

# Maxima on Android

## Plan de Pruebas

Curso: Ingeniería de Software III

Integrantes: Pérez Rodríguez, Angelo Aldo  
Pfuturi Huisa, Oscar David  
Pucho Zevallos, Kelvin Paul  
Quispe Menor, Hermógenes  
Santos Apaza, Yordy Williams  
Vilcapaza Flores, Luis Felipe

Docente: D.Sc Sarmiento Calisaya, Edgar

Fecha de entrega: 31 de Julio de 2021

Arequipa, Perú

## Registro de cambios

Fecha	Descripción	Versión	Autor
02/08/21	Versión inicial	1.0	Hermogenes Quispe Menor
04/08/21	Sección de Requisitos	1.1	Oscar Pfuturi Huisa / Hermogenes Quispe Menor
05/08/21	Definición de estrategia	1.2	Oscar Pfuturi Huisa / Hermogenes Quispe Menor
09/08/21	Sección de Definición de casos de prueba	1.3	Oscar Pfuturi Huisa
12/08/21	Recursos y definición de pruebas de seguridad y rendimiento	1.4	Oscar Pfuturi Huisa

# Índice de Contenidos

<b>1</b>	<b>Introducción</b>	<b>3</b>
1.1	Objetivos . . . . .	3
1.2	Escopo . . . . .	3
<b>2</b>	<b>Requisitos</b>	<b>3</b>
2.1	Definición de Casos de Prueba . . . . .	4
2.1.1	Verificar si el valor de entrada es un valor entero . . . . .	4
2.1.2	Verificar si el valor de entrada es un signo . . . . .	5
2.1.3	Cálculo de Integrales Definidas y no Definidas . . . . .	6
2.1.4	Verificar la raíz cuadrada . . . . .	7
2.1.5	Verificar valores booleanos . . . . .	8
2.1.6	Comprobar la librería <b>ctensor</b> para ejercicios de cálculo tensorial . . . . .	8
2.1.7	Verificar el valor <i>obase</i> de salida . . . . .	12
2.1.8	Verificar el valor booleano de <i>pformat</i> . . . . .	13
<b>3</b>	<b>Definición de Estrategia</b>	<b>13</b>
3.1	Prueba de funcionamiento . . . . .	13
3.2	Prueba de seguridad . . . . .	14
3.3	Prueba de rendimiento . . . . .	14
<b>4</b>	<b>Recursos</b>	<b>15</b>
4.1	Personal . . . . .	15
4.2	Sistema . . . . .	15
<b>5</b>	<b>Cronograma</b>	<b>16</b>

# 1 Introducción

## 1.1 Objetivos

El objetivo de este documento es entregar las pautas, definir la estrategia que se seguirá y recoger los casos de pruebas que verifican que el sistema satisface los requisitos especificados. Este documento de Plan de Pruebas de Maxima on Android se compone de los siguientes objetivos:

- Identificar los componentes del software que deben ser probados.
- La estrategia a seguir en la ejecución de las pruebas.
- Identificarlos recursos necesarios para la realización de las pruebas
- Listar los elementos resultantes del proyecto de pruebas

## 1.2 Escopo

Maxima en Android pasará las pruebas de unidad y funcionales, de seguridad y de sistema. Las pruebas unitarias se ocuparán de la calidad funcional, de interfaz gráfica y de control de acceso (Seguridad); mientras que las pruebas del sistema abordarán problemas de rendimiento.

# 2 Requisitos

La siguiente lista identifica aquellos elementos (casos de uso, requisitos funcionales y no funcionales) que se han identificado como objetivos de prueba. Esta lista representa lo que se probará.

- Verificar si el valor de entrada es un valor entero.
- Verificar si el valor de entrada es un signo.
- Calculo de integrales.
- Calculo de integrales definidas.
- Verificar la raíz cuadrada de un número.

- Verifica el valor booleano.
- Comprobar la librería `çtensor` para ejercicios de Cálculo Tensorial.
- Verifica el `"pfeformat"` de entrada es `false` o `true`.

## 2.1 Definición de Casos de Prueba

En esta sección se presentan los casos de pruebas generales para usarlos con la herramienta Espresso. Cada cuadro está asociado a un caso de Uso, desde ahí se desglosa en los diferentes módulos involucrados para el funcionamiento y se evalúa el resultado obtenido.

