

**Portafolio de Evidencias Testing Fundamentals**

**Tipos de software testing.**

Dalyn Yenisey Lara Cortés

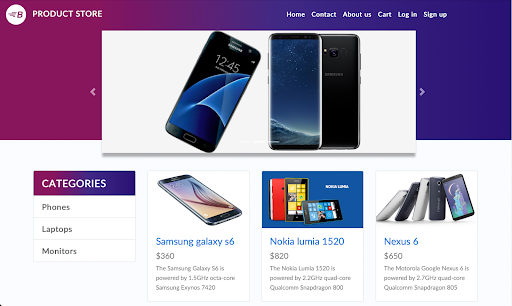
06 de marzo de 2022

# Tipos de software testing.

**Actividad 1**

La siguiente página “Demoblaze” es una página de pruebas con completa funcionalidad. Para esta actividad se plantea dividir 3 sub-funcionalidades:

1. Funcionalidad de Login.
2. Funcionalidad de carrito de compras.
3. Página para mostrar productos.

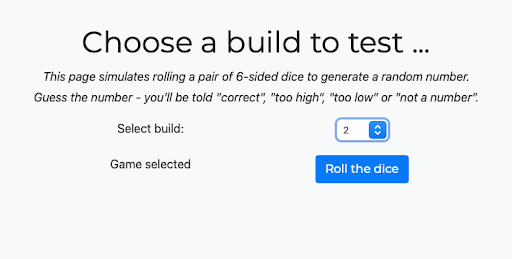


* De acuerdo a esta subdivisión deberás proponer 3 especificaciones que describan características funcionales y 3 que describan 3 características no funcionales para cada una de las sub-funcionalidades propuestas.

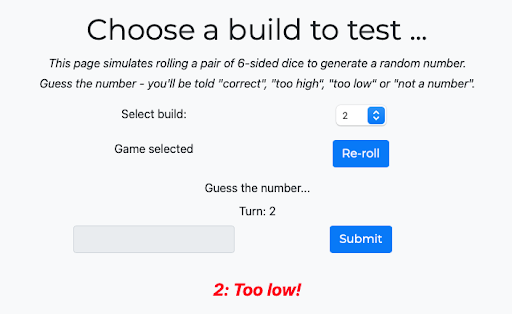
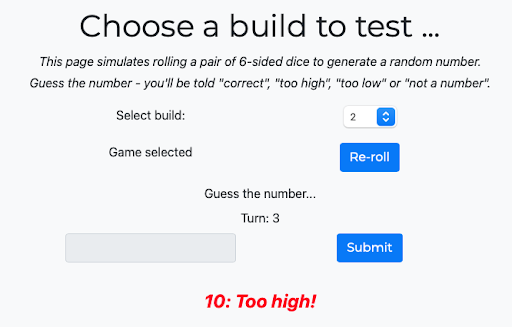
|  |  |  |
| --- | --- | --- |
| Funcionalidad | Especificación Funcional | Especificación no Funcional |
| Login de Usuario | 1. El usuario debe llenar los campos de username y password para poder proceder con el login. | 1. La página debe ser capaz de funcionar en diversos navegadores y distintas plataformas. |
| 1. El usuario debe obtener una alerta en caso de que uno de los campos sea erróneo. | 1. El intercambio y verificación de información debe llevarse a cambo de forma segura. |
| 1. El usuario debe ser capaz de ingresar al sistema en caso de que sus datos sean correctos. | 1. Los datos personales del usuario no deben ser expuestos a aplicaciones de terceros. |
| Carrito de Compras | 1. El usuario debe ser capaz de agregar y eliminar productos a su carrito de compras. | 1. Los datos personales del usuario no deben ser compartidos con terceros. |
| 1. El usuario debe poder concretar su compra una vez complete el formulario correctamente. | 1. Se debe considerar el uso de la página en dispositivos móviles. |
| 1. El carrito de compras debe mostrar el costo total por los productos incluidos en el carrito y actualizarse con cada cambio. | 1. Se deben implementar estándares Web y tomar en cuenta protocolos HTTP y HTTPS. |
| Pantalla Mostrar Productos | 1. El sistema debe mostrar la información de manera legible y ordenada. (Imagen del producto de buen tamaño, título grande y en negritas, etc) | 1. El tiempo de espera de carga de la página debe ser mínima. |
| 1. El sistema debe proporcionar la opción añadir al carrito. | 1. Debe asegurarse la disponibilidad de la información presentada en la pantalla. (Descripciones de productos, archivos multimedia, etc.) |
| 1. El sistema debe permitir filtrar los productos según la categoría a la que pertenezca. | 1. Implementar mecanismos de seguridad que no permitan la manipulación de la información por externos. |

