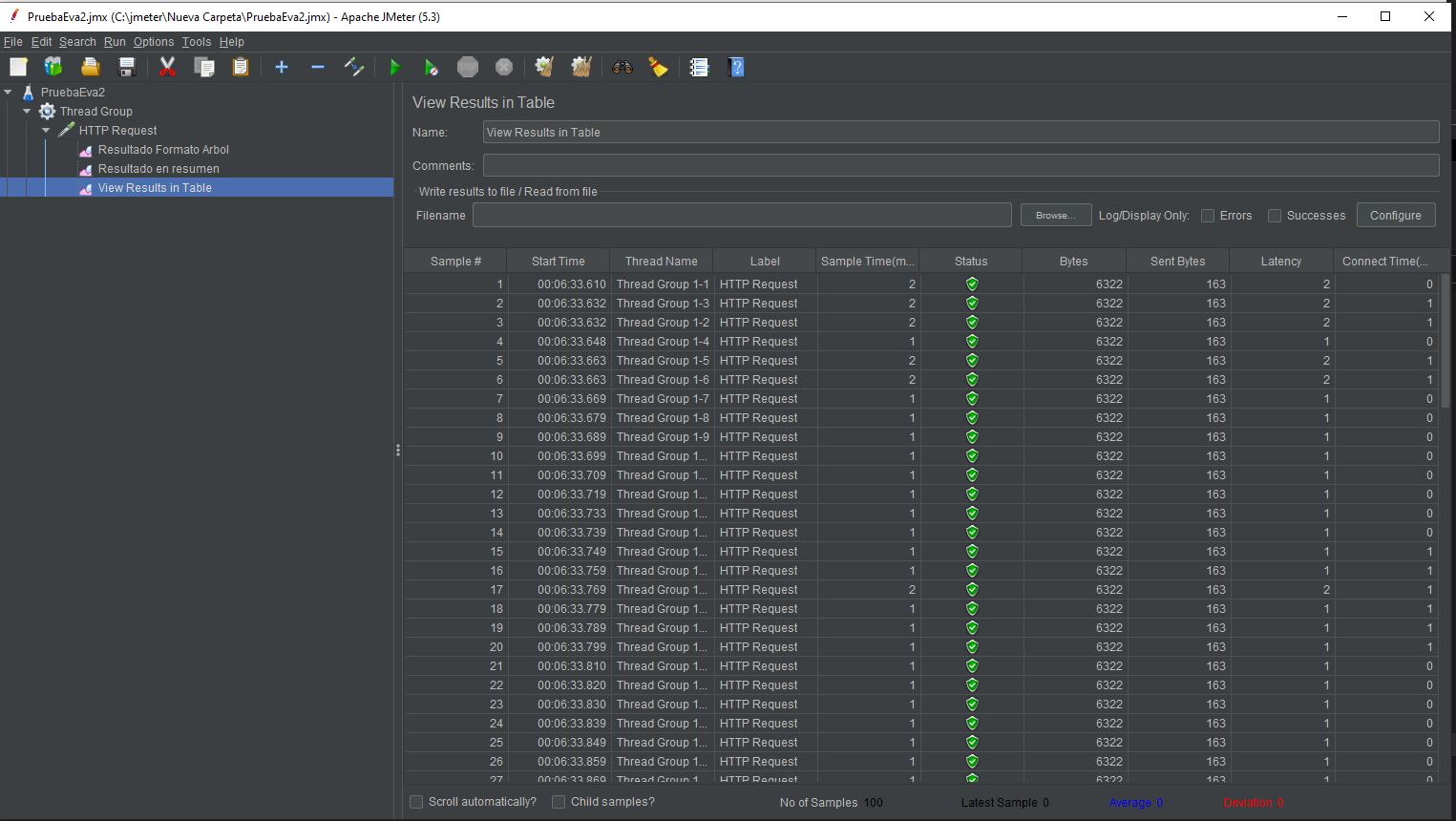
-Configuración inicial de pruebas con 100 usuarios simultáneos.