En las siguientes tablas, se muestran los casos de pruebas a realizar en los distintos escenarios:

### 2.1.1 Verificar si el valor de entrada es un valor entero

Caso de Prueba	Verificación de valores enteros	Código	TC001
Código de Escenario	TS001		
Descripción	En el label la entrada <i>askinteger(n)</i> es aquel que verifica si el parámetro es entero o no.		
Pasos	<ol style="list-style-type: none"><li>1. Escribir números u otros caracteres</li><li>2. Presionar la tecla Enter</li></ol>		
Resultado esperado	El valor de entrada debe ser un valor entero para ser aceptado. Para el resto de casos debe devolver un mensaje de error.		
Resultado de la prueba	Aceptada		

### 2.1.2 Verificar si el valor de entrada es un signo

Caso de Prueba	Verificación de valores con signo	Código	TC001
Código de Escenario	TS002		
Descripción	En el label la entrada <i>asksign(n)</i> es aquel que verifica si la variable escrita en su parámetro es un número positivo, negativo o cero.		
Pasos	<ol style="list-style-type: none"><li>1. Escribir valores enteros, flotantes, o caracteres especiales</li><li>2. Presionar la tecla Enter.</li></ol>		
Resultado esperado	El valor de entrada debe ser un valor numérico c/s signo. Para el resto de casos debe devolver un mensaje de error.		
Resultado de la prueba	Aceptada		

### 2.1.3 Cálculo de Integrales Definidas y no Definidas

Caso de Prueba	Repuesta de ejecución del método <i>integrate()</i>	Código	TC001
Código de Escenario	TS003		
Descripción	En el label de entrada el integral verifica sus resultados usando las variables apropiadas		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir el identificador <i>integrate(f(x),x)</i></li> <li>2. En los parámetros de dicho identificador deben ir la función a integrar y la variable independiente.</li> <li>3. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Función integrada. Si existe ambigüedad, es decir, signo de la variable, variable entera, variable decimal, deberá contestar la pregunta que corresponda.		
Resultado de la prueba	Aceptada		

Caso de Prueba	Respuesta a pregunta durante la integración	Código	TC002
Código de Escenario	TS003		
Descripción	Luego de completar el caso anterior, en el label se debe contestar la respuesta del <i>integrate</i>		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir una de las opciones disponibles.</li> <li>2. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Resuelve la integración caso se ingrese una respuesta válida, de lo contrario devuelve la misma pregunta		
Resultado de la prueba	Aceptada		

Caso de Prueba	Respuesta a función exponencial	Código	TC003
Código de Escenario	TS004		
Descripción	Luego de completar el caso anterior, en el label se debe contestar la respuesta del <i>integrate</i>		
Pasos	<ol style="list-style-type: none"><li>1. Escribir una de las opciones disponibles.</li><li>2. Presionar la tecla Enter.</li></ol>		
Resultado esperado	Resuelve la integración caso se ingrese una respuesta válida, de lo contrario devuelve la misma pregunta		
Resultado de la prueba	Aceptada		

#### 2.1.4 Verificar la raíz cuadrada

Caso de Prueba	Verificación de la raíz cuadrada de un número	Código	TC001
Código de Escenario	TS005		
Descripción	El label de entrada de $\sqrt{n}$ va a comprobar si los valores ingresados pueden tener una raíz		
Pasos	<ol style="list-style-type: none"><li>1. Escribir valores enteros, flotantes o caracteres especiales.</li><li>2. Presionar la tecla Enter.</li></ol>		
Resultado esperado	Resolución para un valor entero positivo, de lo contrario, mostrará un mensaje de error		
Resultado de la prueba	Aceptada		



### 2.1.5 Verificar valores booleanos

Caso de Prueba	Verificación de identificadores booleanos	Código	TC001
Código de Escenario	TS006		
Descripción	En el label de entrada <i>exptdispflag</i> : 'booleano' comprueba que opción booleana se ha elegido para retornarla		
Pasos	<ol style="list-style-type: none"><li>1. Escribir <i>true</i> o <i>false</i></li><li>2. Presionar la tecla Enter.</li></ol>		
Resultado esperado	Devuelve el valor booleano ingresado		
Resultado de la prueba	Aceptada		