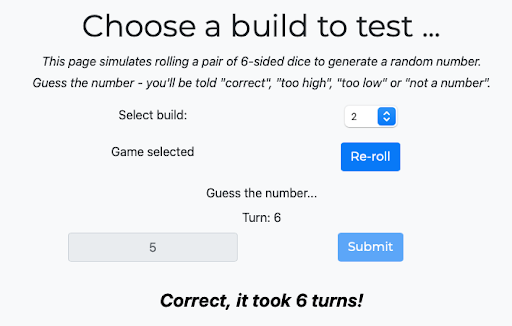
**Actividad 2**

La siguiente página muestra un juego simple con la siguiente interfaz.

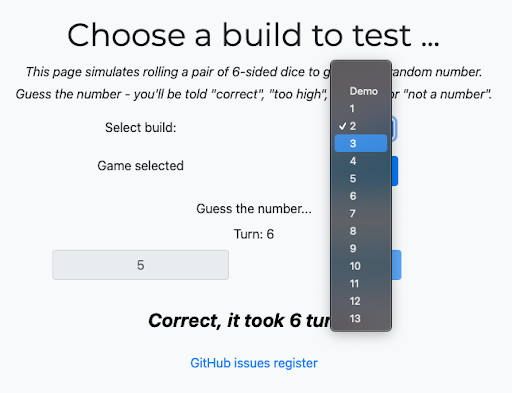


El objetivo del juego es que al presionar el botón “Roll the dice”, el sistema debería calcular un número aleatorio entre 1 y 12. Una vez presionado el botón, el usuario debe de adivinar qué número fue calculado. Si el número dado es más bajo se desplegará el mensaje “Very Low”, si es más alto se despliega “Very High”. El juego acaba cuando el usuario adivina el número calculado y se muestra el número de intentos que tomó para poder adivinarlo.



La aplicación tiene hasta 13 distintas versiones que pueden ser seleccionadas en el dropdown “Select build”.



Cada una de las versiones de este juego tiene al menos un defecto. Por lo que para la entrega de este ejercicio deberás seleccionar al menos 8 versiones (excepto “Demo”) y se debe tratar de proponer la causa raíz de cada uno de ellos en la siguiente tabla.

