Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Chihuahua

Integración de seguridad informática en redes y sistemas de software
(Gpo 401)
TC2007B

Profesores:
Enrique Antonio Sanchez Ordoñez
Faustino Bejarano Romero
Raime Alejandro Bustos Gardea

Tarea: Pruebas de software.

Angélica Alemán Cabrales
A01246666
Ingeniería en Tecnologías Computacionales
25 de Agosto de 2022

1. DOCUMENTACIÓN DE PRUEBAS.

1.1. Pruebas unitarias.

Las pruebas unitarias consisten en verificar el comportamiento de las unidades más pequeñas de una aplicación, como por ejemplo, una clase o incluso un método de clase en los lenguajes orientados a objetos, o un procedimiento o función en los lenguajes procedimentales y funcionales. Estas pruebas brindan una retroalimentación casi instantánea sobre el diseño y la implementación del código. Algunos de los beneficios de estas pruebas es el ahorro de tiempo en las pruebas de regresión, ayuda a obtener menos errores que a su vez representan más tiempo para crear valor, y se reducirá la carga del personal de soporte. (Digite, 2021).

1.2. Pruebas de integración.

Las pruebas de integración son aquellas que buscan probar diferentes módulos de una aplicación de software como grupo (en conjunto). Una aplicación de software se compone de diferentes submódulos que trabajan juntos para diferentes funcionalidades, por lo que el propósito de estas pruebas es validar la integración de diferentes módulos juntos e identificar los errores y problemas relacionados con ellos. Existen muchos tipos o enfoques diferentes en cuanto a las pruebas de integración. Los enfoques más populares y de uso frecuente son las pruebas de integración Big Bang, las pruebas de integración descendente, las pruebas de integración ascendente y las pruebas de integración incremental. (Lee, 2020).

1.3. Pruebas sistema.

Las pruebas del sistema tienen como propósito ejercitar un sistema, comprobando la integración del sistema de información globalmente, asegurando el correcto funcionamiento de las interfaces entre los diferentes subsistemas que lo conforman y con el resto de sistemas de información con los que se comunica. Básicamente, buscan probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones se cumplen (ya sean técnicas o funcionales). (Cillero M., 2016)

1.4. Pruebas de regresión.

Las pruebas de regresión buscan confirmar que un programa existente o un cambio de código no ha afectado negativamente las características ya existentes.. Son solo una selección completa o parcial de casos de prueba anteriormente completados y repetidos para asegurar que las funcionalidades existentes funcionen adecuadamente.Estas pruebas son útiles para garantizar que el código antiguo sigue funcionando una vez que se realicen cambios (Ebooks, 2020).

1.5. Pruebas automatizadas.

Las pruebas automatizadas aplican herramientas de software para automatizar el proceso (manual) de revisión y validación de un producto de software. Estas pruebas garantizan la calidad en todas las fases del desarrollo, pues aseguran que las confirmaciones nuevas no introducen ningún error, lo que significa que el software sigue siendo implementable. (Atlassian, 2022).

1.6. Pruebas bajo condiciones frontera.

En pruebas de software, se refiere a los errores que los usuarios no suelen encontrar. Se trata de una situación o problema que ocurre sólo en un extremo (máximo o mínimo) de un parámetro operativo.

2. HERRAMIENTAS PARA PRUEBAS AUTOMATIZADAS.

2.1. Herramientas de pruebas automatizadas.

Nombre	Tipo de prueba	Licencia	Plataforma	Particularidades
VectorCAST	Integración	Comercial	C o C++	• Valida los sistemas integrados críticos para la seguridad y el negocio • Se usa ampliamente en industrias financieras, dispositivos médicos, controles industriales, ferrocarriles.
Citrus	Integración	Open Source	Java	• Establece secuencia de mensajes y crea mensajes de error • Enviar y recibir mensajes • Espere el mensaje y active otro mensaje
LDRA	Integración	Open Source	C C++ Java Ada95	• Las pruebas se pueden generar y ejecutar fácilmente • Variedad de soporte para que las pruebas tengan un entorno común para una amplia gama de proyectos
FitNesse	Integración	Open Source	Java C# Python	• No requiere instalación por separado • Permite validar los requisitos con la implementación del software real

