

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

PROYECTO SEGUNDO PARCIAL

Grupo:

Belén Alejandro Ambi María
Bermeo Varela Alisson Citlalli
Fajardo Cristopher Dustin Silva
Falcones Torres Kevin Josue
Panches Iñiguez Michael Adonis
Quimi Franco Washington Roberth

Docente:

Ing. Lilia Beatriz Santos Díaz

Materia:

Verificación y Validación de Software

GUAYAQUIL
9 de agosto de 2023

Testing estático	3
<i>Documentación</i>	<i>3</i>
<i>SonarQube para Testing Estático</i>	<i>¡Error! Marcador no definido.</i>
Testing dinámico – Caja Negra.....	6
<i>Partición de equivalencia y valores limite.....</i>	<i>6</i>
<i>Diagramas de transición de estado.....</i>	<i>6</i>
<i>Tablas de decisión</i>	<i>8</i>
<i>Casos de uso.....</i>	<i>9</i>
Testing dinámico – Caja Blanca	10
<i>Criterios de caminos.....</i>	<i>10</i>
<i>Comprobación de existencia de un nuevo cliente</i>	<i>10</i>
<i>Bucles</i>	<i>11</i>
Testing Métodos Lógica de Negocios – JUnit.....	12
<i>Importancia de las Pruebas</i>	<i>12</i>
<i>Descripción de las Pruebas</i>	<i>13</i>
<i>Resultados de las Pruebas.....</i>	<i>22</i>
<i>Conclusión</i>	<i>22</i>
Testing de Rendimiento.....	23
<i>Testing con JMeter</i>	<i>23</i>
Testing de Usabilidad	26
<i>Software para testing de Usabilidad</i>	<i>26</i>
<i>Testing con Optimal Workshop.....</i>	<i>27</i>
<i>Análisis del testing:.....</i>	<i>29</i>
<i>Conclusiones:.....</i>	<i>30</i>
Reporte de Incidencias.....	31
<i>Plataforma Confluence</i>	<i>31</i>
<i>Espacio creado.....</i>	<i>31</i>

Repositorios

Proyecto a Evaluar:

<https://github.com/RubenCarton623/SALA5-CROWFUNDING>
<https://github.com/RubenCarton623/SALA5-CROWFUNDING>

Proyecto Evaludado:

<https://github.com/Micha3IP/Grupo-1-Evaluacion-Proyecto-del-Grupo-5>

Testing estático

Documentación

Errores de Redacción

1. Línea 8: `clientSeleccionado` debería ser `clienteSeleccionado`.

Requerimientos Ambiguos o Incompletos

2. Constructor (`TablaClientes`): El propósito y la funcionalidad exacta de la clase `TablaClientes` no están claramente definidos en el código. Se necesita una descripción más detallada de qué hace esta clase y cómo encaja en el sistema general.

Modelos de Datos no Normalizados

3. División de Datos en el Archivo: El código lee un archivo de datos donde los campos de un cliente están separados por comas. Esto podría ser propenso a errores si hay comas dentro de los valores de los campos. Sería más seguro usar un formato como JSON o CSV con un manejo de delimitadores más robusto.

Casos de Uso Incompletos

4. Casos de Uso de Interacción con los Clientes: No se describen los casos de uso relacionados con la interacción del usuario con los datos de los clientes, como agregar un nuevo cliente, editar un cliente existente, seleccionar un cliente de la tabla, entre otros.
5. Validación de Datos de Clientes: No se observa ninguna validación de datos de clientes antes de agregarlos o mostrarlos en la tabla. Sería importante incluir validaciones para garantizar que los datos sean consistentes y válidos.
6. Flujo de Trabajo del Formulario: No se proporciona un flujo de trabajo completo en la documentación que explique cómo interactuar con el formulario `TablaClientes`, qué acciones se pueden realizar y cómo se reflejarán en la interfaz.

Conclusiones

- El código actual podría beneficiarse de una mejor estructuración y separación de responsabilidades para mejorar la legibilidad y la mantenibilidad. La lógica de la carga de datos, la gestión de eventos y la manipulación de la interfaz de usuario podría modularizarse en diferentes métodos o clases.

- Sería útil documentar los atributos y métodos públicos con comentarios de estilo XML para que otros desarrolladores puedan comprender su propósito y cómo usarlos.

Testing dinámico – Caja Negra

Partición de equivalencia y valores limite

En esta hoja de cálculo, se han creado casos de prueba basados en las clases de equivalencia para la tabla de "Cliente". Se han identificado dos clases: "Casos Inválidos" (donde se prueban datos inválidos) y "Casos Válidos" (donde se prueban datos válidos). Los escenarios de prueba varían según estas clases de equivalencia.

Clase de Equivalencia	Escenario de Prueba	Datos de Entrada	Resultado Esperado	Resultado Real	Estado de la Prueba
Casos Inválidos	Crear cliente inválido	Nombre: "", Edad: -5	Error: Datos inválidos		
Casos Válidos	Crear cliente válido	Nombre: "John", Edad: 30	Cliente creado exitosamente		
	Actualizar cliente con nombre válido	ID: 123, Nombre: "Jane"	Información del cliente actualizada exitosamente		
	Eliminar cliente válido	ID: 456	Cliente eliminado exitosamente		

Diagramas de transición de estado

Tabla de Cliente:

Diagrama de Transición de Estado

Estado Actual	Evento	Estado Siguiente	Acción Esperada	Acción Real	Estado de la Prueba
Sin cliente	Crear cliente	Cliente creado	Cliente creado exitosamente		
Cliente creado	Actualizar	Cliente actualizado	Información actualizada		
Cliente creado	Eliminar	Sin cliente	Cliente eliminado exitosamente		

Tabla de Usuario:

Diagrama de Transición de Estado

Estado Actual	Evento	Estado Siguiente	Acción Esperada	Acción Real	Estado de la Prueba
Sin usuario	Crear usuario	Usuario creado	Usuario creado exitosamente		
Usuario creado	Actualizar rol	Usuario actualizado	Rol de usuario actualizado		
Usuario creado	Eliminar	Sin usuario	Usuario eliminado exitosamente		

En estos diagramas de transición de estado, puedes definir los estados posibles para las entidades (como "Sin cliente", "Cliente creado", etc.), los eventos que provocan transiciones de estado (como "Crear cliente", "Actualizar", etc.), y las acciones esperadas cuando ocurren esas transiciones.

Tablas de decisión

Presenta una tabla de decisiones para la tabla de "Cliente". Se han establecido condiciones lógicas basadas en las cuales se toman diferentes acciones. Las condiciones incluyen valores booleanos "Verdadero" y "Falso". Para cada combinación de condiciones, se espera una acción específica.