|  |  |  |
| --- | --- | --- |
| Versión | Defecto Encontrado | Causa Raíz Propuesta |
| 12 | La aplicación nunca te indica cuál es la respuesta correcta, las únicas respuestas que brinda son “Too high” y “Too Low”. | La sentencia de lógica que identifica si los valores son iguales no fue definida y el mensaje de “Correct” no se implementó. |
| 7 | Permite usar valores negativos y mayores a doce. | Se valida que la entrada sea numérica pero no se valida si el número ingresado esta dentro del rango de valores aceptados. |
| 3 | La aplicación nunca te indica cuál es la respuesta correcta, las únicas respuestas que brinda son “Too high” y “Too Low”. | La sentencia de lógica que identifica si los valores son iguales no fue implementada y el mensaje de “Correct” no se implementó. |
| Permite introducir valores negativos y mayores a doce. | No se valida si el número ingresado esta dentro del rango de valores aceptados. |
| 9 | Permite ingresar letras y caracteres especiales. | No existe validación de que el valor ingresado sea numérico. |
| La respuesta correcta puede ser un número igual o menor a 0, así como también mayor a 12. | El rango de valores válidos de la función generadora de la respuesta no fue definido correctamente, por lo que permite que la respuesta sea un número negativo, o positivo sin importar el rango. |
| 13 | En caso de ingresar un carácter especial, un número negativo o un número superior a 13, la aplicación se reinicia. | El manejo del ingreso de caracteres especiales, números negativos o números superior a 13 no se implementó correctamente. Siempre manda a llamar a la función principal para cada uno de estos casos, reiniciando la aplicación. |
| 5 | La respuesta correcta puede ser un número igual o menor a 0, así como también mayor a 12. | El rango de valores válidos de la función generadora de la respuesta no fue definido correctamente, por lo que permite que la respuesta sea un número negativo, o positivo sin importar el rango. |
| 8 | Permite introducir valores negativos y mayores a doce. | No se valida si el número ingresado está dentro del rango de valores aceptados. |
| 4 | La respuesta correcta puede ser un número igual o menor a 0, así como también mayor a 12. | El rango de valores válidos de la función generadora de la respuesta no fue definido correctamente, por lo que permite que la respuesta sea un número negativo, o positivo sin importar el rango. |

