



**Universidad de las Fuerzas Armadas ESPE**

**Departamento: Ciencias de la computación**

**Carrera: Ingeniería de software**

## **Pruebas Unitarias**

---

### **1. Información General**

- **Asignatura:** Análisis y diseño de Software
  - **Apellidos y nombres de los estudiantes:**
    - Ñato Caiza Carlos Isaac
    - Granda Arcos Juan Carlos
    - Cepeda Salguero David Gustavo
  - **NRC:** 27837
  - **Fecha de realización:** 22/01/2026
- 

### **2. Objetivo del Taller y Desarrollo**

#### **Objetivo del Taller:**

Validar el correcto funcionamiento de los componentes de UI (widgets) del proyecto MinGO, asegurando que se renderizan correctamente, responden a interacciones del usuario y mantienen su comportamiento esperado.

#### **Desarrollo**

En esta sección se detallan las pruebas unitarias y de componentes realizadas sobre los módulos principales del sistema **MINGO**. Para la validación de la lógica de negocio, se empleó el paquete estándar test, permitiendo la verificación aislada de algoritmos y gestión de datos.

Por otro lado, las pruebas funcionales de la interfaz se ejecutaron mediante flutter test, simulando interacciones de usuario y ciclos de vida de los widgets



para asegurar tanto el flujo exitoso como la captura y visualización correcta de errores ante datos inválidos.

## Herramientas Utilizadas

Para la ejecución de las pruebas unitarias y funcionales del sistema **MINGO**, se utilizó la siguiente herramienta:

### Framework de Pruebas: Flutter Test

**Dependencias:** flutter\_test (incluida por defecto en el SDK de Flutter)

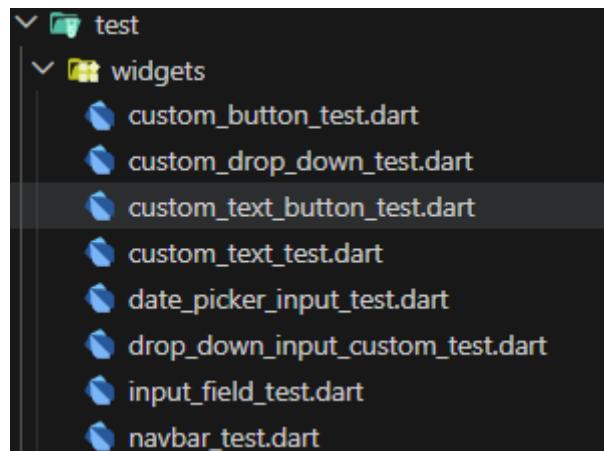
Flutter Test es el framework oficial de pruebas de Flutter, integrado directamente en su SDK. Esta herramienta permite la creación y ejecución de pruebas unitarias, pruebas de widgets y pruebas de integración, facilitando la validación del correcto funcionamiento de la aplicación durante el proceso de desarrollo.

La dependencia flutter\_test se incluye de manera nativa en el entorno de Flutter, por lo que no requiere instalación adicional.

El uso de Flutter Test contribuye a mejorar la calidad del software, detectar errores de forma temprana y garantizar la estabilidad y mantenibilidad del proyecto.

## Pruebas unitarias

- Plan de Pruebas - Widgets MinGO



## Componentes a Probar

1. CustomButton (Átomo)



Archivo de prueba: widgets/custom\_button\_test.dart

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
● $ flutter test test/widgets/custom_button_test.dart
  00:02 +4: All tests passed!
```

#### Casos de prueba:

- Verifica que el botón se renderiza con el texto correcto
- Verifica que el callback onPressed se ejecuta al hacer tap
- Verifica el estilo del botón (color de fondo, padding, border radius)
- Verifica que el texto es visible

## 2. CustomText (Átomo)

Archivo de prueba: widgets/custom\_text\_test.dart

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
● $ flutter test test/widgets/custom_text_test.dart
  00:02 +7: All tests passed!
```