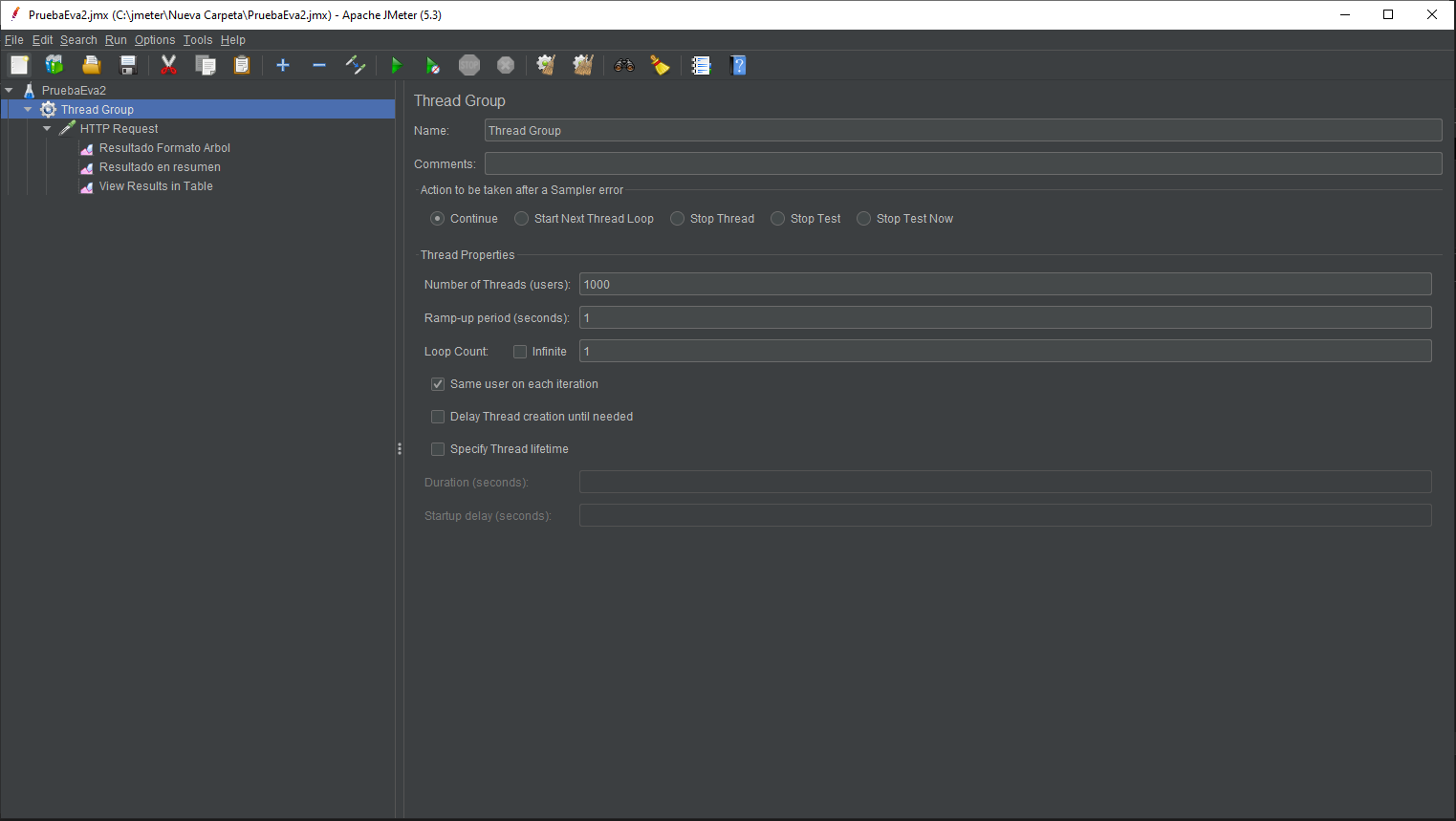
-Resultado de prueba con 100 usuarios en tipo resumen