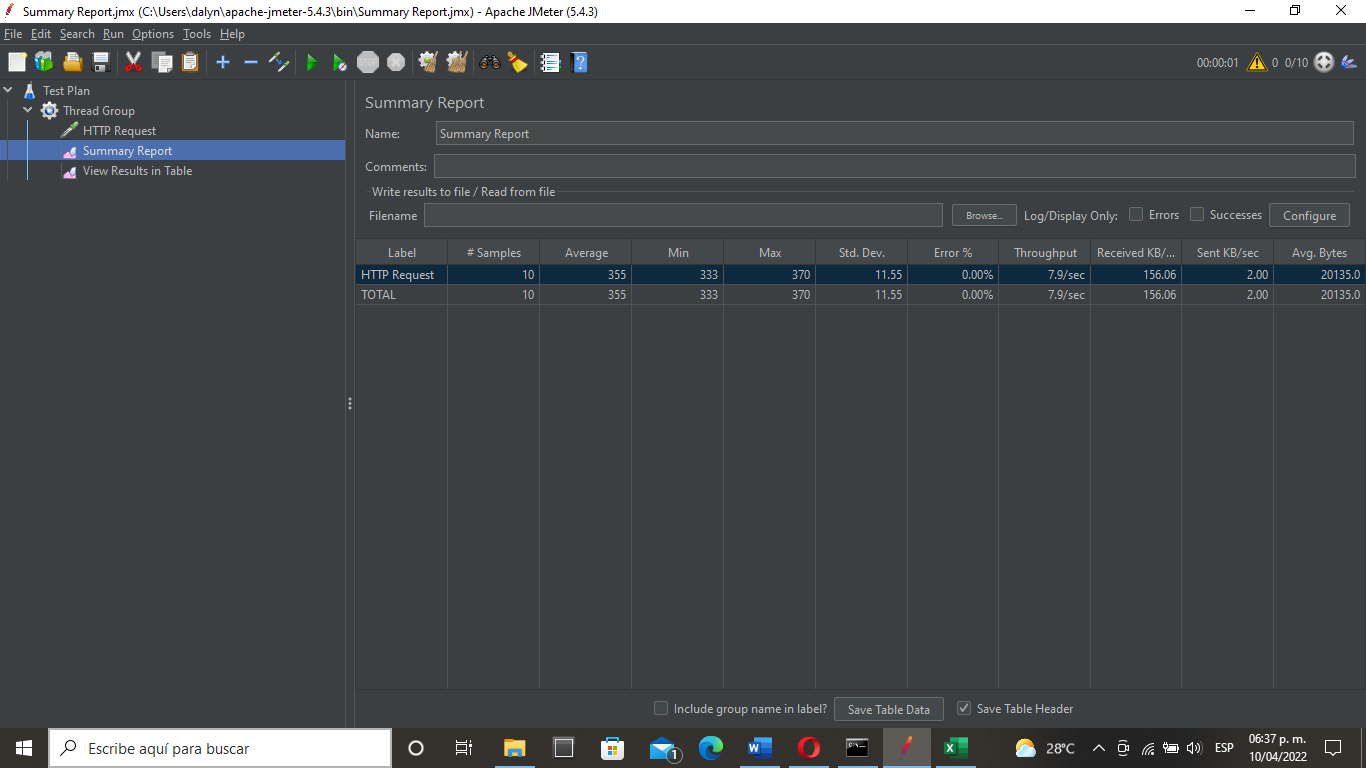
**Actividad 3**

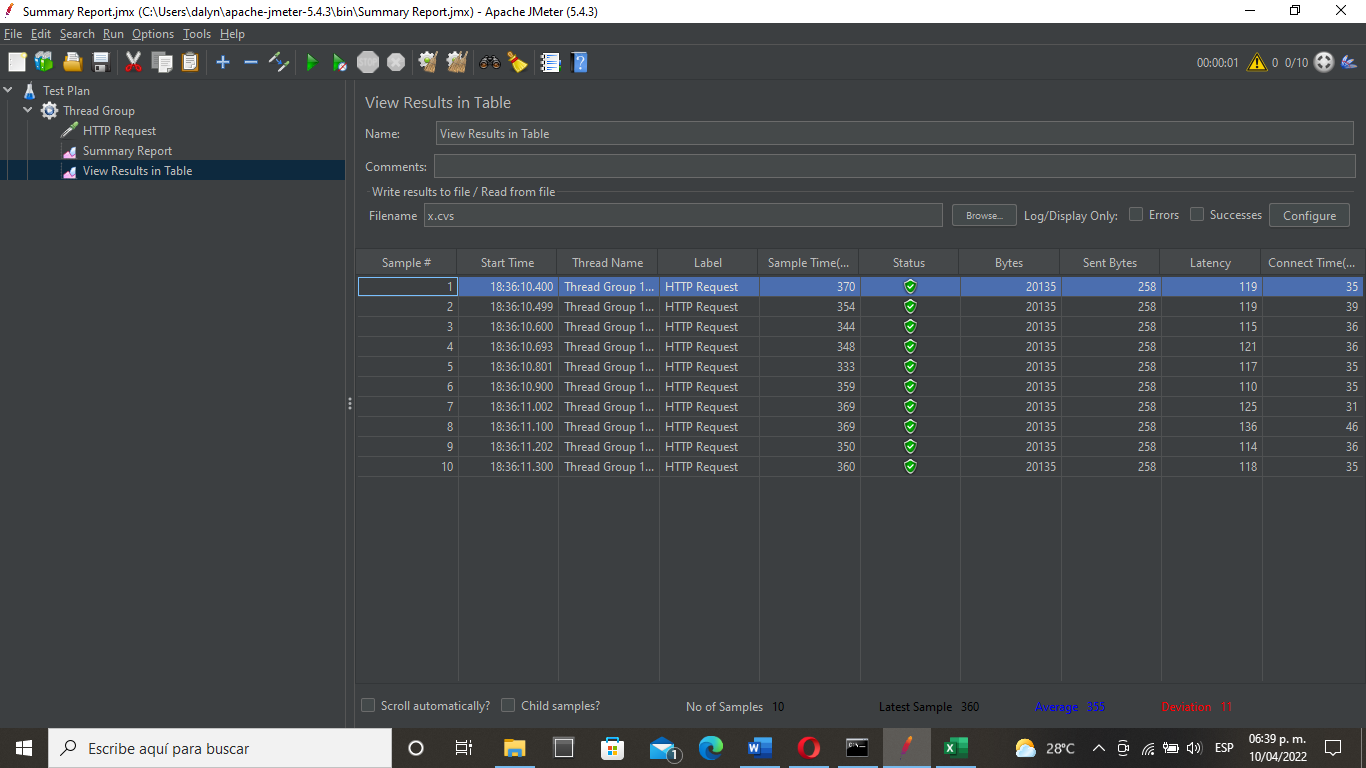
Para el último ejercicio se utilizará la herramienta de jMeter para realizar pruebas en la página Demoblaze.