TESSY	Integración	Comercial	C C++	• Genera un informe de prueba para el resultado de la ejecución de la prueba • Admite cobertura de código sin esfuerzo adicional
Embunit	Unitaria	Comercial	C C++	• Crea el código para las pruebas de manera automática
MochaJS	Unitaria	Open Source	JavaScript	• Implementado en nodejs • Soporte para diferentes navegadores • Proporciona una base limpia para desarrollar pruebas • Utilizar cualquier librería
Junit	Unitaria	Open Source	Java	• Permite escribir casos de prueba mientras se desarrolla el software, esto puede ayudar a realizar pruebas tempranas y detectar problemas. • Soportado por todos los entornos de desarrollo integrados
NUnit	Unitaria	Open Source	.NET	• Proviene de un software llamado Junit, utilizado para Java
SimpleTest	Unitaria	Open Source	Php	• Permite crear los casos de prueba en scripts ejecutables
SpiraTest	Sistema	Comercial	Ruby	• Maneja tareas y encuentra bugs de manera efectiva • Agrega problemas y tareas al usuario designado en cierta área
Rapise	Sistema	Comercial	Java .NET Flash Qt SWT Ajax WPF Silverlight	• Admite la ejecución en paralelo. • Cuenta con automatización asistida. • Los test que realiza están basados en los requisitos y en modelos, son parametrizados y evalúan la seguridad. • Conformidad Unicode.
Leapworks	Sistema	Comercial	Cloud SaaS Web Windows	• Admite la ejecución en paralelo • Gestión de requisitos • Tests basados en los

				requisitos • Tests de seguridad • Verificación de script de tests
LambdaTest	Sistema	Comercial	JIRA Asana Github Trello Slack	• Pruebas interactivas en vivo a través de VM alojado en la nube LambdaTest • Prueba automática de navegador • Integración con varias herramientas de CI / CD
Katalon Studio	Sistema	Comercial	Eclipse	• Permite automatizar aplicaciones web, aplicaciones móviles y pruebas API. • Grabación y reproducción para aplicaciones web y móviles. • Permite ejecutar pruebas en una configuración diferente.

3. REPORTE GERENCIAL

HISTORIA DE USUARIO.

Como usuario de la aplicación, quiero jugar piedra, papel o tijera para sumar puntos.

CASO DE PRUEBA 1

- Identificador: TEST-001
- Nombre: Presionar botón de “Piedra” (testOnClickRock)
- Escenario:
  - Dado una eleccion de “piedra” del usuario
  - Cuando el usuario presiona la imagen de piedra en la interfaz
  - Entonces la etiqueta de ganador debe cambiar forzosamente
- Instrucciones:
  - Abrir la aplicación
  - Presionar la imagen/botón de piedra en la interfaz de usuario
  - La etiqueta que anuncia el ganador no debe ser igual a “Winner: “, sino que tiene que cambiar para mostrar el ganador del juego
- Entradas:
  - entrada 1: clickRock (botón de piedra)
- Salida Esperada:
  - resultado 1: “Winner: —” winnerLabel debe actualizarse

CASO DE PRUEBA 2

- Identificador: TEST-002
- Nombre: Presionar botón de “Papel” (testOnClickPaper)
- Escenario:
  - Dado una eleccion de “papel” del usuario
  - Cuando el usuario presiona la imagen de papel en la interfaz
  - Entonces la etiqueta de ganador debe cambiar forzosamente
- Instrucciones:
  - Abrir la aplicación
  - Presionar la imagen/botón de papel en la interfaz de usuario
  - La etiqueta que anuncia el ganador no debe ser igual a “Winner: “, sino que tiene que cambiar para mostrar el ganador del juego
- Entradas:
  - entrada 1: clickPaper (botón de papel)
- Salida Esperada:
  - resultado 1: “Winner: —” winnerLabel debe actualizarse

CASO DE PRUEBA 3

- Identificador: TEST-003
- Nombre: Presionar botón de “Tijeras” (testOnClickScissors)

- Escenario:
  - Dado una eleccion de “tijeras” del usuario
  - Cuando el usuario presiona la imagen de tijeras en la interfaz
  - Entonces la etiqueta de ganador debe cambiar forzosamente
- Instrucciones:
  - Abrir la aplicación
  - Presionar la imagen/botón de tijeras en la interfaz de usuario
  - La etiqueta que anuncia el ganador no debe ser igual a “Winner: “, sino que tiene que cambiar para mostrar el ganador del juego
- Entradas:
  - entrada 1: clickScissors (botón de tijeras)
- Salida Esperada:
  - resultado 1: “Winner: —” winnerLabel debe actualizarse

CASO DE PRUEBA 4

- Identificador: TEST-004
- Nombre: Presionar botón de “Restart” (testOnClickRestart)
- Escenario:
  - Dado una eleccion de “restart” del usuario
  - Cuando el usuario presiona el botón de restart en la interfaz
  - Entonces la etiqueta de ganador debe ser “Winner: “, ya que es el inicio del juego donde no existe un ganador aún
- Instrucciones:
  - Abrir la aplicación
  - Presionar el botón de restart en la interfaz de usuario
  - La etiqueta que anuncia el ganador debe ser igual a “Winner: “
- Entradas:
  - entrada 1: clickRestart (botón de restart/reinicio)
- Salida Esperada:
  - resultado 1: winnerLabel = “Winner: “