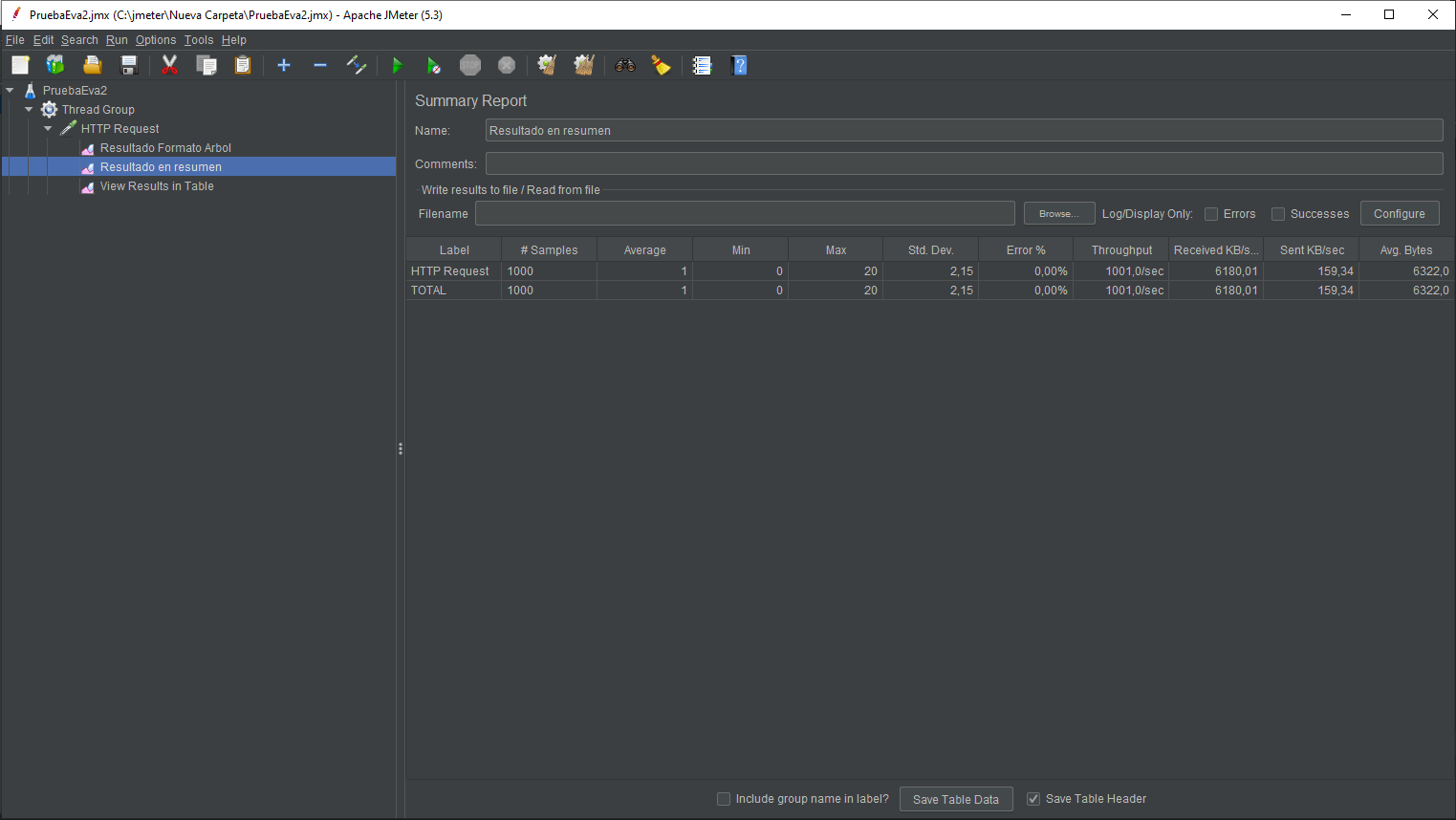
-Resultado de prueba con 100 usuarios en tipo tabla, prueba totalmente exitosa sin colapsos