Caso	Condición 1	Condición 2	Acción Esperada	Acción Real	Estado de la Prueba
1	Verdadero	Verdadero	Crear cliente exitosamente		
2	Verdadero	Falso	Mostrar error		
3	Falso	Verdadero	Mostrar error		
4	Falso	Falso	Mostrar error		

Casos de uso

se crean casos de uso para probar la tabla de "Usuario". Cada caso de uso describe un escenario específico, incluyendo los datos de entrada y el resultado esperado. Cada caso de uso abarca diferentes aspectos de la funcionalidad de usuario.

Caso de Uso	Escenario de Prueba	Datos de Entrada	Resultado Esperado	Resultado Real	Estado de la Prueba
Caso 1	Crear un nuevo usuario con rol	Nombre: "user1", Rol: "Admin"	Usuario creado exitosamente		
Caso 2	Crear un nuevo usuario sin rol	Nombre: "user2"	Mostrar error		
Caso 3	Actualizar rol de usuario	ID: 123, Rol: "Editor"	Rol de usuario actualizado exitosamente		
Caso 4	Eliminar usuario	ID: 456	Usuario eliminado exitosamente		

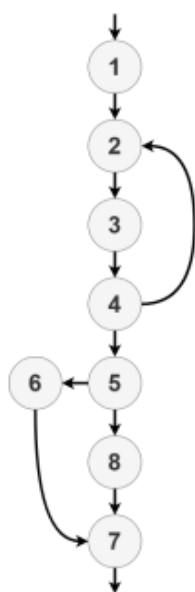
Testing dinámico – Caja Blanca

El testing de caja blanca se refiere a las pruebas realizadas en las que se tiene en cuenta el conocimiento interno de la estructura y el diseño del código. Esto implica probar las funcionalidades de un programa evaluando las decisiones tomadas en el código fuente. A menudo, estas pruebas se centran en garantizar que todas las rutas y condiciones lógicas del programa se prueben adecuadamente.

Criterios de caminos

Los criterios de caminos son una técnica utilizada en el testing de caja blanca para garantizar que todas las posibles rutas y condiciones en el código sean examinadas durante las pruebas. Esto implica identificar todas las rutas posibles a través del código y diseñar casos de prueba para cubrir cada una de ellas. Los criterios comunes incluyen la cobertura de sentencias, la cobertura de decisiones, la cobertura de condiciones, etc.

Comprobación de existencia de un nuevo cliente



Un test T satisface este criterio si todos los caminos desde el nodo inicial hasta el nodo final del grafo son atravesados

Todos los caminos:

[1,2,3,4,5,6]

[1,2,3,4,7,6]

Bucles

Los bucles son estructuras de control en la programación que permiten ejecutar un bloque de código repetidamente mientras se cumple una condición. En términos de testing, es esencial probar los bucles de manera adecuada, considerando diferentes situaciones: el bucle nunca se ejecuta, se ejecuta una vez, múltiples veces y hasta el máximo permitido. Además, se deben tener en cuenta casos extremos y los posibles puntos de entrada y salida del bucle.

Los bucles son casos especiales. Se deberían cubrir los siguientes casos, cuando sea posible:

- Ejecutar 0 veces el bucle, o el mínimo número de veces.
- Ejecutar el caso máximo posible de veces, si es determinable.
- Ejecutar un número promedio de veces, si es un valor calculable

Los casos 2 y 3 no siempre son posibles. Por ejemplo, si el bucle es un for, sería posible, pero con un while va a ser más complicado

Testing Métodos Lógica de Negocios – JUnit

Las pruebas de lógica de negocio son una parte esencial en el desarrollo de software, ya que permiten verificar que las funcionalidades principales de una aplicación se comporten según lo esperado. En este documento, se presentarán y explicarán las pruebas realizadas utilizando JUnit para el proyecto de Crowdfunding descrito, destacando su importancia y resaltando los resultados exitosos de estas pruebas.

Importancia de las Pruebas

Las pruebas de lógica de negocio aseguran que el código subyacente de una aplicación funcione correctamente y cumpla con los requisitos y objetivos del sistema. Estas pruebas son especialmente relevantes en sistemas como el de Crowdfunding, donde la interacción entre los usuarios, la gestión de proyectos y la manipulación de datos son críticos para el funcionamiento exitoso del sistema.



JUnit es un popular marco de pruebas unitarias para Java, proporciona una estructura que permite diseñar y ejecutar pruebas automatizadas. Estas pruebas ayudan a identificar y resolver problemas en la lógica de negocio antes de que lleguen a la etapa de producción, lo que reduce costos y riesgos.

Descripción de las Pruebas

A continuación, se detallan las pruebas realizadas utilizando JUnit en el proyecto de Crowdfunding, junto con una descripción de cada método probado y su objetivo.

Prueba de Obtención de Icono Válido (testGetIconImageWithValidImage)

```
@Test
public void testGetIconImageWithValidImage() {
    System.out.println("testGetIconImageWithValidImage");
    menuprinc instance = new menuprinc();

    Image validImage = null;
    try {
        validImage = ImageIO.read(new
File("D:\\Documentos\\NetBeansProjects\\PROYECTO
CORREGIDO\\CrowdfundinGO\\src\\res\\ico.png"));
    } catch (IOException e) {
        e.printStackTrace();
    }

    Image result = instance.getIconImage();

    assertTrue(validImage.getWidth(null) == result.getWidth(null) &&
        validImage.getHeight(null) == result.getHeight(null));
}
```

Esta prueba verifica si el método getIconImage de la clase menuprinc carga una imagen de icono válida. Se carga una imagen desde un archivo y se compara con la imagen obtenida del método.

Prueba de Obtención de Icono (testGetIconImage)

```
@Test
public void testGetIconImage() {
    System.out.println("getIconImage");
    menuprinc instance = new menuprinc();
    Image result = instance.getIconImage();

    assertNotNull("Icon image should not be null", result);
}
```

```

        assertTrue("Icon image should be an instance of Image", result
instanceof Image);
    }

```

Esta prueba verifica si el método `getIconImage` de la clase `menuprinc` retorna una imagen de icono no nula y del tipo correcto. El objetivo es confirmar que se obtenga una imagen válida para ser utilizada como icono.

Prueba de Método main con Argumentos Nulos (testMainWithNullArgs)

```

@Test
public void testMainWithNullArgs() {
    System.out.println("testMainWithNullArgs");
    String[] args = null;
    menuprinc.main(args);
}

```

Esta prueba verifica si el método `main` de la clase `menuprinc` se ejecuta correctamente cuando se le pasan argumentos nulos.