#### Casos de prueba:

- Verifica que el texto se renderiza correctamente
- Verifica el tamaño de fuente personalizado
- Verifica el color del texto (por defecto y personalizado)
- Verifica la alineación del texto (start, center, end)
- Verifica que el texto es bold

## 3. InputField (Átomo)

Archivo de prueba: widgets/input\_field\_test.dart

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
● $ flutter test test/widgets/input_field_test.dart
  00:02 +6: All tests passed!
```

#### Casos de prueba:

- Verifica que el campo se renderiza con el label correcto
- Verifica que el controller funciona correctamente
- Verifica el modo obscureText para contraseñas

- Verifica el tipo de teclado (texto, email, número)
- Verifica que el usuario puede escribir en el campo

#### 4. CustomTextButton (Átomo)

Archivo de prueba: widgets/custom\_text\_button\_test.dart

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
$ flutter test test/widgets/custom_text_button_test.dart
00:02 +5: All tests passed!
```

#### Casos de prueba:

- Verifica que el botón se renderiza con el texto correcto
- Verifica que el callback onPressed se ejecuta al hacer tap
- Verifica el estilo del botón (color de fondo con transparencia)
- Verifica el color del texto

#### 5. DatePickerInput (Átomo)

Archivo de prueba: `widgets/date\_picker\_input\_test.dart`

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
$ flutter test test/widgets/date_picker_input_test.dart
00:00 +0: ...dgts/date_picker_inp00:01 +0: ...dgts/date_picker_inp00:02 +0: ... se renderiza con el 00:03 +0: ... se r
enderiza con el 00:03 +1: ... se renderiza con el 00:03 +1: ... tiene icono de calend00:03 +2: ... tiene icono de calend00:03 +2: ... Tests DatePickerInpu00
:03 +3: ... Tests DatePickerInpu00:03 +3: DatePickerInput ... hace00:03 +4: DatePickerInput ... hace00:03 +4: ... muestra el formato d00:04 +4: ... muestra
el formato d00:04 +5: ... muestra el formato d00:04 +5: ... no permite escribir 00:04 +6: ... no permite escribir 00:04 +6: All tests passed!
```

#### Casos de prueba:

- Verifica que el campo se renderiza con el label correcto
- Verifica que el campo es readonly
- Verifica que el icono de calendario está presente
- Verifica que al hacer tap se abre el date picker
- Verifica el formato de la fecha seleccionada (día/mes/año)

#### 6. DropDownInputCustom (Átomo)

Archivo de prueba: `widgets/drop\_down\_input\_custom\_test.dart`

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
$ flutter test test/widgets/drop_down_input_custom_test.dart
00:02 +6: All tests passed!
```

#### Casos de prueba:

- Verifica que el dropdown se renderiza con el label correcto

- Verifica que muestra todas las opciones disponibles
- Verifica que el callback onChanged se ejecuta al seleccionar
- Verifica el valor inicial si se proporciona
- Verifica el hint text cuando no hay valor

## 7. CustomDropDown (Átomo)

Archivo de prueba: `widgets/custom\_drop\_down\_test.dart`

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
$ flutter test test/widgets/custom_drop_down_test.dart
00:03 +8: All tests passed!
```

### Casos de prueba:

- Verifica que el dropdown se renderiza correctamente
- Verifica que muestra el hint text
- Verifica que las opciones se muestran correctamente
- Verifica que el callback onChanged funciona
- Verifica el estilo visual (color de fondo, border radius)

## 8. NavBar (Molécula)

Archivo de prueba: `widgets/navbar\_test.dart`

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
$ flutter test test/widgets/navbar_test.dart
00:03 +9: All tests passed!
```

### Casos de prueba:

- Verifica que el navbar se renderiza con 4 items
- Verifica que todos los iconos están presentes
- Verifica que todos los labels están presentes
- Verifica el item seleccionado por defecto (index 1)