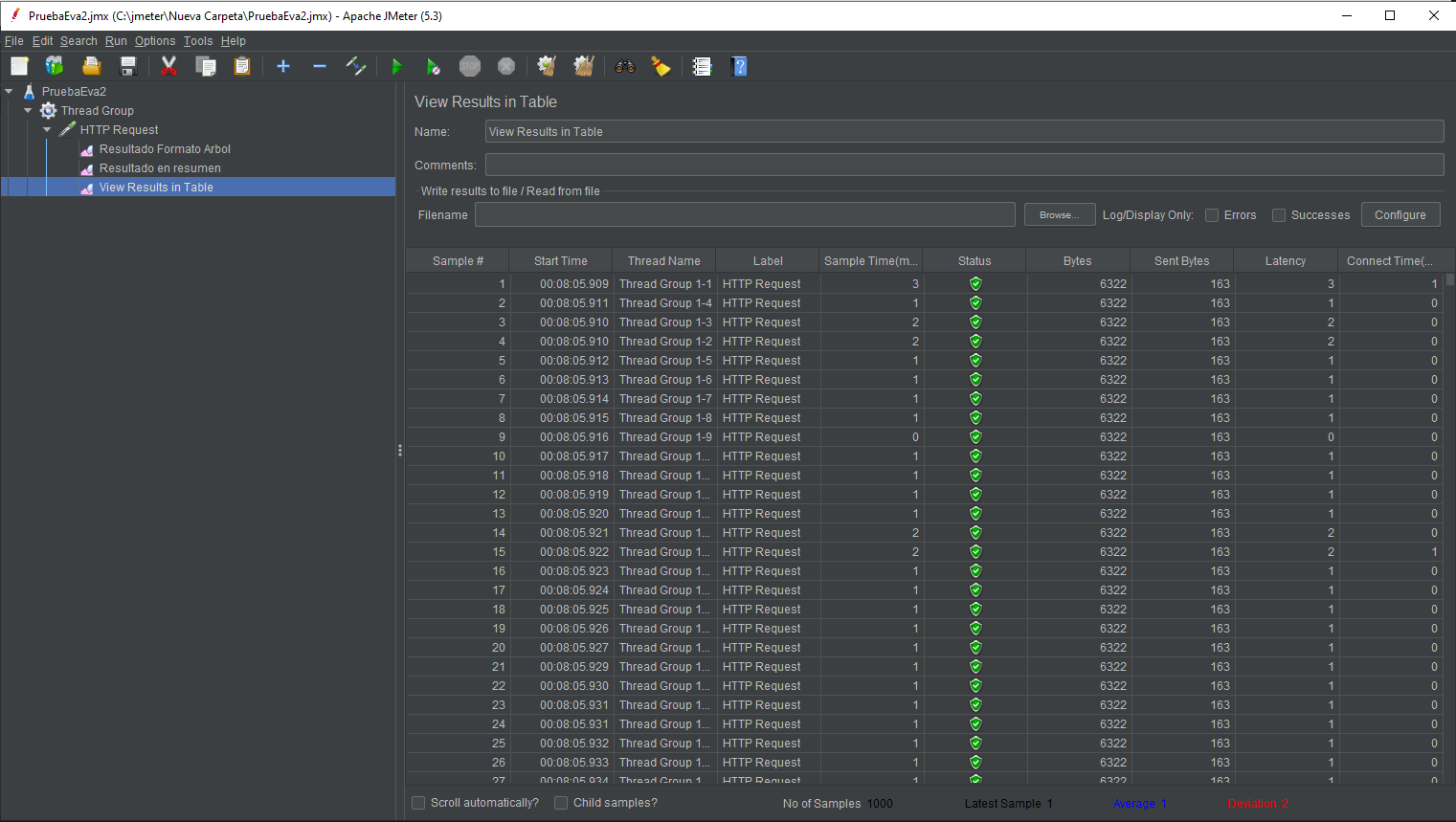
-Configuración inicial de pruebas con 1000 usuarios simultáneos.