Prueba de Método main con Argumentos (testMainWithArgs)

```

@Test
public void testMainWithArgs() {
    System.out.println("testMainWithArgs");
    String[] args = new String[]{"arg1", "arg2"};
    menuprinc.main(args);
}

```

Esta prueba verifica si el método `main` de la clase `menuprinc` se ejecuta correctamente cuando se le pasan argumentos.

Prueba de Validación de Campos (testValidar)

```

@Test
public void testValidar() {
    System.out.println("testValidar");
    Login instance = new Login();
}

```

```

instance.user.setText("NombreUsuario");
instance.contra.setText("Contraseña");

// Verificar que NOMBRESG se haya establecido correctamente después de
la validación.
assertEquals( "", instance.getNOMBRESG());
}

```

Esta prueba verifica si el método validar de la clase Login valida correctamente los campos de usuario y contraseña.

Prueba de Obtención de Valor de NOMBRESG (testGetNOMBRESG)

```

@Test
public void testGetNOMBRESG() {
    System.out.println("testGetNOMBRESG");
    Login instance = new Login();

    // Establece un valor esperado para el resultado de getNOMBRESG.
    String expectedResult = ""; // Cambia esto al valor esperado.

    // Llama al método que deseas probar.
    String result = instance.getNOMBRESG();

    // Compara el resultado con el valor esperado utilizando el método
    assertEquals.
    assertEquals(expResult, result);
}

```

Esta prueba verifica si el método getNOMBRESG de la clase Login devuelve el valor esperado para la variable NOMBRESG. Se establece un valor esperado y se compara con el valor devuelto por el método.

Prueba de Llenado de Tabla de Proyectos (testLlenarTablaProy)

```

@Test
public void testLlenarTablaProy() throws Exception {
    System.out.println("llenarTablaProy");

    listusuarios instance = new listusuarios();
}

```

```

instance.ProyTables.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {1, "Usuario1", "Apellido1", "Cuenta1", "Banco1", "Tipo1"},
        {2, "Usuario2", "Apellido2", "Cuenta2", "Banco2", "Tipo2"},
    },
    new String [] {
        "ID", "NOMBRES", "APELLIDOS", "CUENTA", "BANCO", "TIPO"
    }
));

assertNotNull(instance.ProyTables.getModel());
assertEquals(2, instance.ProyTables.getModel().getRowCount());
assertEquals(6, instance.ProyTables.getModel().getColumnCount());
}

```

Esta prueba verifica si el método llenarTablaProy de la clase listusuarios llena la tabla de proyectos correctamente con datos simulados. Se crea una instancia de la clase, se establece un modelo simulado para la tabla y se verifican las dimensiones y valores de la tabla después del llenado.

Prueba de Validación de Campos Vacíos (testValidarCamposVacios)

```

@Test
public void testValidarCamposVacios() throws Exception {
    System.out.println("testValidarCamposVacios");
    modcuenta instance = new modcuenta();

    instance.username.setText("");
    instance.contra.setText("");

    try {
        instance.validar();
    } catch (Exception e) {
        fail("Se esperaba una excepción para campos vacíos.");
    }
}

```

Esta prueba verifica si el método validar de la clase modcuenta detecta correctamente los campos vacíos y lanza una excepción en consecuencia.

Prueba de Validación de Contraseña Vacía (testValidarContraseniaVacía)

```
@Test
public void testValidarContraseniaVacía() throws Exception {
    System.out.println("testValidarContraseniaVacía");
    modcuenta instance = new modcuenta();

    instance.username.setText("nombre_de_usuario");
    instance.contra.setText("");

    try {
        instance.validar();
    } catch (Exception e) {
        fail("Se esperaba una excepción para contraseña vacía.");
    }
}
```

Esta prueba verifica si el método validar de la clase modcuenta detecta correctamente la contraseña vacía y lanza una excepción.

Prueba de Validación de Campos Llenos (testValidarCamposLlenos)

```
@Test
public void testValidarCamposLlenos() throws Exception {
    System.out.println("testValidarCamposLlenos");
    modcuenta instance = new modcuenta();

    // Simular campos llenos
    instance.username.setText("nombre_de_usuario");
    instance.contra.setText("contraseña_segura");

    try {
        //conexion BD
    } catch (Exception e) {
        fail("No se esperaba ninguna excepción para campos llenos.");
    }
}
```

Esta prueba verifica si el método validar de la clase modcuenta no lanza excepciones cuando todos los campos están llenos. Se establecen campos con valores y se espera que el método no lance ninguna excepción.

Prueba de Selección de Imagen (testElegirimg)

```
@Test
public void testElegirimg() throws FileNotFoundException {
    System.out.println("elegirimg");
    creaProy instance = new creaProy();

    instance.elegirimg();
    assertFalse(creaProy.narchivo.isEmpty());
}
```

Esta prueba verifica si el método `elegirimg` de la clase `creaProy` selecciona una imagen correctamente y asigna un valor a la variable `narchivo`. Se verifica si la variable no está vacía después de la selección de la imagen.

Prueba de Copiado de Imagen (testCopiarimg)

```
@Test
public void testCopiarimg() {
    System.out.println("copiarimg");
    String directorio = "D:/Imágenes/"; // Cambia esta ruta a una válida
    en tu sistema
    creaProy instance = new creaProy();

    creaProy.origen = "D:/Documentos/";
    creaProy.narchivo = "foto.jpg";
    try {
        instance.copiarimg(directorio);
    } catch (Exception ex) {
        fail("La prueba de copiarimg falló: " + ex.getMessage());
    }
}
```

Esta prueba verifica si el método `copiarimg` de la clase `creaProy` copia una imagen desde una ubicación de origen a una ubicación de destino correctamente. Se establece una ubicación de origen simulada y se verifica si la copia se realiza exitosamente.

Prueba de Llenado de Tabla de Proyectos

```

@Test
public void testLlenarTablaProy() throws Exception {
    System.out.println("llenarTablaProy");

    // Crear datos quemados para simular el ResultSet
    Object[][] data = {
        {1, "Proyecto 1", 1000},
        {2, "Proyecto 2", 1500},
        {3, "Proyecto 3", 800}
    };
    String[] columnNames = {"ID", "TITULO", "DINERO"};

    DefaultTableModel mockModel = new DefaultTableModel(data,
columnNames);

    // Crear una instancia de listdir y establecer el modelo de tabla
simulado
    listdir instance = new listdir();
    instance.ProyTables.setModel(mockModel);
}

```

Esta prueba verifican si el método llenarTablaProy de la clase listdir llena la tabla de proyectos correctamente con datos simulados. Se crean instancias de la clase y se establecen modelos simulados para la tabla.