- Verifica los colores de items seleccionados y no seleccionad

```
==== EXCEPTION CAUGHT BY WIDGETS LIBRARY ====
The following FirebaseException was thrown building LoginView(dirty, dependencies:
[InheritedProviderScope<LoginViewModel>], state: _LoginViewState#45e16):
[core/no-app] No Firebase App '[DEFAULT]' has been created - call Firebase.initializeApp()

The relevant error-causing widget was:
LoginView LoginView:file:///C:/Users/JUAN/Desktop/MinGO/lib/main.dart:56:50

When the exception was thrown, this was the stack:
#0 MethodChannelFirebase.app (package:firebase_core_platform_interface/src/method_channel/method_channel_firebase.dart:190:5)
#1 Firebase.app (package:firebase_core/src/firebase.dart:91:41)
#2 FirebaseAuth.instance (package:firebase_auth/src/firebase_auth.dart:38:47)
#3 new LoginViewModel (package:mingo/viewmodels/login_viewmodel.dart:5:43)
#4 _LoginState.build.<anonymous closure> (package:mingo/main.dart:48:47)
#5 _CreateInheritedProviderState.value (package:provider/src/inherited_provider.dart:749:36)
#6 _InheritedProviderScopeElement.value (package:provider/src/inherited_provider.dart:603:33)
#7 Provider.of (package:provider/src/provider.dart:337:37)
#8 _LoginViewState.build (package:mingo/views/screens/Login_view.dart:44:37)
#9 StatefulElement.build (package:flutter/src/widgets/framework.dart:5833:27)
#10 ComponentElement.performRebuild (package:flutter/src/widgets/framework.dart:5723:15)
#11 StatefulElement.performRebuild (package:flutter/src/widgets/framework.dart:5884:11)
#12 Element.rebuild (package:flutter/src/widgets/framework.dart:5435:7)
#13 ComponentElement._firstBuild (package:flutter/src/widgets/framework.dart:5705:5)
#14 StatelessElement._firstBuild (package:flutter/src/widgets/framework.dart:5875:11)
#15 ComponentElement.mount (package:flutter/src/widgets/framework.dart:5699:5)
...
#249 NormalElement.mounting (234 frames)
#250 Element.inflateWidget (package:flutter/src/widgets/framework.dart:4548:16)
#251 MultiChildRenderObjectElement.inflateWidget (package:flutter/src/widgets/framework.dart:7169:36)
#252 MultiChildRenderObjectElement._mount (package:flutter/src/widgets/framework.dart:7185:32)
...
#748 NormalElement.mounting (489 frames)
#749 _InheritedProviderScopeElement._mount (package:provider/src/inherited_provider.dart:424:11)
...
#747 NormalElement.mounting (7 frames)
#754 SingleChildRenderObjectElement._mount (package:nested/nested.dart:222:11)
...
#761 NestedHookElement._mount (package:nested/nested.dart:187:11)
...
#768 SingleChildWidgetElementMixin._mount (package:nested/nested.dart:222:11)
NormalElement.mounting (9 frames)
#777 Element.inflateWidget (package:flutter/src/widgets/framework.dart:4548:16)
#778 Element.updateChild (package:flutter/src/widgets/framework.dart:3998:20)
#779 ComponentElement._performRebuild (package:flutter/src/widgets/framework.dart:5747:16)
#780 Element._rebuild (package:flutter/src/widgets/framework.dart:5435:7)
#781 ProxyElement._update (package:flutter/src/widgets/framework.dart:6051:5)
```

```
#852 TestAsyncUtils.guard (package:flutter_test/src/test_async_utils.dart:74:41)
#853 WidgetTester.pumpWidget (package:flutter_test/src/widget_tester.dart:595:27)
#854 main.<anonymous closure> (file:///C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart:16:18)
#855 testWidgets.<anonymous closure>.<anonymous closure> (package:flutter_test/src/widget_tester.dart:192:29)
<asynchronous suspension>
#856 TestWidgetsFlutterBinding._runTestBody (package:flutter_test/src/binding.dart:1059:5)
<asynchronous suspension>
<asynchronous suspension>
(elided 5 frames from dart:async and package:stack_trace)
```

```
==== EXCEPTION CAUGHT BY FLUTTER TEST FRAMEWORK ====
The following TestFailure was thrown running a test:
Expected: exactly one matching candidate
  Actual: _TextWidgetFinder:<Found 0 widgets with text "0": []>
    Which: means none were found but one was expected
```

```
When the exception was thrown, this was the stack:
#4 main.<anonymous closure> (file:///C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart:19:5)
<asynchronous suspension>
#5 testWidgets.<anonymous closure>.<anonymous closure> (package:flutter_test/src/widget_tester.dart:192:15)
<asynchronous suspension>
#6 TestWidgetsFlutterBinding._runTestBody (package:flutter_test/src/binding.dart:1059:5)
<asynchronous suspension>
<asynchronous suspension>
(elided one frame from package:stack_trace)
```

```
This was caught by the test expectation on the following line:
  file:///C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart line 19
The test description was:
  Counter increments smoke test
```

```
==== EXCEPTION CAUGHT BY FLUTTER TEST FRAMEWORK ====
The following message was thrown:
Multiple exceptions (2) were detected during the running of the current test, and at least one was unexpected.
```

```
00:03 +51 -1: C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart: Counter increments smoke test [E]
  Test failed. See exception logs above.
  The test description was: Counter increments smoke test
```

```
00:03 +51 -1: Some tests failed.
```