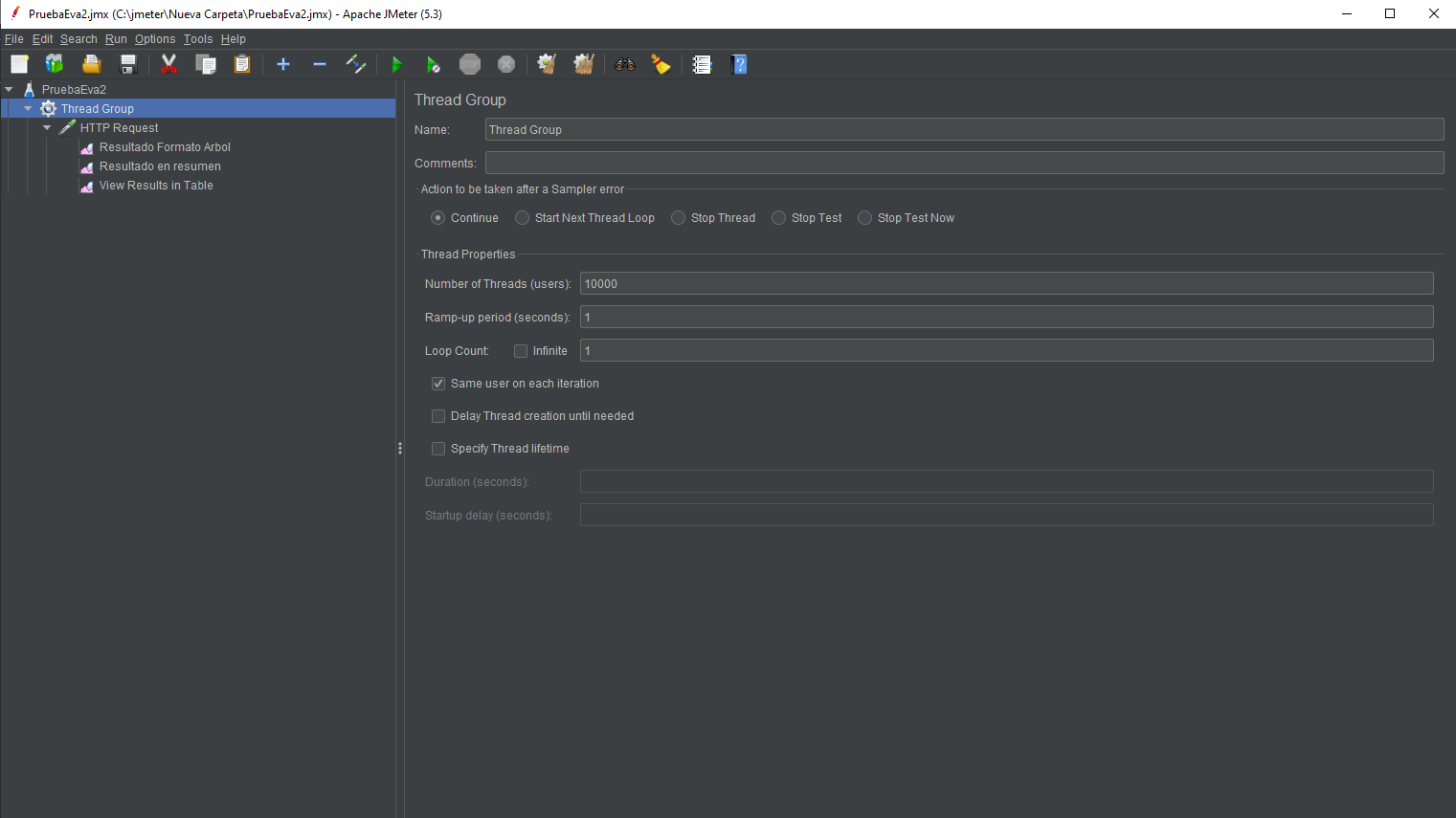
-Resultado de prueba con 1000 usuarios en tipo resumen