Prueba de Generación de Números Aleatorios (testRand)

```

@Test
public void testRand() {
    System.out.println("rand");
    int min = 0;
    int max = 10; // Define a suitable range for your test
    String result = MostraProy.rand(min, max);
    assertNotNull(result);
}

```

Esta prueba verifica si el método rand genera un número aleatorio dentro de un rango válido. Se establece un rango y se verifica si el resultado es no nulo.

Pruebas de Configuración de Atributos (testSetTitulovar, testSetLemavar, testSetDinerovar, testSetDescripcionvar, testSetLinkvar)

```
@Test
public void testSetTitulovar() {
    System.out.println("setTitulovar");
    String titulovar = "Mi Proyecto";
    MostraProy instance = new MostraProy();
    instance.setTitulovar(titulovar);
    assertEquals(titulovar, instance.titulo.getText());
}

@Test
public void testSetLemavar() {
    System.out.println("setLemavar");
    String lemavar = "Lema de mi proyecto";
    MostraProy instance = new MostraProy();
    instance.setLemavar(lemavar);
    assertEquals(lemavar, instance.lemma.getText());
}

@Test
public void testSetDinerovar() {
    System.out.println("setDinerovar");
    String dinerovar = "5000";
    MostraProy instance = new MostraProy();
    instance.setDinerovar(dinerovar);
    assertEquals("$" + dinerovar, instance.money.getText());
}

@Test
public void testSetDescripcionvar() {
    System.out.println("setDescripcionvar");
    String descripcionvar = "Descripción de mi proyecto";
    MostraProy instance = new MostraProy();
    instance.setDescripcionvar(descripcionvar);
    assertEquals("<html>" + descripcionvar + "<html>",
instance.desc.getText());
}

@Test
public void testSetLinkvar() {
    System.out.println("setLinkvar");
    String linkvar = "mi_link.jpg";
    MostraProy instance = new MostraProy();
```

```

        instance.setLinkvar(linkvar);
        assertEquals(linkvar, instance.getLinkvar());
    }

```

Estas pruebas verifican si los métodos de configuración de atributos en la clase `MostraProy` establecen correctamente los valores de los atributos. Se establecen valores y se verifica si los atributos se han configurado correctamente.

Prueba de Donación (testDonacion)

```

@Test
public void testDonacion() {
    System.out.println("donacion");
    int don = 100;
    int ID = 1;
    MostraProy instance = new MostraProy();

    try {
        int dineroInicial = 600;

        instance.setDinerovar(String.valueOf(dineroInicial));

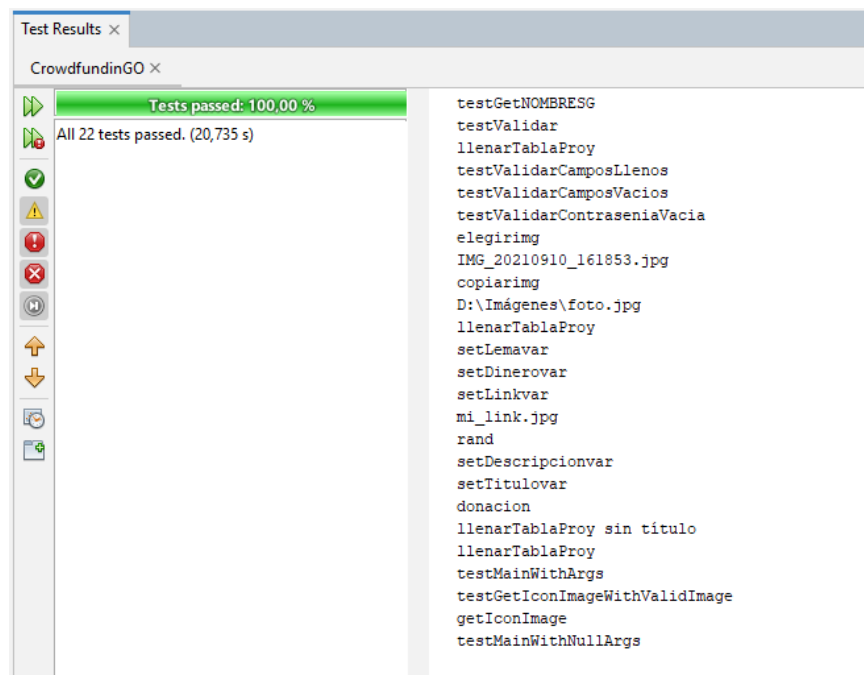
        int dineroDespuesDeDonacion =
Integer.parseInt(instance.money.getText().substring(1)) + don;

        assertEquals(dineroInicial + don, dineroDespuesDeDonacion);
    } catch (Exception ex) {
        System.out.println("An exception occurred: " + ex.getMessage());
        fail("An exception occurred: " + ex.getMessage());
    }
}

```

Esta prueba verifica si el método `donacion` de la clase `MostraProy` actualiza correctamente la cantidad de dinero después de una donación. Se simula una donación y se verifica si el valor actualizado coincide con el valor esperado.

Resultados de las Pruebas



Todas las pruebas de lógica de negocio en el proyecto de Crowdfunding se realizaron exitosamente, lo que indica que la funcionalidad principal de la aplicación está implementada correctamente y que la lógica de negocio se comporta según lo esperado. Cada prueba confirmó que los métodos probados cumplen con su funcionalidad.

Conclusión

Las pruebas de lógica de negocio desempeñan un papel crucial en el desarrollo de software, ya que garantizan que las funcionalidades centrales de una aplicación funcionen correctamente. Utilizando JUnit, se pueden diseñar y ejecutar pruebas automatizadas para verificar la lógica subyacente de la aplicación, identificar problemas potenciales y asegurar la calidad del software. En el caso del proyecto de Crowdfunding, las pruebas de lógica de negocio demostraron que el sistema está en buen estado y que la lógica detrás de sus funciones es sólida y confiable.

Testing de Rendimiento

Testing con JMeter



Apache JMeter se puede usar para probar el rendimiento tanto en recursos estáticos como dinámicos, aplicaciones dinámicas web.

Se puede usar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su fuerza o para analizar el rendimiento general bajo diferentes tipos de carga.