### 2.1.6 Comprobar la librería **ctensor** para ejercicios de cálculo tensorial

Caso de Prueba	Invocar la librería <b>ctensor</b>	Código	TC001
Código de Escenario	TS007		
Descripción	Para escribir el paquete del <b>ctensor</b> se debe invocar la librería escribiendo el comando <i>load</i>		
Pasos	<ol style="list-style-type: none"><li>1. Escribir <i>load(ctensor)</i></li><li>2. Presionar la tecla Enter.</li></ol>		
Resultado esperado	Devuelve la ruta del paquete		
Resultado de la prueba	Aceptada		

Caso de Prueba	Interactuar con la librería <b>ctensor</b>	Código	TC002
Código de Escenario	TS007		
Descripción	Para empezar a interactuar con la librería se debe escribir el tipo <i>csetup()</i>		
Pasos	<ol style="list-style-type: none"><li>1. Escribir <i>csetup()</i></li><li>2. Presionar la tecla Enter.</li></ol>		
Resultado esperado	Devuelve una pregunta pidiendo las dimensiones de una matriz		
Resultado de la prueba	Aceptada		

Caso de Prueba	Ingresar las dimensiones de la matriz	Código	TC003
Código de Escenario	TS007		
Descripción	Luego del caso anterior se debe escoger la dimensión de la matriz escribiendo el número de dimensiones		
Pasos	<ol style="list-style-type: none"><li>1. Escribir un valor entero</li><li>2. Presionar la tecla Enter.</li></ol>		
Resultado esperado	Confirmación de las dimensiones ingresadas		
Resultado de la prueba	Aceptada		

Caso de Prueba	Confirmar las dimensiones de la matriz	Código	TC004
Código de Escenario	TS007		
Descripción	Afirmación o negación si considera cambiar las coordenadas		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir 'y' o 'n'</li> <li>2. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Muestra una lista de opciones		
Resultado de la prueba	Aceptada		

Caso de Prueba	Elegir una de las opciones	Código	TC005
Código de Escenario	TS007		
Descripción	Las opciones disponibles que se muestran , sugiere que puede escribir una matriz, importar la matriz, o usar métricas con una serie de Taylor. Se debe seleccionar el número de identifica la opción		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir el número de una opción</li> <li>2. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Muestra el resultado correspondiente a la opción elegida		
Resultado de la prueba	Aceptada		

Caso de Prueba	Seleccionar la opción de matriz e ingresar valores	Código	TC006
Código de Escenario	TS007		
Descripción	Rellenar la matriz según la opción escogida y en este caso será rellenar solo la diagonal de la matriz		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir 4 valores numéricos o funciones</li> <li>2. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Petición de ingreso de dependencias o ninguna		
Resultado de la prueba	Aceptada		

Caso de Prueba	Escribir la dependencia	Código	TC007
Código de Escenario	TS007		
Descripción	Escribir la dependencia funcional		
Pasos	<ol style="list-style-type: none"> <li>1. Definir un vector <math>V</math></li> <li>2. Escribir el método <math>depends(V,x)</math></li> <li>3. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Pregunta si se desea ver la matriz		
Resultado de la prueba	Aceptada		

Caso de Prueba	Responder a la pregunta para visualizar la matriz	Código	TC008
Código de Escenario	TS007		
Descripción	Responder a la pregunta (Do you wish to see the metric?) con 'y' para confirmar		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir 'y'</li> <li>2. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Muestra la matriz previamente creada		
Resultado de la prueba	Aceptada		

### 2.1.7 Verificar el valor *obase* de salida

Caso de Prueba	Utilizar el método <i>obase: n</i>	Código	TC001
Código de Escenario	TS008		
Descripción	En el label de entrada se debe escribir <i>obase: n</i> , y luego verificar la salida		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir valores numéricos</li> <li>2. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	Muestra la base del número ingresado		
Resultado de la prueba	Aceptada		