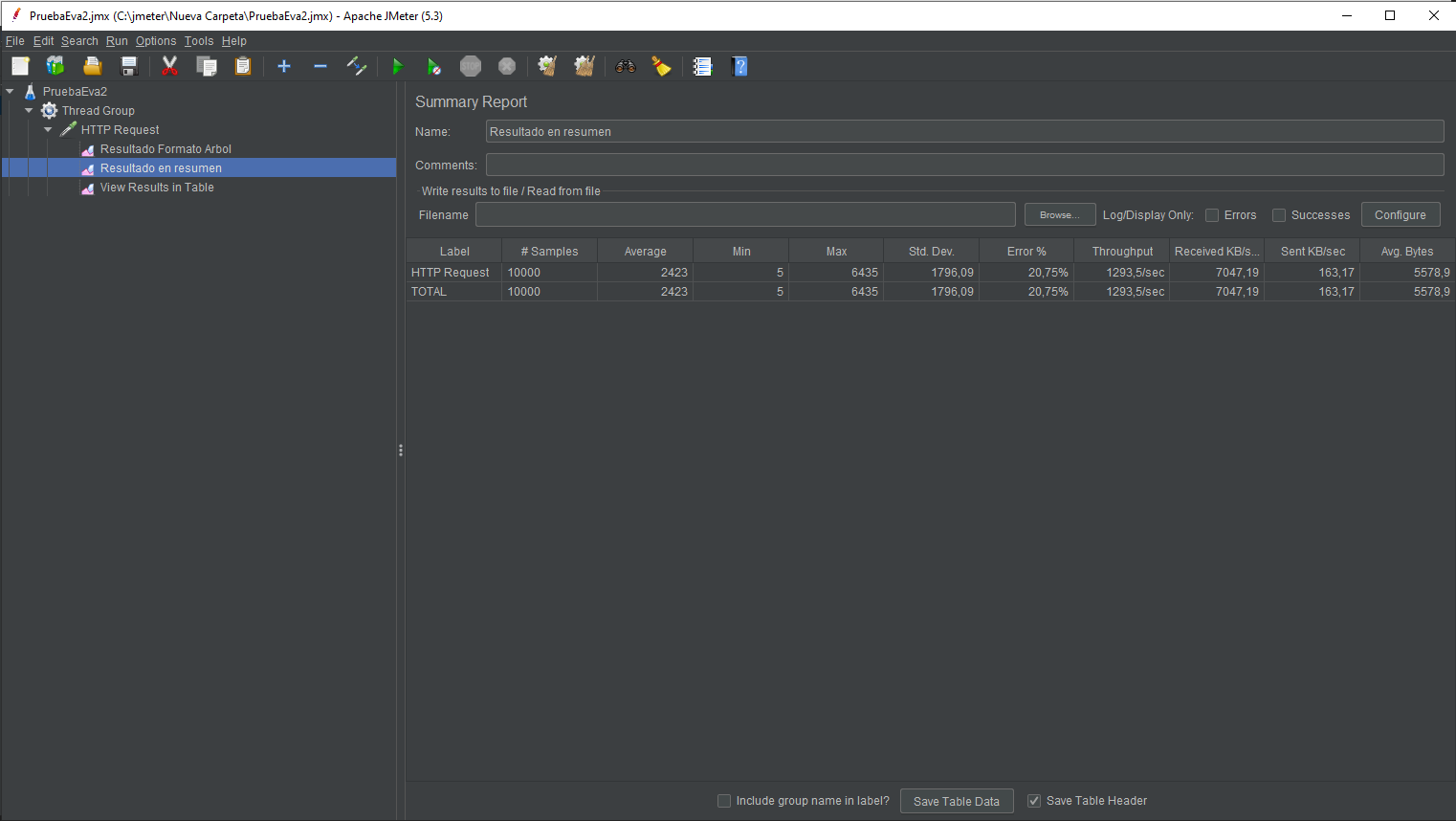
-Resultado de prueba con 1000 usuarios en tipo tabla, prueba totalmente exitosa sin colapsos