Las características de Apache JMeter incluyen:

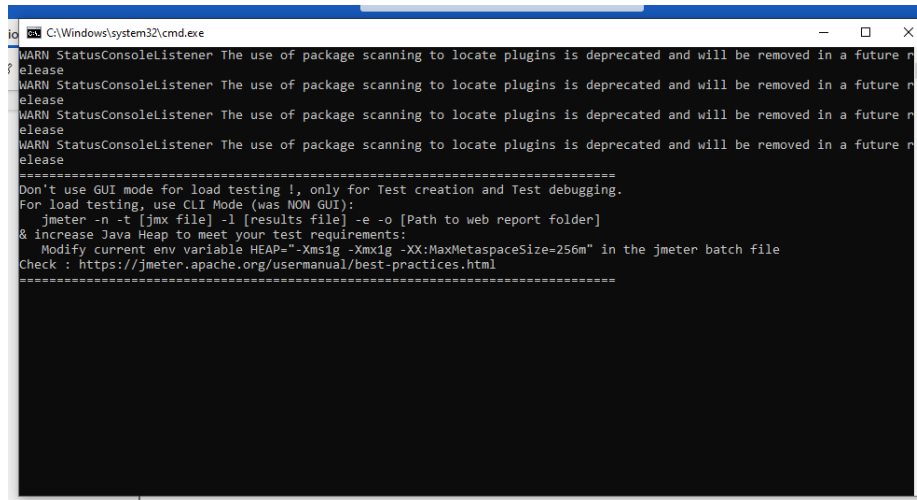
Capacidad para cargar y probar el rendimiento de muchas aplicaciones/servidores/tipos de protocolos diferentes:

- Web: HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...)
- Servicios web SOAP/REST
- FTP
- Base de datos a través de JDBC
- LDAP
- Middleware orientado a mensajes (MOM) a través de JMS
- Correo - SMTP(S), POP3(S) e IMAP(S)
- Comandos nativos o scripts de shell
- TCP

- Objetos Java
- IDE de prueba con todas las funciones que permite la grabación rápida del plan de prueba (desde navegadores o aplicaciones nativas), compilación y depuración .
- Modo CLI (modo de línea de comandos (anteriormente llamado Non GUI) / modo sin cabeza) para cargar la prueba desde cualquier sistema operativo compatible con Java (Linux, Windows, Mac OSX, ...)
- Un informe HTML dinámico completo y listo para presentar
- Fácil correlación a través de la capacidad de extraer datos de los formatos de respuesta más populares, HTML , JSON , XML o cualquier formato de texto
- Portabilidad completa y pureza 100% Java .
- El marco completo de subprocessos múltiples permite el muestreo concurrente por muchos subprocessos y el muestreo simultáneo de diferentes funciones por grupos de subprocessos separados.
- Almacenamiento en caché y análisis/reproducción fuera de línea de los resultados de las pruebas.

JMeter no es un navegador, funciona a nivel de protocolo. En lo que respecta a los servicios web y los servicios remotos, JMeter parece un navegador (o más bien, múltiples navegadores); sin embargo, JMeter no realiza todas las acciones compatibles con los navegadores. En particular, JMeter no ejecuta el Javascript que se encuentra en las páginas HTML. Tampoco representa las páginas HTML como lo hace un navegador (es posible ver la respuesta como HTML, etc., pero los tiempos no se incluyen en ninguna muestra, y solo se muestra una muestra en un hilo a la vez).

Bueno cuando se empieza por abrir el JMeter nos aparece lo siguiente, pero toca esperar unos minutos...



```

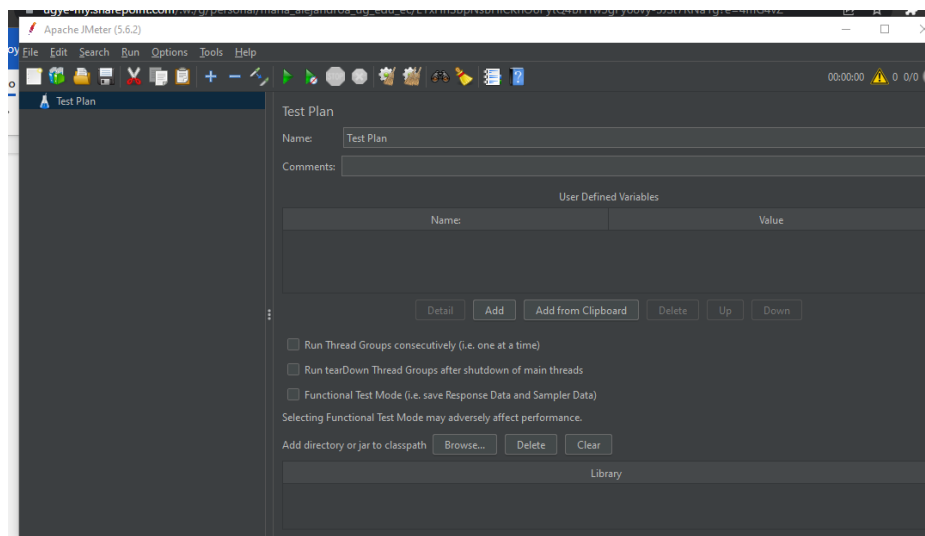
C:\Windows\system32\cmd.exe

WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release

=====
Don't use GUI mode for load testing !, only for Test creation and Test debugging.
For load testing, use CLI Mode (was NON GUI):
jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]
& increase Java Heap to meet your test requirements:
  Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" in the jmeter batch file
Check : https://jmeter.apache.org/usermanual/best-practices.html
=====

```

Una vez cargada se muestra lo siguiente luego de unos minutos...



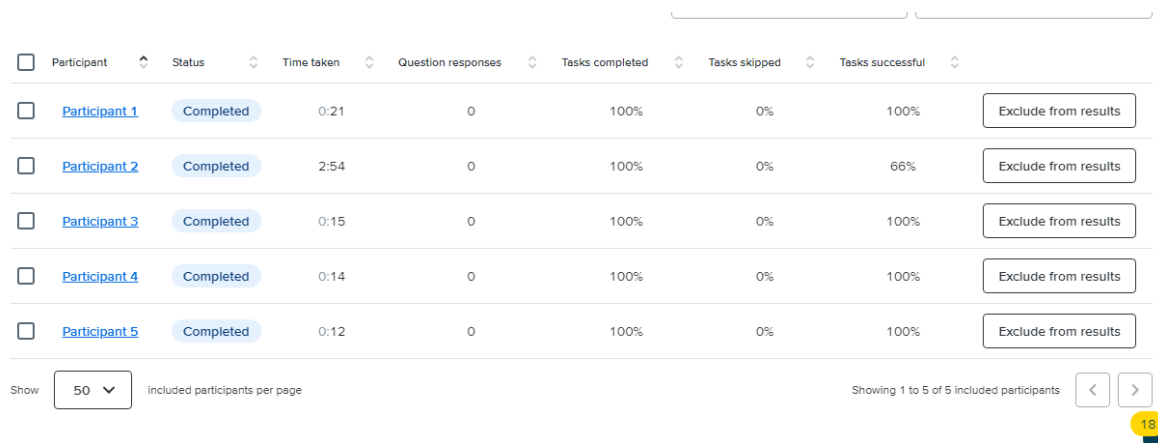
Luego ya se abre el documento que desiamos testear ya tenemos la ejecucion y los resultados.