CASO DE PRUEBA 5

- Identificador: TEST-005
- Nombre: Crear respuesta de la computadora (testComputerAnswer)
- Escenario:
  - Dado una respuesta del usuario (piedra, papel o tijera)
  - Cuando el programa calcula una respuesta para la computadora según el número aleatorio
  - Entonces ese valor/respuesta se regresa para poder utilizarse
- Instrucciones:
  - Abrir la aplicación
  - Crear el número aleatorio
  - Obtener la respuesta de la computadora

    - Regresar valor
- Entradas:
    - entrada 1: 1
    - entrada 2: 2
    - entrada 3: 3
- Salida Esperada:
    - resultado 1: Piedra (“rock”)
    - resultado 2: Papel (“paper”)
    - resultado 3: Tijeras (“scissors”)

CASO DE PRUEBA 6

- Identificador: TEST-006
- Nombre: Misma elección de jugada (testGameRockRock)
- Escenario:
    - Dado una eleccion “x” por el usuario, y una elección “x” por la computadora (misma elección)
    - Cuando se realiza el juego
    - Entonces se obtiene el caso número 1 como resultado (empate)
- Instrucciones:
    - Abrir la aplicación
    - Escoger una jugada presionando la imagen correspondiente (piedra, papel o tijera)
    - Obtener la respuesta de la computadora (Automático)
    - Validar que el resultado del juego sea 1 (por el caso 1)
- Entradas:
    - entrada 1: Piedra (clickRock)
    - entrada 2: Papel (clickPaper)
    - entrada 3: Tijera (clickScissors)
- Salida Esperada:
    - resultado 1: 1

CASO DE PRUEBA 7

- Identificador: TEST-007
- Nombre: El usuario gana un punto (testGamePaperRock)
- Escenario:
    - Dado una elección “x” por el usuario, y una elección “y” por la computadora (elección perdedora)
    - Cuando se realiza el juego
    - Entonces se obtiene el caso número 2 como resultado (usuario ganador)
- Instrucciones:
    - Abrir la aplicación

    - Escoger una jugada presionando la imagen correspondiente (piedra, papel o tijera)
    - Obtener la respuesta de la computadora (Automático)
    - Validar que el resultado del juego sea 2 (caso 2)
- Entradas:
    - entrada 1: Piedra (clickRock)
    - entrada 2: Papel (clickPaper)
    - entrada 3: Tijera (clickScissors)
- Salida Esperada:
    - resultado 1: 2

CASO DE PRUEBA 8

- Identificador: TEST-008
- Nombre: La computadora gana un punto (testGameScissorsPaper)
- Escenario:
    - Dado una elección “y” por el usuario, y una elección “x” por la computadora (elección ganadora)
    - Cuando se realiza el juego
    - Entonces se obtiene el caso número 3 como resultado (computadora ganadora)
- Instrucciones:
    - Abrir la aplicación
    - Escoger una jugada presionando la imagen correspondiente (piedra, papel o tijera)
    - Obtener la respuesta de la computadora (Automático)
    - Validar que el resultado del juego sea 3 (caso 3)
- Entradas:
    - entrada 1: Piedra (clickRock)
    - entrada 2: Papel (clickPaper)
    - entrada 3: Tijera (clickScissors)
- Salida Esperada:
    - resultado 1: 3

CASO DE PRUEBA 9

- Identificador: TEST-006
- Nombre: Actualización en los puntajes (testScoresChanging)
- Escenario:
    - Dado un caso de juego determinado
    - Cuando se calculan los resultados de puntajes
    - Entonces se obtienen ambos puntajes y el ganador de la partida
- Instrucciones:
    - Abrir la aplicación
    - Generar un caso de juego (1, 2 o 3)

    - Obtener puntajes y ganador
- Entradas:
    - entrada 1: Caso de juego (caseGame)
- Salida Esperada:
    - resultado 1: 1 (expectedResultPlayer = 1)
    - resultado 2: 0 (expectedResultComputer = 0)
    - resultado 3: “Player”

3.2. Reporte gerencial.

ID	Nombre	Escenario	Salida esperada	Salida real	P/F status
001	Presionar botón de “Piedra” testOnClickRock	Dado una elección de “piedra” del usuario, cuando el usuario presiona la imagen de piedra en la interfaz, entonces la etiqueta de ganador debe cambiar forzosamente	“Winner: “ + string	“Winner: “ + string	100% SÍ
002	Presionar botón de “Papel” testOnClickPaper	Dado una eleccion de “papel” del usuario, cuando el usuario presiona la imagen de papel en la interfaz, entonces la etiqueta de ganador debe cambiar forzosamente	“Winner: “ + string	“Winner: “ + string	100% SÍ
003	Presionar botón de “Tijeras” testOnClick Scissors	Dado una eleccion de “tijeras” del usuario, cuando el usuario presiona la imagen de papel en la interfaz. entonces la etiqueta de ganador debe cambiar forzosamente	“Winner: “ + string	“Winner: “ + string	100% SÍ
004	Presionar botón de “Restart” testOnClick Restart	Dado una eleccion de “restart” del usuario, cuando el usuario presiona el botón de restart en la interfaz, entonces la etiqueta de ganador debe cambiar forzosamente	“Winner: “	“Winner: “	100% SÍ
005	Crear respuesta de computadora testComputer Answer	Dado una respuesta del usuario (piedra, papel o tijera), cuando el programa calcula una respuesta para la computadora según un número aleatorio, esa respuesta se regresa para poder utilizarse	“rock” “paper” “scissors”	“rock” “paper” “scissors”	100% SÍ