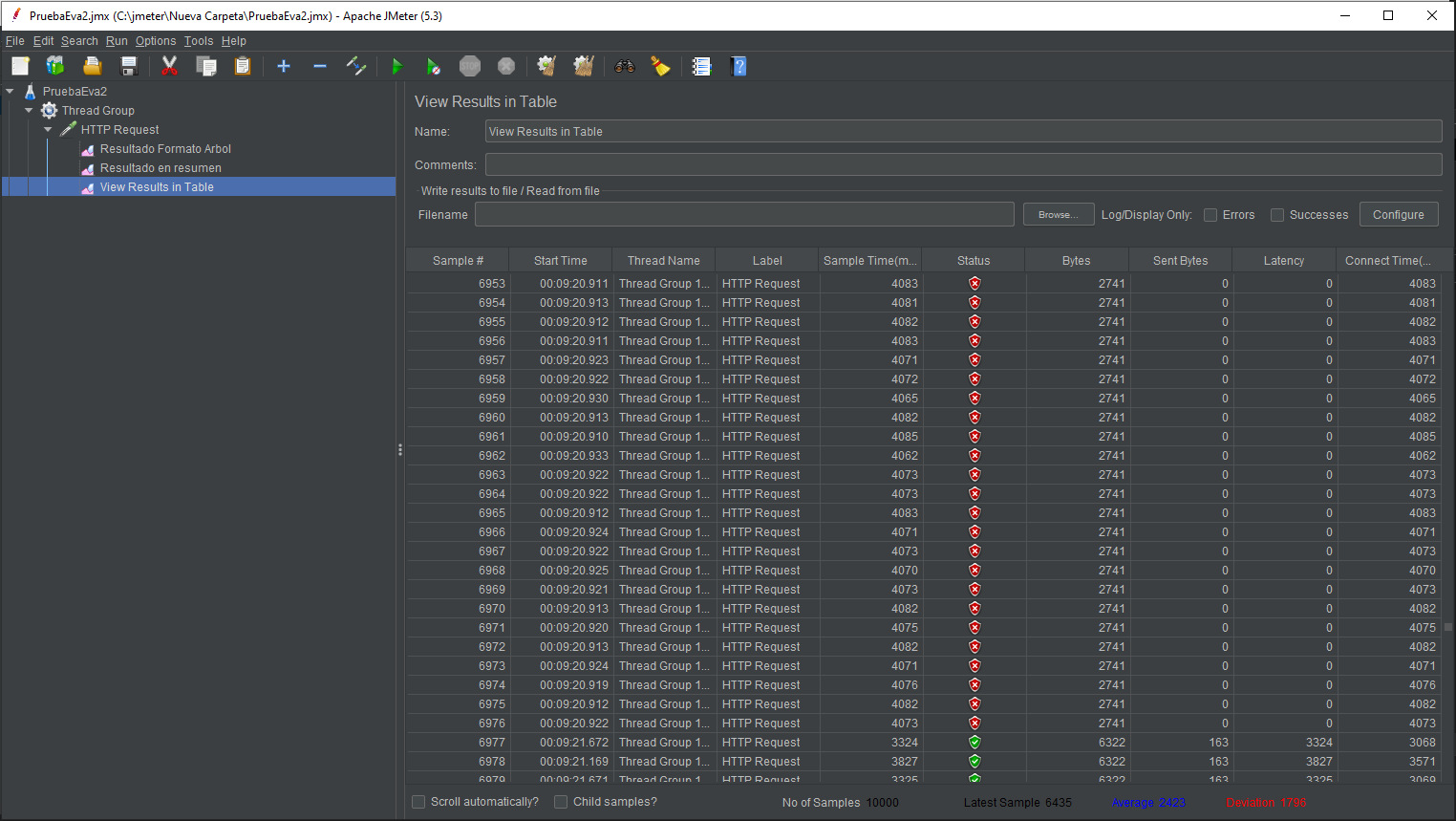


-Configuración inicial de pruebas con 10000 usuarios simultáneos.



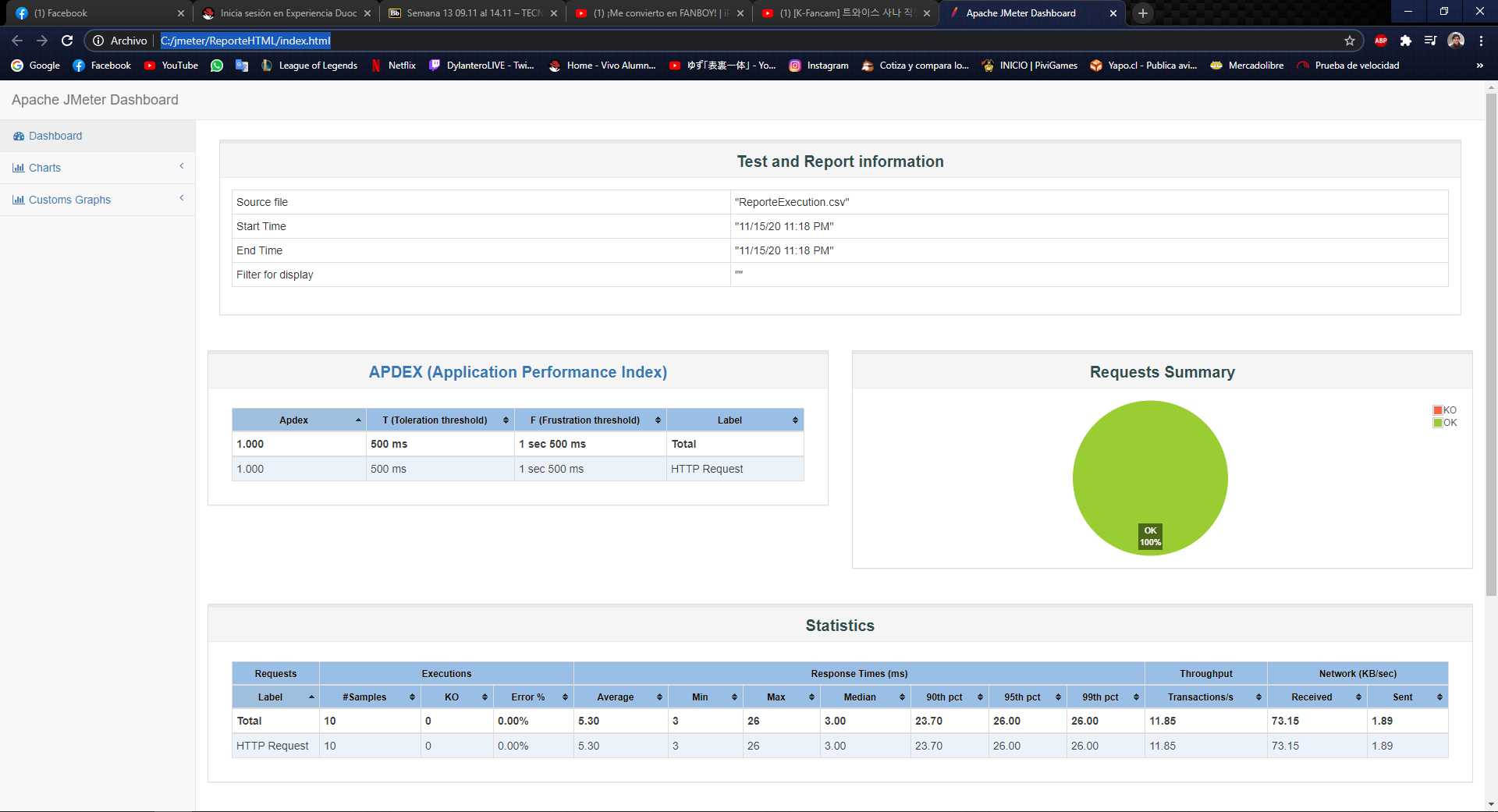
-Resultado de prueba con 10000 usuarios en tipo resumen