Testing de Usabilidad

Software para testing de Usabilidad

Para el testing de usabilidad se optó por usar la herramienta Optimal Workshop.

Optimal Workshop es una plataforma de investigación y diseño centrado en la experiencia del usuario que proporciona herramientas y soluciones para ayudar a los equipos de diseño y desarrollo a mejorar la usabilidad y la efectividad de sus productos digitales.



<input type="checkbox"/> Participant	Status	Time taken	Question responses	Tasks completed	Tasks skipped	Tasks successful	
<input type="checkbox"/> Participant 1	Completed	0:21	0	100%	0%	100%	<button>Exclude from results</button>
<input type="checkbox"/> Participant 2	Completed	2:54	0	100%	0%	66%	<button>Exclude from results</button>
<input type="checkbox"/> Participant 3	Completed	0:15	0	100%	0%	100%	<button>Exclude from results</button>
<input type="checkbox"/> Participant 4	Completed	0:14	0	100%	0%	100%	<button>Exclude from results</button>
<input type="checkbox"/> Participant 5	Completed	0:12	0	100%	0%	100%	<button>Exclude from results</button>

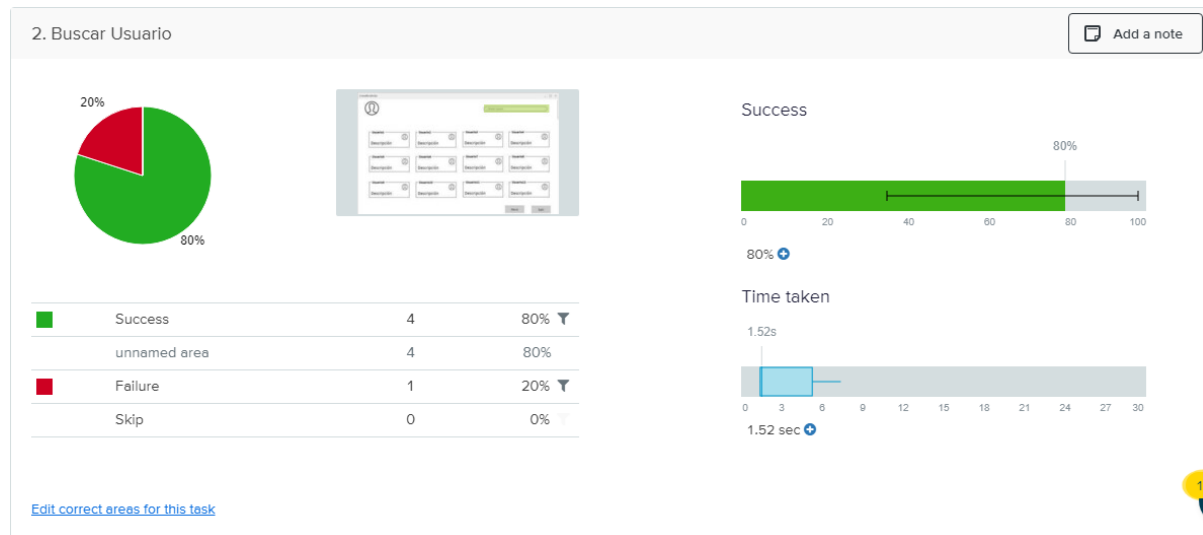
Show included participants per page

Showing 1 to 5 of 5 included participants < >

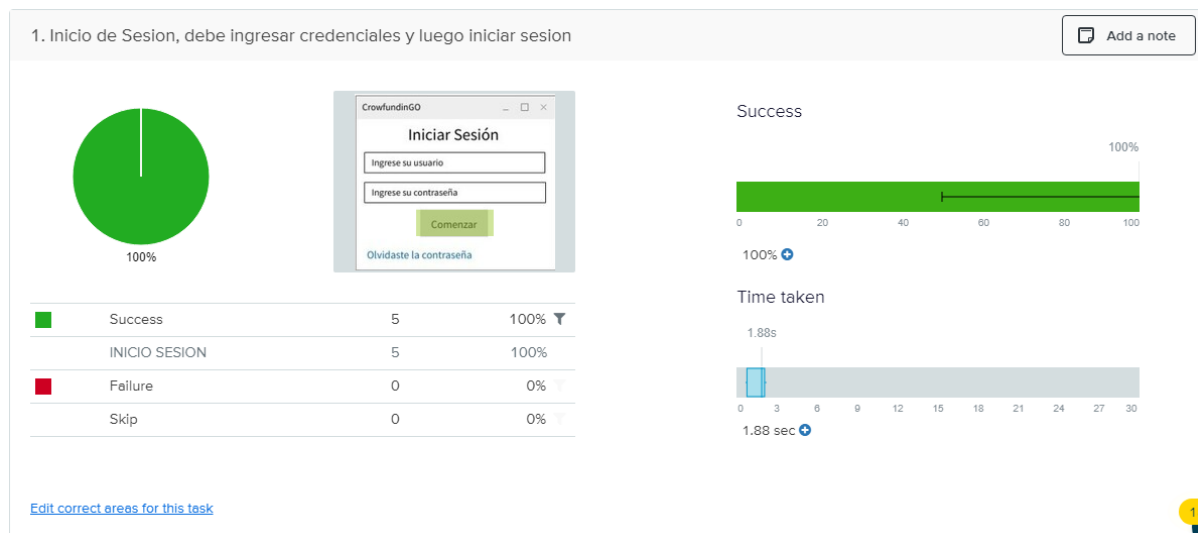
Esta plataforma ofrece una variedad de herramientas y métodos que permiten a los profesionales de UX (experiencia de usuario) y a los diseñadores de interacción llevar a cabo investigaciones, pruebas y análisis de la usabilidad de sus productos.

