



# Testing de emulador de microprocesaro Leon3 para desarrollo de software satelital y simuladores

Especificación de requerimientos de software

Autor:

Ing. Iriarte Fernandez, Nicolás Ezequiel (NicolasIriarte95@gmail.com)

Director:

Esp. Lic. Horro, Nicolás Eduardo (INVAP.S.E.)

Docentes:

Alejandro Permingeat; Esteban Volentini; Mariano Finochietto y Rafael Oliva.

*Este documento fue realizado en el curso Testing de Software en Sistemas Embebidos  
el 10 de Marzo de 2024, cuarto bimestre.*

## Índice

<b>Introducción</b> . . . . .	<b>4</b>
1.1 Contenidos . . . . .	4
<b>Asignaciones</b> . . . . .	<b>4</b>
2.1 Responsable . . . . .	4
2.2 Alcance . . . . .	4
2.3 Objetivos . . . . .	4
2.4 Precondiciones . . . . .	5
<b>Bases del test</b> . . . . .	<b>5</b>
<b>Estrategia general del test</b> . . . . .	<b>5</b>
4.1 Características de calidad . . . . .	5
4.2 Características de calidad de niveles de prueba . . . . .	6
<b>Estrategia por nivel de prueba</b> . . . . .	<b>7</b>
5.1 División del sistema en subsistemas . . . . .	7
5.2 Importancia relativa de los subsistemas . . . . .	7
5.3 Determinación de la importancia de test por combinaciones de subsistema y características de calidad . . . . .	7

## Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento.	10 de Marzo de 2024
1	Se aplican cambios sugeridos por Salamandri Santiago.	20 de noviembre de 2023

## Documentos anexos

Ref.	Nombre	Descripción
AD.01	NEMU-SRD-1	Especificación de requerimientos de software.
AD.02	NEMU-UCD-0	Defenición de casos de uso y arquitectura de software.

Cuadro 1. Documentos anexos.

## Introducción

En el presente documento se detallarán todos los aspectos relacionados con la especificación del Master Test Plan referentes al desarrollo del software del “Emulador de microprocesador Leon3 para desarrollo de software satelital y simuladores” referenciado en la entrada **AD.01** en la Tabla 1.

### 1.1. Contenidos

Los contenidos del presente Master Test Plan son:

- Asignaciones
- Bases del test
- Estrategia general del test
- Estrategia por nivel de prueba

## Asignaciones

### 2.1. Responsable

El responsable de la elaboración de este documento es Nicolás Iriarte, ingeniero a cargo del desarrollo del proyecto.

### 2.2. Alcance

El alcance del test de aceptación se restringirá exclusivamente a la ejecución precisa de instrucciones por parte del emulador, excluyendo la evaluación del funcionamiento de los periféricos que puedan estar emulados.

### 2.3. Objetivos

Los objetivos son:

- Determinar si el sistema cumple con los requerimientos.
- Reportar las diferencias entre el comportamiento esperado del sistema y el observado.
- Generar un ambiente de pruebas automáticas que permita verificar el correcto funcionamiento del emulador a medida que su desarrollo avanza.

## 2.4. Precondiciones

Las precondiciones extenas son:

- Se dispondrá en todo momento del emulador en su fase de desarrollo.
- Se tendrá acceso a un servidor GitLab para pruebas automáticas.
- Se tendrá comunicación con expertos y se podrán realizar pruebas contra un modelo de referencia para clarificar dudas y chequear resultados.
- El emulador a ser testeado dispondrá de una API bien documentada para su comunicación.

## Bases del test

Las bases del test consisten en las siguientes especificaciones:

- Planificación del producto y requerimientos.
- Especificación de requerimientos de software.
- Documento de especificación de arquitectura.

## Estrategia general del test

### 4.1. Características de calidad

Las caracterpísticas de calidad para realizar el test:

Característica de calidad	Importancia relativa (%)
Funcionalidad	50
Confiabilidad	25
Cambiabilidad	15
Usabilidad	10
<b>TOTAL</b>	<b>100</b>

- Funcionalidad: El emulador del procesador debe ser capaz de ejecutar las instrucciones del procesador de manera precisa y sin errores, ya que esto es esencial para garantizar la fiabilidad de los resultados de la emulación.
- Confiabilidad: Debe ser capaz de mantener un rendimiento estable y confiable durante largos períodos de procesamiento continuo, incluso durante horas de ejecución ininterrumpida. El emulador debe ser capaz de gestionar eficientemente los recursos y prevenir cualquier degradación del rendimiento que pueda surgir debido a la carga prolongada, garantizando así una emulación consistente y precisa del procesador.
- Cambiabilidad: El software debe poder ser fácilmente actualizable y adaptable. La razón de esto es que se trata de un software en desarrollo activo, donde nuevas versiones serán emitidas constantemente.