### 2.1.8 Verificar el valor booleano de *pfefor*mat

Caso de Prueba	Comprobar la salida de <i>pfefor</i> mat	Código	TC001
Código de Escenario	TS009		
Descripción	En el label de entrada comprobamos las salidas del <i>pfefor</i> mat con los true o false		
Pasos	<ol style="list-style-type: none"> <li>1. Escribir valores booleanos</li> <li>2. Presionar la tecla Enter.</li> </ol>		
Resultado esperado	...		
Resultado de la prueba	Aceptada		

## 3 Definición de Estrategia

### 3.1 Prueba de funcionamiento

Propósito de la prueba:	Garantice la funcionalidad adecuada del objetivo de prueba, incluida la navegación, la entrada, el procesamiento y la recuperación de datos.
Técnica:	<p>Ejecute cada caso de uso, flujo de caso de uso, utilizando datos válidos y no válidos, para verificar lo siguiente:</p> <ul style="list-style-type: none"> <li>- Los resultados esperados ocurren cuando se utilizan datos válidos</li> <li>- Se muestran los mensajes de error o advertencia correspondientes cuando se utilizan datos no válidos.</li> </ul>
Criterios de finalización:	<p>Se han realizado todas las pruebas planificadas.</p> <p>Se han abordado todos los defectos identificados.</p>
Consideraciones Especiales:	Ninguno
Herramienta:	Espresso

### 3.2 Prueba de seguridad

Propósito de la prueba:	Seguridad a nivel de aplicación. Verificar que el usuario pueda acceder a las funciones o datos software a las cuales tenga permitido Seguridad a nivel de sistema. Verificar que el usuario pueda acceder al sistema cuenta con el permiso.
Técnica:	Identificar las funciones y datos a las que el usuario tenga permitido acceder. Realizar pruebas de acceso a las funciones.
Criterios de finalización:	Las funciones y datos están disponibles para el usuario. Se han realizado todas las pruebas planificadas. Se han abordado todos los defectos identificados.
Consideraciones Especiales:	Ninguno
Herramienta:	Mobile Security Framework (MobSF)

### 3.3 Prueba de rendimiento

Propósito de la prueba:	Determinar la respuesta del sistema ante cierta carga, como validar otros atributos relacionados a la calidad (escalabilidad, fiabilidad, uso de recursos).
Técnica:	Comprobar el comportamiento del sistema ante una determinada carga: - Usar múltiples métodos para la resolución de una función - Definir conjuntos de grandes cantidades de datos, p.e., vectores, matrices.
Criterios de finalización:	Se han realizado todas las pruebas sin ninguna falla y dentro de un tiempo óptimo Se han abordado todos los defectos identificados.
Consideraciones Especiales:	Ninguno
Herramienta:	Apache JMeter

## 4 Recursos

### 4.1 Personal

Recursos Humanos		
Nombres y Apellidos	Rol(es)	Responsabilidades Específicas
Angelo Aldo Pérez Rodríguez	Analista, diseñador y programador	Evaluar las condiciones de término para el proceso de pruebas. Implementar las pruebas
Oscar David Pfuturi Huisa	Analista, diseñador y programador	Identificar los casos de prueba. Generación del plan de pruebas
Kelvin Paul Pucho Zevallos	Analista, diseñador y programador	Identificar, priorizar, e implementar los casos de prueba
Hermogenes Quispe Menor	Analista, diseñador y programador	Identificar los casos de prueba. Generación del plan de pruebas
Yordy Williams Santos Apaza	Analista, diseñador y programador	Evaluar las condiciones de término para el proceso de pruebas. Implementar las pruebas
Luis Felipe Vilcapaza Flores	Analista, diseñador y programador	Evaluar las condiciones de término para el proceso de pruebas. Ejecutar las pruebas

### 4.2 Sistema

Para el proceso de pruebas del proyecto se requiere de la disponibilidad de los siguientes entornos:

Recursos del Sistema	
Dispositivo:	Dispositivo móvil con sistema operativo Android
Servicio de conexión a internet:	Acceso a la aplicación App Store para la descarga del software



## 5 Cronograma

Tarea	Fecha de inicio	Fecha de término
Planear pruebas	24/07/21	25/07/21
Diseñar pruebas	26/07/21	27/07/21
Implementar pruebas	28/07/21	06/08/21
Ejecutar pruebas	07/08/21	09/08/21
Evaluar pruebas	10/08/21	12/08/21