El objetivo es simular la petición de 10, 100 y 1000 usuarios, y reportar los siguientes datos en la tabla de abajo.

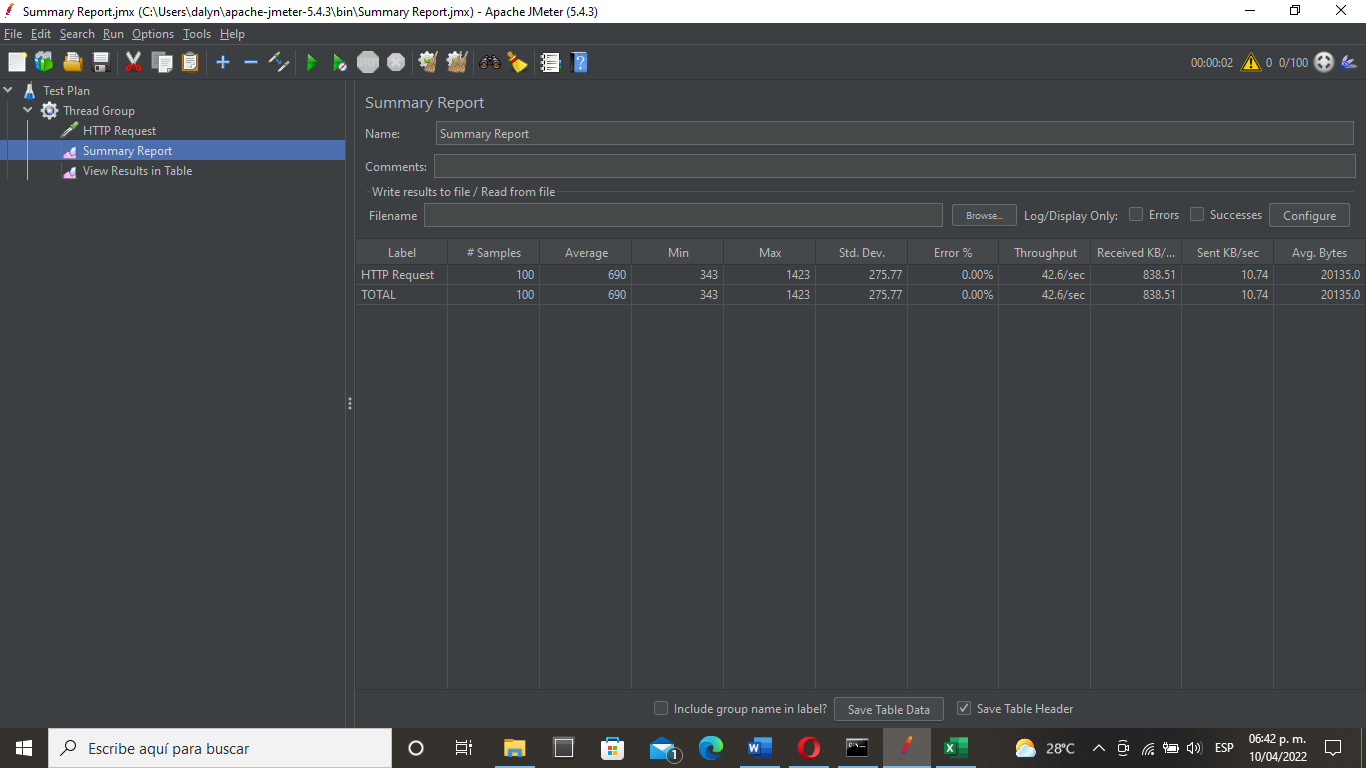
|  |  |  |  |
| --- | --- | --- | --- |
| # Usuarios | Latencia Promedio | Tiempo de Carga Promedio | # Petici0nes Correctas / Incorrectas |
| 10 | 0.1535 s | 0.355 s | 10- Correctas  0- Incorrectas |
| 100 | 0.31327 s | 0.69 s | 10- Correctas  0 - Incorrectas |
| 1000 | 6.43586 s | 745.544 s | 635- Correctas  347- Incorrectas |