-Resultado de prueba con 10000 usuarios en tipo tabla, la prueba ya muestra errores de carga y colapsos por la cantidad de usuarios simultáneos 

4-Muestra reporte HTML

-Captura de pantalla de reporte HTML



# Informe resultados casos de prueba ejecutados y propuestas de mejoras

En este informe se propondrá al equipo desarrollador del programa las mejoras que se podrían hacer para arreglar los casos de prueba fallidos en ambas iteraciones.

|  |  |  |
| --- | --- | --- |
| **N° de Prueba** | **Descripción** | **Propuesta** |
| RF-8/Buscar Producto | El usuario ingresa a la opción Home dentro del menú principal, el usuario ingresa en la opción buscar el nombre de un producto, el usuario presiona el botón “Search” (Si el nombre del producto ingresado no se encuentra en la lista, el sistema deberá mostrar “producto no encontrado” como mensaje al usuario). | Nuestra propuesta para mejorar este módulo es principalmente usar otro tipo de método para buscar el producto, ya que el módulo intenta buscar el producto no por su nombre sino por su código de identificación, también modificar el código del módulo en general ya que al momento de buscar un producto en sistema se cae. |
| RF-10/Cerrar sesión | Una vez el proceso de iniciar sesión ha sido completado, la página se refrescará para mostrar el nuevo menú el cual incluye el botón "Cerrar sesión". Una vez el usuario decida que quiere abandonar la sesión, este deberá presionar el botón "Cerrar sesión" para poder salir. | La propuesta para este módulo es solamente hacer que el código te redireccione a la página principal una vez se ha cerrado sesión, ya que al momento de cerrar sesión se muestra una página en blanco. Al no saber realmente que es lo que sucede al cerrar sesión gracias al problema dicho anteriormente, no se puede dar un feedback verdadero, ya que podría haber más problemas que solamente una pestaña en blanco. |
| RF-13/Precio igual a 0 | Para modificar el precio de un producto o agregar un producto nuevo se debe iniciar sesión, buscar el módulo de "Producto" en el menú principal, Ingresar los datos en los campos requeridos y presionar el botón "agregar" o "modificar" dependiendo que se quiera hacer | Se propone agregar una variable que impida que el precio de un producto sea igual a 0, como también modificar la BD para que el campo de “Precio” sea mayor a 0. |
| RF-14/Stock de Producto nuevo Igual a 0 | Para modificar el stock de un producto o agregar un producto nuevo se debe iniciar sesión, buscar el módulo de "Producto" en el menú principal, Ingresar los datos en los campos requeridos y presionar el botón "agregar" o "modificar" dependiendo que se quiera hacer | Al igual que la propuesta del precio, se sugiere agregar alguna variable, pero con una importante modificación. La variable debería hacer que solamente un producto agregado recientemente tenga la restricción de un stock igual a 0, ya que un producto que tenía un stock anteriormente pero ahora está en 0 se vería afectado. |

# Conclusión

De acuerdo con lo señalado y los resultados demostrados, podemos acertar que sin duda la calidad de un sistema es de gran importancia para el desarrollo del mismo, ya que es indispensable para un profesional entregar un producto de alta calidad y valor, los cuales se pueden pulir gracias a diferentes técnicas de calidad como las señaladas en este informe.

# Bibliografía