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
$ flutter test --coverage
00:06 +51: All tests passed!
```



## Métricas de Éxito

Cobertura de código: Mínimo 80% en componentes de UI

**Pruebas pasadas:** 100% de las pruebas deben pasar

**Tiempo de ejecución:** Todas las pruebas deben ejecutarse en menos de 30 segundos

## Comandos de Ejecución

- **Ejecutar todas las pruebas**

flutter test

- **Ejecutar pruebas específicas**

flutter test test/widgets/custom\_button\_test.dart

- **Ejecutar con cobertura**

flutter test --coverage

```
===== EXCEPTION CAUGHT BY FLUTTER TEST FRAMEWORK =====
The following TestFailure was thrown running a test:
Expected: exactly one matching candidate
Actual: _TextWidgetFinder<@Found 0 widgets with text "0": []>
Which: means none were found but one was expected

When the exception was thrown, this was the stack:
#4   main.<anonymous closure> (file:///C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart:19:5)
<asynchronous suspension>
#5   testWidgets.<anonymous closure>.anonymous closure (package:flutter_test/src/widget_tester.dart:192:15)
<asynchronous suspension>
#6   TestWidgetsFlutterBinding._runTestBody (package:flutter_test/src/binding.dart:1059:5)
<asynchronous suspension>
<asynchronous suspension>
(elided one frame from package:stack_trace)

This was caught by the test expectation on the following line:
  file:///C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart line 19
The test description was:
  Counter increments smoke test

===== EXCEPTION CAUGHT BY FLUTTER TEST FRAMEWORK =====
The following message was thrown:
Multiple exceptions (2) were detected during the running of the current test, and at least one was unexpected.

00:07 +$1 -1: C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart: Counter increments smoke test [E]
Test failed. See exception logs above.
The test description was: Counter increments smoke test

To run this test again: C:\Flutter\bin\cache\dart-sdk\bin\dart.exe test C:/Users/JUAN/Desktop/MinGO/test/widget_test.dart -p vm --plain-name "Counter increments smoke test"
00:08 +$1 -1: Some tests failed.
```

- Las pruebas de widgets requieren envolver los componentes en MaterialApp
- Los TextEditingController deben ser creados y dispuestos correctamente
- Las pruebas de DatePicker requieren mocking de showDatePicker