006	Misma elección de jugada testGame RockRock	Dado una eleccion “x” por el usuario, y una elección “x” por la computadora (misma elección), cuando se realiza el juego, entonces se obtiene el caso número 1 como resultado (empate)	1	1	100% SÍ
007	El usuario gana un punto testGame PaperRock	Dado una elección “x” por el usuario, y una elección “y” por la computadora (elección perdedora), cuando se realiza el juego, entonces se obtiene el caso número 2 como resultado (usuario ganador)	2	2	100% SÍ
008	La computadora gana un punto testGame ScissorsPaper	Dado una elección “y” por el usuario, y una elección “x” por la computadora (elección ganadora), cuando se realiza el juego, entonces se obtiene el caso número 3 como resultado (computadora ganadora)	3	3	100% SÍ
009	Actualización en los puntajes testScores Changing	Dado un caso de juego determinado, cuando se calculan los resultados de puntajes, entonces se obtienen ambos puntajes y el ganador de la partida	1 0 “Player”	1 0 “Player”	100% SÍ

3.3. Repositorio de Github.

https://github.com/angelicaalemanc/RockPaperScissors

Referencias.

Digite. (2021, Mayo 20). *Prueba Unitaria: Un Tutorial Sobre Qué Es, Cómo Hacerlo + Herramientas Para Usar*. https://www.digite.com/es/agile/pruebas-unitarias/

Lee, G. (2020, Octubre 16). *Tipos de pruebas de software: diferencias y ejemplos. LoadView;* LoadView by Dotcom-Monitor. https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/

Cillero, M. (2016, Octubre 17). *Pruebas del Sistema; Manuel.cillero.es.* https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/pruebas/sistema/

Ebooks. (2020). *¿Qué es una prueba de regresión? Definición, casos de prueba (ejemplo).* Ebooks Online. https://ebooksonline.es/que-es-una-prueba-de-regresion-definicion-casos-de-prueba-ejemplo

Atlassian. (2022). *Pruebas de software automatizadas para la entrega continua.* Atlassian. https://www.atlassian.com/es/continuous-delivery/software-testing/automated-testing

Edge case. (2021, 8 de diciembre). *Wikipedia, La enciclopedia libre.* Fecha de consulta: 01:19, agosto 19, 2022 desde https://es.wikipedia.org/w/index.php?title=Edge_case&oldid=140220309.

My Server Name. (2012). *Las 10 mejores herramientas de prueba de integración para escribir pruebas de integración.* https://spa.myservername.com/top-10-integration-testing-tools-write-integration-tests

Software Testing Help. (2022, 7 agosto). 20 Most Popular Unit Testing Tools In 2022. Recuperado 24 de agosto de 2022, de https://www.softwaretestinghelp.com/unit-testing-tools/

Xcode Cloud. Recuperado el 23 de agosto del 2022, de https://developer.apple.com/xcode-cloud/

EZscript - a cross-platform interpreter. (2013, March 8). SourceForge. Retrieved August 24, 2022, from https://sourceforge.net/projects/ezscript/

NA. (NA). 50 Best Integration Testing Tools. 24/08/2022, de threat stack Sitio web: https://www.threatstack.com/blog/50-best-integration-testing-tools

Test Management Software - 30 Day Trial. (2022). Inflectra.com. https://www.inflectra.com/Landing/Test-Management.aspx?source=CapterraAd&listing=SpiraTest&utm_source=GetApp

JUnit - Overview. (s. f.). Recuperado de https://www.tutorialspoint.com/junit/junit_overview.htm#

Abhishek Kothari. (2021). 11 mejores herramientas y marco de pruebas unitarias. 25/8/2022, de GEEKFLARE Sitio web: https://geekflare.com/es/javascript-unit-testing/

Capterra. (2022, junio). Rapise. https://www.capterra.mx/software/132835/rapise#features

Shaik, H. K. (2021, November 8). Prueba unitaria con el Módulo unittest de python. Geekflare. Retrieved August 25, 2022, from https://geekflare.com/es/unit-testing-with-python-unittest/#