- Usabilidad: Se debe tener acceso rápido y sencillo a información relevante sobre el estado del emulador. Se debe poder visualizar de manera clara y accesible los parámetros relacionados con la emulación del procesador, como el estado de sus registros y cualquier mensaje de error o advertencia que pueda surgir durante la operación. Esto facilitará la monitorización y el diagnóstico de posibles problemas, garantizando un uso eficiente y efectivo del emulador.

#### 4.2. Características de calidad de niveles de prueba

-	Funcionalidad	Confiabilidad	Cambiabilidad	Usabilidad
<b>Importancia relativa</b>	50 %	25 %	15 %	10 %
<b>Unit test</b>	++	+		+
<b>Software integration test</b>	+	++	+	+
<b>Hardware / Software integration test</b>				
<b>Acceptance Tests</b>	++	++	++	++
<b>Field test</b>	+	+	+	+

Referencias:

- ++: La característica de calidad es predominante para el nivel de test.
- +: La característica de calidad es relevante para el nivel de test.
- Vacío: La característica de calidad no es relevante para el nivel de test.

Se indica a continuación el motivo de la selección de cada tipo de test para cada tipo de característica de calidad.

- Funcionalidad: Esta característica se evalúa en todos los niveles de test, ya que es esencial para garantizar el correcto funcionamiento del emulador. Siendo los tests unitarios y de aceptación los más relevantes, ya que el primero permite evaluar el comportamiento de cada componente del emulador de manera aislada, y el segundo permite evaluar el comportamiento del emulador en su conjunto.
- Confiabilidad: Esta característica se evalúa en todos los niveles de test, ya que es esencial para garantizar la confiabilidad del emulador. Siendo los tests de integración de software y de aceptación los más relevantes, ya que ambos permiten verificar el comportamiento del emulador en su conjunto.
- Cambiabilidad: Esta característica se evalúa predominantemente en los tests de aceptación, ya que en ellos se puede garantizar que el emulador puede ser actualizado y adaptado de manera sencilla.
- Usabilidad: Se evalúa en todos los niveles de test, siendo los tests de aceptación los más relevantes, ya que en ellos se puede garantizar que el emulador es fácil de usar y que se puede acceder de manera rápida y sencilla a información relevante sobre su estado.

## Estrategia por nivel de prueba

### 5.1. División del sistema en subsistemas

A continuación se determinan los subsistemas de software en los que se divide el emulador del procesador. Dichos subsistemas son descritos en mayor detalle en la entrada **AD.02** de la Tabla 1.

- Subsistema A, CPU: Modelo del procesador Leon3, encargado de interpretar cada instrucción y ejecutarla. Dicho subsistema es el núcleo del emulador, y se comunica con el resto de los subsistemas.
- Subsistema B, Memoria: Encargado de gestionar la memoria emulada del sistema, permitiendo la lectura y escritura de datos. Dicho modelo también permite el mapeo de dispositivos de entrada/salida.
- Subsistema C, Registros: Encargado de gestionar los registros del procesador, permitiendo la lectura, escritura y volcado de los mismos.
- Subsistema D, Scheduler y TimeKeeper: Encargado de gestionar el tiempo y la planificación de tareas, permitiendo la emulación de sistemas en tiempo real.
- Subsistema E, API: Encargado de su instanciación, configuración y de gestionar su comunicación con otros dispositivos, permitiendo el control del mismo.

### 5.2. Importancia relativa de los subsistemas

Subsistema	Importancia relativa (%)
CPU	30
Memoria	15
Registros	30
Scheduler & TimeKeeper	10
API	15
<b>TOTAL</b>	<b>100</b>

### 5.3. Determinación de la importancia de test por combinaciones de subsistema y características de calidad

-	Sub. A	Sub. B	Sub. C	Sub. D	Sub.E
<b>Importancia relativa</b>	30 %	15 %	30 %	10 %	15 %
<b>Funcionalidad</b>	++	+	++		+
<b>Confiabilidad</b>	++	+	++	++	
<b>Cambiabilidad</b>					++
<b>Usabilidad</b>	+	+	+	+	++

Referencias:

- ++: La característica de calidad es predominante para el subsistema.
- +: La característica de calidad es relevante para el subsistema.
- Vacío: La característica de calidad no es relevante para el subsistema.