



TEC

Tecnológico de Costa Rica

Simplex Educativo

Plan de Pruebas

Profesora: Maria Estrada.

Curso: Proyecto de Ingeniería de Software.

18.01.2017

Jose Fernando Molina Chacón

Yordan Jiménez Hernández

1. Índice

Índice	2
Introducción	3
Propósito	3
Alcance	3
Estrategia de pruebas	3
Identificación del proyecto	4
Enfoque de las pruebas	4
Pruebas	5
Pruebas de Sistema Móvil	5
Otros tipos de pruebas	8
Pruebas de interfaz gráfica	8
Pruebas de rendimiento	8
Requerimientos de ambiente	9
Hardware	9
Software	9
Encargados	9
Aprobación	9

1. Introducción

1.1. Propósito

El plan de pruebas tiene el propósito de informar a los miembros del equipo o futuras personas que ingresen a desarrollar dentro del proyecto, acerca de las distintas pruebas desarrolladas dentro del programa para validar los casos de uso de la tercera iteración y los componentes involucrados. Es necesario indicar al lector, que el desarrollo de las pruebas fue utilizando el IDE Android Studio, haciendo utilización del framework Espresso de Google para desarrollar todas las pruebas

Dentro de las secciones de este documento, se podrá informar acerca de las características de las pruebas construidas, ya sea como ejecutarlas, resultados esperados y una breve explicación de cómo está constituida. Para la

1.2. Alcance

El plan de pruebas está destinado a comunicar acerca de los elementos de validación y verificación definidos por parte del grupo de trabajo para ratificar la funcionalidad de las características presentes en la solución de software propuesta.

A través del plan de pruebas se podrá aclarar el origen de los errores, defecto o fallas que se generan al ejecutar una prueba gracias a la trazabilidad a los componentes pertenecientes al programa y a los casos de uso.

1.3. Estrategia de pruebas

Para la iteración actual solamente se realizarán pruebas de sistema para probar la funcionalidad completa de la aplicación móvil desarrollado, debido a que el código fuente de las iteraciones 1 y 2 pudo ser reutilizado para la creación de la solución, y dicho código fuente ya había sido probado de manera extensiva durante las iteraciones anteriores.

Las pruebas de sistema que se realizarán harán uso del framework Espresso para realizar testing automatizado de interfaz gráfica e integración de componentes. Dicho framework realiza

automáticamente acciones en la aplicación como si fuera el usuario el que estuviera utilizando la misma, y hace aserciones durante su ejecución para velar por el correcto funcionamiento del sistema.

1.4. Identificación del proyecto

En la tabla siguiente, se especifican los documentos los cuales fueron tomados como apoyo para realizar el plan de pruebas:

Nombre	Hábil	Utilizado
Especificación de requerimientos del software	Sí	Sí
Documento de Visión	Sí	Sí
Diagrama de interacción	Sí	No
Diagrama de componentes	Sí	No
Diagrama de clases	Sí	Sí
Manual de usuario	No	No
Diagrama de despliegue	Sí	No

La enumeración y descripción de los casos de uso se encuentran dentro del documento de especificación de requerimientos del software.

2. Enfoque de las pruebas

En el contenido de esta sección, se describen las pruebas realizados a la aplicación móvil para validar el correcto funcionamiento de la misma. Dentro de las descripciones se detallan características relevantes que el usuario o equipo de mantenimiento debería de saber para ejecutar la pruebas e interpretar los resultados que se obtuvieron.

2.1. Pruebas

2.1.1. Pruebas de Sistema Móvil

Id	PSM-1
Nombre	testProblemaDirecto1
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema factible con el simplex de dos fases de manera directa.
Parámetros	Ninguna. Problema lineal ingresado directamente en el código fuente.
Resultados esperados	<ul style="list-style-type: none">El sistema muestra una ventana de alerta con la solución $z = 45$, $x_1 = 1$, $x_2 = 3$.
Componentes utilizados	PaginaInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 1.

Id	PSM-2
Nombre	testProblemaDirecto2
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema factible con el simplex de dos fases de manera directa y una solución no entera, en formato fraccionario.
Parámetros	Ninguna. Problema lineal ingresado directamente en el código fuente.
Resultados esperados	<ul style="list-style-type: none">El sistema muestra una ventana de alerta con la solución $z = 279$, $x_1 = 7/4$, $x_2 = 7$, $x_3 = 5$.
Componentes utilizados	PaginaInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 1.