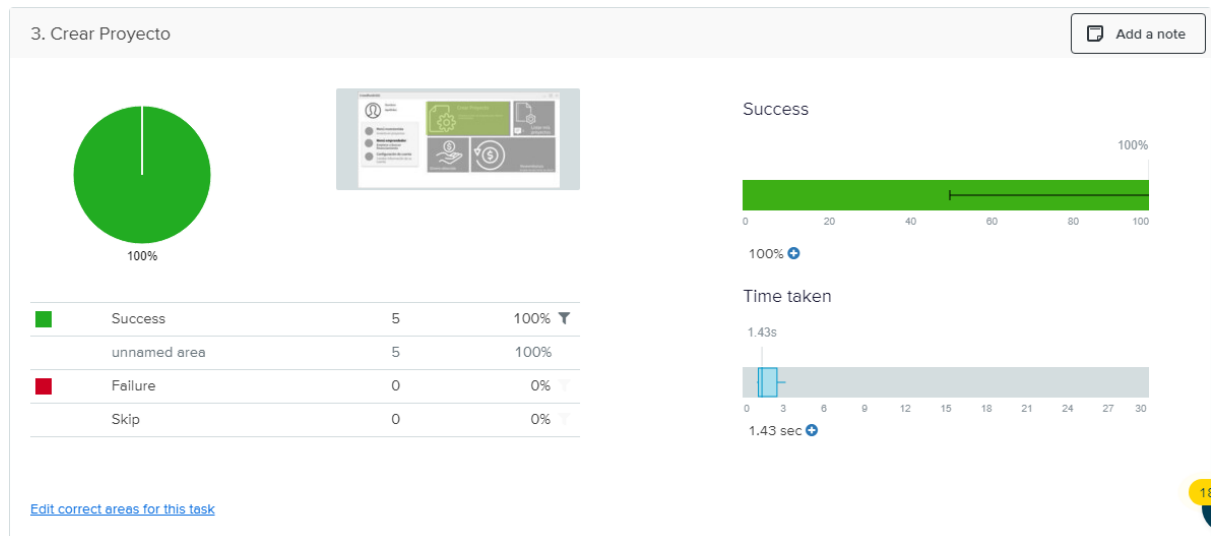
Testing con Optimal Workshop

Para el testing se usaron a 5 participantes que tenían que cumplir con 3 tareas, el inicio sesión, la búsqueda de usuarios y la creación de proyectos:

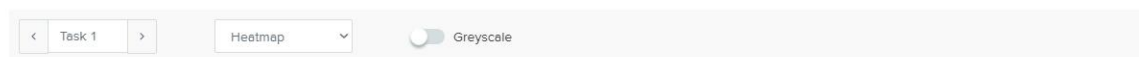


Luego de que se completaron las tareas, se obtuvieron los siguientes resultados:





Mapas de calor donde se indican las zonas con las que el usuario tuvo interacción:



Inicio de Sesión, debe ingresar credenciales y luego iniciar sesión

5 clicks 5 successful 0 skips 1.5 avg (secs)



Buscar Usuario

5 clicks 4 successful 0 skips 3.0 avg (secs)



Crear Proyecto

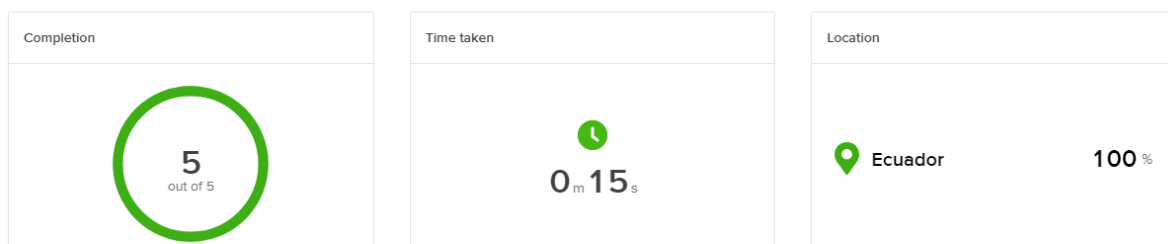
5 clicks 5 successful 0 skips 1.8 avg (secs)



18

Vista General:

Participants



Análisis del testing:

En el contexto de la fase de pruebas y evaluación, es notable destacar que los resultados obtenidos de las pruebas realizadas son sumamente alentadores. De manera específica, al analizar los datos emergentes del proceso de testing en el que participaron los cinco involucrados, se puede discernir que todos los participantes lograron completar las tareas encomendadas en un intervalo de tiempo promedio de 15 segundos, presentándose incluso un impresionante mínimo de 12 segundos en un caso particular.

Conclusiones:

Estos resultados aportan un valioso indicio acerca de la eficiencia y la usabilidad de la interfaz que ha sido desarrollada. Emerge claramente que el sistema en cuestión exhibe un nivel de dinamismo e intuición que facilita la comprensión y la ejecución de las tareas planteadas. Este logro es de particular relevancia, ya que resalta la capacidad de la interfaz para guiar a los usuarios de manera efectiva y eficiente hacia la consecución de sus objetivos dentro del sistema.

Este hallazgo no solo refleja la solidez del diseño y la implementación de la interfaz, sino que también valida el enfoque orientado al usuario que ha sido incorporado en todas las etapas del proceso de desarrollo. La combinación de elementos visuales atractivos con una estructura de navegación fluida y lógica ha resultado en un entorno que no solo se amolda a las expectativas de los usuarios, sino que también supera sus requisitos en términos de accesibilidad y desempeño.

Reporte de Incidencias

Plataforma Confluence

The screenshot shows the Confluence user profile page for M. Belén Alejandro. The top navigation bar includes links for 'Página principal', 'Reciente', 'Espacios', 'Equipos', 'Aplicaciones', 'Plantillas', 'Invitar a personas', and a 'Crear' button. A search bar is also present. The left sidebar shows the user's profile, a 'Resumen' section, and a list of 'Páginas' including 'ProyectoSoftware' and 'Informes de Jira'. The main content area features a welcome message, a 'Páginas recientes en las que he trabajado' section with a list of recent updates, a 'Flujo de blog' section with a 'Crear una entrada de blog' button, and an 'Información de contacto' section. A 'Inicio rápido' button is located in the bottom right corner.

Espacio creado

The screenshot shows the Confluence 'Testing Sala 5' space page. The top navigation bar is similar to the previous page. The left sidebar shows the space's 'Resumen' and a list of 'Páginas' including 'Plantilla - Requisitos del producto', 'Plantilla - Notas', and 'Plantilla - Documentación sobre decisiones'. The main content area features a large orange banner, a green box with a welcome message and a list of suggestions, and a section titled 'Estado' with a 'Escribe una actualización' button. A 'Inicio rápido' button is located in the bottom right corner.

Plataforma Jira

The screenshot shows the Jira Software interface. The top navigation bar includes 'Jira Software', 'Tu trabajo', 'Proyectos', 'Filtros', 'Paneles', 'Equipos', 'Aplicaciones', and a 'Crear' button. A search bar with 'Buscar' is on the right. The left sidebar shows the project 'ProyectoSoftware' and a menu with 'PLANIFICACIÓN' (Cronograma, Backlog, Tablero) and 'DESARROLLO' (Código, Páginas de proyec..., Añadir acceso rápido, Configuración del pro...). The main content area is titled 'Backlog' and shows a list of issues under 'Tablero Sprint 1' (9 ago - 16 ago, 2 incidencias). The issues are: 'PROYEC-1 (Login) El botón de Olvidar contraseña no permite un cambio sin autenticación' and 'PROYEC-2 Sección "Crear cuenta" campos nombre y apellido permite todo tipo de carácter'. Both issues have a 'TAREAS POR HACER' status. Below the sprint board is an empty 'Backlog' section with the message 'Tu backlog está vacío.' and a '+ Crear incidencia' button. A banner at the top of the main area asks: '¿Necesita más de Jira tu equipo? Consigue una versión de prueba gratuita de nuestro plan Standard.'