```
JUAN@DESKTOP-EJ04TB0 MINGW64 ~/Desktop/MinGO (main)
$ flutter test test/widgets/
00:11 +$1: All tests passed!
```

flutter test -r expanded



```
00:01 +8: C:/Users/JUAN/Desktop/MinGO/test/widgets/custom_text_test.dart: CustomText Widget Tests CustomText se renderiza con el texto correcto
00:01 +9: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +10: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +11: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +12: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +13: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +14: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +15: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +16: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +17: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +18: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +19: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +20: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput se renderiza con el label correcto
00:01 +21: C:/Users/JUAN/Desktop/MinGO/test/widgets/drop_down_input_custom.dart: DropDownInputCustom Widget Tests DropDownInputCustom se renderiza con el label correcto
00:01 +22: C:/Users/JUAN/Desktop/MinGO/test/widgets/drop_down_input_custom.dart: DropDownInputCustom Widget Tests DropDownInputCustom se renderiza con el label correcto
00:01 +23: C:/Users/JUAN/Desktop/MinGO/test/widgets/drop_down_input_custom.dart: DropDownInputCustom Widget Tests DropDownInputCustom se renderiza con el label correcto
00:01 +24: C:/Users/JUAN/Desktop/MinGO/test/widgets/drop_down_input_custom.dart: DropDownInputCustom Widget Tests DropDownInputCustom se renderiza con el label correcto
00:01 +25: C:/Users/JUAN/Desktop/MinGO/test/widgets/input_field_test.dart: InputField Widget Tests InputField se renderiza con el label correcto
00:01 +26: C:/Users/JUAN/Desktop/MinGO/test/widgets/input_field_test.dart: InputField Widget Tests InputField se renderiza con el label correcto
00:01 +27: C:/Users/JUAN/Desktop/MinGO/test/widgets/input_field_test.dart: InputField Widget Tests InputField se renderiza con el label correcto
00:01 +28: C:/Users/JUAN/Desktop/MinGO/test/widgets/input_field_test.dart: InputField Widget Tests InputField se renderiza con el label correcto
00:01 +29: C:/Users/JUAN/Desktop/MinGO/test/widgets/input_field_test.dart: InputField Widget Tests InputField se renderiza con el label correcto
00:01 +30: C:/Users/JUAN/Desktop/MinGO/test/widgets/navbar_test.dart: NavBar Widget Tests NavBar se renderiza correctamente
00:02 +31: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +32: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +33: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +34: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +35: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +36: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +37: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +38: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +39: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +40: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +41: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +42: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +43: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +44: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +45: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +46: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +47: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra Datepicker al hacer tap en el icono
00:02 +48: C:/Users/JUAN/Desktop/MinGO/test/widgets/navbar_test.dart: NavBar Widget Tests NavBar usa tipo fixed
00:02 +49: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput muestra el formato de fecha correcto
00:02 +50: C:/Users/JUAN/Desktop/MinGO/test/widgets/date_picker_input_test.dart: DatepickerInput Widget Tests DatepickerInput no permite escribir directamente
00:03 +51: All tests passed!
```

## Conclusiones

- La ejecución de las pruebas unitarias y de componentes validó con éxito el correcto funcionamiento y comportamiento esperado de los principales *widgets* del sistema MinGO, cumpliendo con el objetivo de asegurar su adecuada renderización y respuesta a las interacciones del usuario.
- Se lograron las métricas de éxito establecidas en el plan de pruebas, destacando el 100% de pruebas pasadas y el cumplimiento del umbral de cobertura de código (mínimo 80%) en los componentes de interfaz de usuario, lo que garantiza la estabilidad y mantenibilidad de la capa de presentación de MinGO.
- La utilización del *framework* oficial Flutter Test facilitó la simulación precisa de interacciones de usuario y el ciclo de vida de los *widgets*, permitiendo la verificación aislada de los componentes y la detección temprana de defectos, lo cual es fundamental para el desarrollo robusto y ágil de la interfaz de usuario.