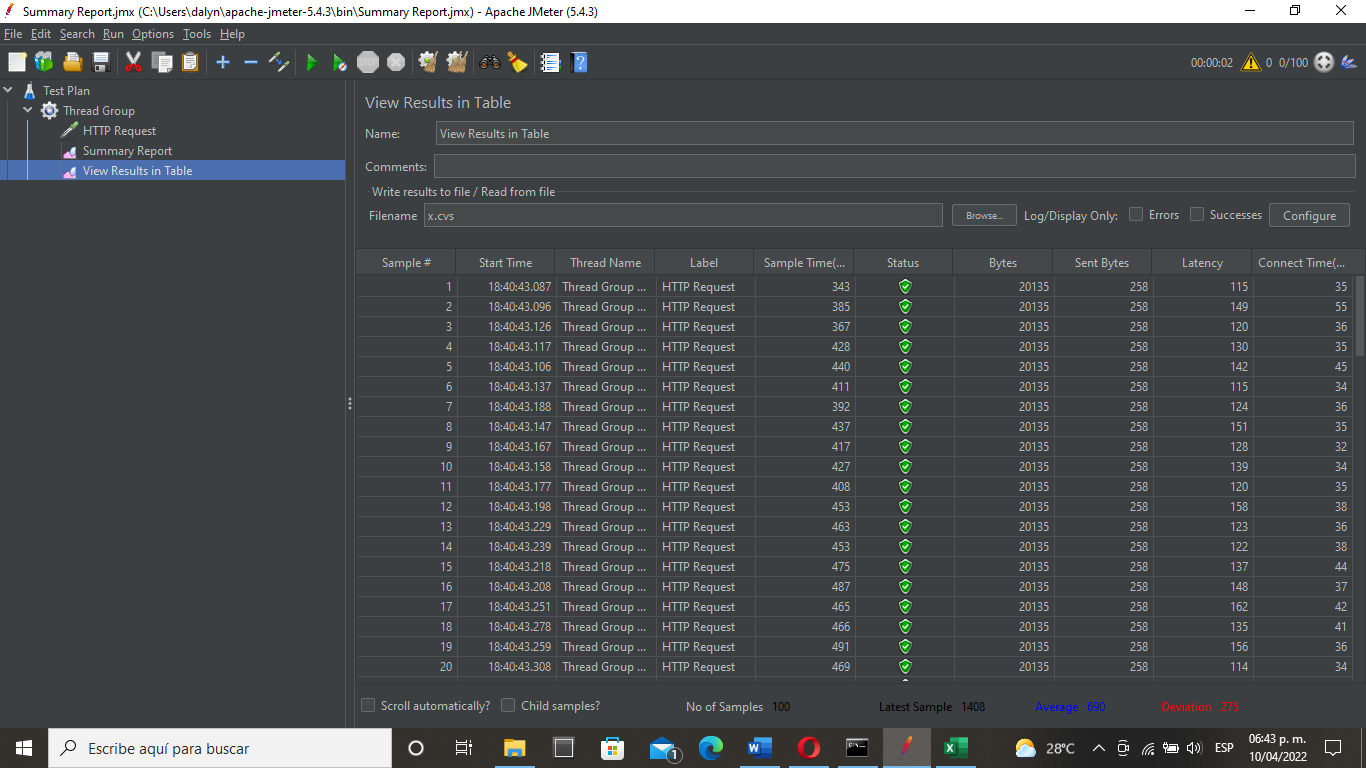
**10 Usuarios**





**100 Usuarios**





**1000 usuarios**