Id	PSM-3
Nombre	testProblemaDirecto3
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema factible con el simplex de dos fases de manera directa y una solución no entera, en formato decimal.
Parámetros	Ninguna. Problema lineal ingresado directamente en el código fuente.
Resultados esperados	<ul style="list-style-type: none"> El sistema muestra una ventana de alerta con la solución $z = 279$, $x_1 = 1.75$, $x_2 = 7$, $x_3 = 5$.
Componentes utilizados	PaginalInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 1.

Id	PSM-3
Nombre	testProblemaInfactible
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema infactible.
Parámetros	Ninguna. Problema lineal ingresado directamente en el código fuente.
Resultados esperados	<ul style="list-style-type: none"> El sistema muestra una ventana de alerta con la información sobre la infactibilidad.
Componentes utilizados	PaginalInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 1.

Id	PSM-4
Nombre	testProblemaNoAcotado
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema no acotado.
Parámetros	Ninguna. Problema lineal ingresado directamente en el

	código fuente.
Resultados esperados	<ul style="list-style-type: none"> El sistema muestra una ventana de alerta que informa sobre la no acotabilidad del problema.
Componentes utilizados	PaginalInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 1.

Id	PSM-5
Nombre	testProblemaPasos1
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema factible con el simplex de dos fases por pasos.
Parámetros	Ninguna. Problema lineal ingresado directamente en el código fuente.
Resultados esperados	<ul style="list-style-type: none"> El sistema toma exactamente 4 iteraciones en llegar a la solución. El sistema muestra una ventana de alerta con la solución $z = 45$, $x_1 = 1$, $x_2 = 3$.
Componentes utilizados	PaginalInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 2.

Id	PSM-6
Nombre	testProblemaPasos2
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema factible con el simplex de dos fases por pasos y una solución no entera, en formato fraccionario.
Parámetros	Ninguna. Problema lineal ingresado directamente en el código fuente.
Resultados esperados	<ul style="list-style-type: none"> El sistema toma exactamente 5 iteraciones en llegar a la solución. El sistema muestra una ventana de alerta con la solución $z = 279$, $x_1 = 7/4$, $x_2 = 7$, $x_3 = 5$.

Componentes utilizados	PaginalInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 2.

Id	PSM-7
Nombre	testProblemaPasos3
Descripción	Prueba desde la interfaz gráfica, la ejecución de un problema factible con el simplex de dos fases por pasos y una solución no entera, en formato decimal.
Parámetros	Ninguna. Problema lineal ingresado directamente en el código fuente.
Resultados esperados	<ul style="list-style-type: none"> • El sistema toma exactamente 5 iteraciones en llegar a la solución. • El sistema muestra una ventana de alerta con la solución $z = 279$, $x_1 = 1.75$, $x_2 = 7$, $x_3 = 5$.
Componentes utilizados	PaginalInicial.java, PasoIntermedios.java.
Caso de uso implicados	Caso móvil número 2.

2.2. Otros tipos de pruebas

2.2.1. Pruebas de interfaz gráfica

Dentro de esta iteración no se realizaron pruebas de interfaz gráfica.

2.2.2. Pruebas de rendimiento

Como se estableció en el documento de especificación del software, el rendimiento no se encuentra dentro de los requerimientos del sistema, por ello no se presentan pruebas de rendimiento.

3. Requerimientos de ambiente

3.1. Hardware

Equipo	Características
Equipo cliente	<ul style="list-style-type: none">• Dispositivo Android con versión 6.0 o superior.• Procesador con mínimo un núcleo de 2.0 GHz• Memoria ram con 1 Gb de espacio.

3.2. Software

Equipo	Sistema Operativo	Software requerido para las pruebas
Equipo Cliente	Cualquier sistema operativo que contenga el programa Android Studio.	<ul style="list-style-type: none">- Librerías necesarias:<ul style="list-style-type: none">• JUnit• Espresso.- Android Studio junto con el SDK 7.0 o superior.

4. Encargados

Plan de pruebas	Fernando Molina
Casos de pruebas	Yordan Jiménez
Automatización de pruebas	Yordan Jiménez

5. Aprobación

	Administrador de proyectos	Administrador de calidad
--	----------------------------	--------------------------

Nombre	Fernando Molina	Yordan Jiménez
Firma		