Tablero Spring en Jira

The screenshot shows the Jira Software interface with the 'Tablero Sprint 1' view selected. The top navigation bar is the same as the previous screenshot. The left sidebar shows the 'Tablero' option selected under 'PLANIFICACIÓN'. The main content area is titled 'Tablero Sprint 1' and shows a Kanban board with three columns: 'POR HACER 2', 'EN CURSO', and 'LISTO ✓'. The 'POR HACER' column contains two issues: 'PROYEC-1 (Login) El botón de Olvidar contraseña no permite un cambio sin autenticación' and 'PROYEC-2 Sección "Crear cuenta" campos nombre y apellido permite todo tipo de carácter'. The 'EN CURSO' and 'LISTO' columns are currently empty. A '+ ' button is visible to the right of the 'LISTO' column. The board shows 'Restantes: 4 días' and a 'Completar sprint' button. A banner at the top of the main area is the same as the previous screenshot. The bottom of the sidebar shows the text 'Estás en un proyecto gestionado por el equipo' and 'Más información'.

Conexión Confluence – Jira

Jira Software Tu trabajo ▾ Proyectos ▾ Filtros ▾ Paneles ▾ Equipos ▾ Aplicaciones ▾ **Crear**

Q Buscar

ProyectoSoftware
Proyecto de software

PLANIFICACIÓN

- Cronograma
- Backlog
- Tablero

DESARROLLO

- Código
- Páginas de proyec...**
- Añadir acceso rápido
- Configuración del pro...

Estás en un proyecto gestionado por el equipo

Más información

Proyectos / ProyectoSoftware

Páginas de proyectos

Testing Sala 5

Mostrar borradores

Plantilla	MA	Creación
Plantilla - Requisitos del producto	MA	Hace 28 minutos
Plantilla - Notas	MA	Hace 28 minutos
Plantilla - Documentación sobre decisiones	MA	Hace 28 minutos
Saca el máximo partido al espacio de tu proyec...	MA	Hace 28 minutos

Con tecnología de Confluence

Crear una página

PLANTILLAS POPULARES

- Página en blanco**
Inicia una página desde cero.
- Decisión de diseño**
Registra las decisiones importantes del proyecto y comunícalas a tu equipo.
- Notas**
Planifica tus reuniones, toma notas y comparte acciones con tu equipo.
- Requisitos del producto**
Define, haz un seguimiento y determina el alcance de los requisitos de tu producto o función.
- Retrospectiva**
¿Qué ha funcionado correctamente? ¿Qué podría haber salido mejor? Colabora de forma abierta con tu equipo para sugerir mejoras.
- Más plantillas**

Informe incidencias Jira

Confluence Página principal Reciente ▾ Espacios ▾ Equipos ▾ Aplicaciones ▾ Plantillas Invitar a personas **Crear**

Q Buscar

M. Belén Alejandro

M. Belén Alejandro

Informes de Jira

Propiedad de M. Belén Alejandro
Hace 10 minutos

Añadir informe de Jira

Título	Fecha	tiqes	estado
ProyectoSoftware	ago 09, 2023		

+ Añadir etiqueta

Sé la primera persona en añadir una reacción

MA Escribe un comentario...

Inicio rápido

Contenido informe incidencias Jira desde Confluence

Confluence

Página principal Reciente Espacios Equipos Aplicaciones Plantillas Invitar a personas Crear

Q Buscar

M. Belén Alejandro

M. Belén Alejandro Enlaces de JIRA de 2

Resumen

Blog

Calendarios

Configuración del espacio

ATAJOS

Informes de Jira

Páginas

ProyectoSoftware

Informes de Jira

ProyectoSoftware

Propiedad de M. Belén Alejandro, creada con una plantilla

Última actualización: hace un momento

Fecha	ago 09, 2023
Tiques	2 issues

Resumen

El presente documento es un resumen de los errores analizados en la implementación del programa en cuestión, se espera una nueva verificación para la entrega final.

✗ Durante la ejecución del programa y las pruebas realizadas, se observó un error en la funcionalidad dentro del módulo de Login, lo que puede implicar un inconveniente a futuro: **PROYEC-1: (Login) El botón de Olvidar contraseña no permite un cambio sin autenticación** [TAREAS POR HACER](#)

✗ Existen errores de análisis y funcionamiento, como la validación de los datos que serán ingresados en el programa: **PROYEC-2: Sección "Crear cuenta" campos nombre y apellido permite todo tipo de carácter** [TAREAS POR HACER](#)

Inicio rápido

Confluence

Página principal Reciente Espacios Equipos Aplicaciones Plantillas Invitar a personas Crear

Q Buscar

M. Belén Alejandro

M. Belén Alejandro Enlaces de JIRA de 2

Resumen

Blog

Calendarios

Configuración del espacio

ATAJOS

Informes de Jira

Páginas

ProyectoSoftware

Informes de Jira

ProyectoSoftware

Propiedad de M. Belén Alejandro, creada con una plantilla

Última actualización: hace un momento

Fecha	ago 09, 2023
Tiques	2 issues

Resumen

El presente documento es un resumen de los errores analizados en la implementación del programa en cuestión, se espera una nueva verificación para la entrega final.

✗ Durante la ejecución del programa y las pruebas realizadas, se observó un error en la funcionalidad dentro del módulo de Login, lo que puede implicar un inconveniente a futuro: **PROYEC-1: (Login) El botón de Olvidar contraseña no permite un cambio sin autenticación** [TAREAS POR HACER](#)

✗ Existen errores de análisis y funcionamiento, como la validación de los datos que serán ingresados en el programa: **PROYEC-2: Sección "Crear cuenta" campos nombre y apellido permite todo tipo de carácter** [TAREAS POR HACER](#)

Inicio rápido

Lo más destacado

Todas las actualizaciones

Error

- PROYEC-2
 [TAREAS POR HACER](#)

Sección "Crear cuenta" campos nombre y apellido permite todo tipo de carácter
- PROYEC-1
 [TAREAS POR HACER](#)

(Login) El botón de Olvidar contraseña no permite un cambio sin autenticación

jirareport + Añadir etiqueta

Inicio rápido