



# Emulador de microprocesador Leon3 para desarrollo de software satelital y simuladores

Especificación de casos de uso

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  
Santiago Salamandri.

*Este documento fue realizado en el curso Ingeniería de Software  
el 16 de noviembre de 2023, tercer bimestre.*

## Índice

Introducción a los casos de uso . . . . .	4
---	---

## Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	16 de noviembre de 2023

## Introducción a los casos de uso

El presente documento describe los casos de uso del emulador de microprocesador Leon3 para desarrollo de software satelital y simuladores. Se presentan tres casos de uso representativos al uso nominal del software propuesto:

1. Cargar software de vuelo.
2. Inspeccionar registros y memorias.
3. Validar programa introducido al emulador.

Cuadro 1. Caso de Uso: Cargar software de vuelo [NEMU-UC-01]

Título	Descripción
<b>1. Nombre</b> 1.1 código 1.2 Breve descripción 1.3 Actor principal 1.4 Disparadores	<b>Cargar software de vuelo</b> <b>[NEMU-UC-01]</b> Capacidad de carga de software de vuelo arbitrario. Desarrolladores de FSW y operadores. Llamada a funcion por API.
<b>2. Flujo de eventos</b>  2.1 Flujo básico       2.2 Flujo alternativo	<ol style="list-style-type: none"><li>1. Se inicia una instancia de emulación.</li><li>2. Se llama a la funcion de API para la carga de binarios.</li><li>3. El software dado es cargado en la memoria emulada y se actualizan los registros correspondientes (Como contadores de programa).</li><li>4. La función retorna al usuario.</li></ol> <ul style="list-style-type: none"><li>■ Ante un binario inválido, como podría ser por compilador inválido o arquitectura incorrecta, la función retornará un código de error.</li><li>■ Ante un archivo no existente, la función retornará un código de error.</li></ul>
<b>3. Requerimientos especiales</b>	N/A.
<b>4. Pre-Condiciones</b>	Se posee un binario para ejecutar dentro del emulador.
<b>5. Post-Condiciones</b>	Contador de programa actualizado y programa cargado en memoria.

Cuadro 2. Caso de Uso: Inspeccionar registros y memorias [NEMU-UC-02]

Título	Descripción
<b>1. Nombre</b> 1.1 código 1.2 Breve descripción 1.3 Actor principal 1.4 Disparadores	<b>Inspeccionar registros y memorias</b> <b>[NEMU-UC-02]</b> Acceso a los registros y memorias emuladas, tanto para verificación de funcionamiento como para desarrollo de drivers. Desarrolladores de FSW y Desarrolladores de modelos. Llamada a funcion por API.
<b>2. Flujo de eventos</b>  2.1 Flujo básico   2.2 Flujo alternativo	<ol style="list-style-type: none"><li>1. Se inicia una instancia de emulación.</li><li>2. Se llama a la funcion de API para la carga de binarios.</li><li>3. Se llama a la funcion de API para volcar los registros a archivo CSV.</li><li>4. Alternativamente se puede llamar a la funcion de volcar memoria, especificando la posición initial, el tamaño a volcar y el archivo de salida para su analisis.</li></ol> <ul style="list-style-type: none"><li>■ Ante una posición de memoria o memoria invalida la función retornará un código de error.</li></ul>
<b>3. Requerimientos especiales</b>	N/A.
<b>4. Pre-Condiciones</b>	Se posee un binario para ejecutar dentro del emulador.
<b>5. Post-Condiciones</b>	Se tendrá una instancia del emulador con un binario cargado y un archivo en formato CSV con los datos solicitados.

Cuadro 3. Caso de Uso: Validar programa introducido al emulador [NEMU-UC-03]

Título	Descripción
<b>1. Nombre</b> 1.1 código 1.2 Breve descripción 1.3 Actor principal 1.4 Disparadores	<b>Validar programa introducido al emulador [NEMU-UC-03]</b> El emulador permitirá ejecutar y validar binarios producidos para el microprocesador. Desarrolladores de FSW y Desarrolladores de modelos. Llamada a funcion por API.
<b>2. Flujo de eventos</b>  2.1 Flujo básico  2.2 Flujo alternativo	  1. Se inicia una instancia de emulación.  2. Se llama a la funcion de API para la carga de binarios.  3. Se analiza si el binario dado realiza las operaciones esperadas.  N/A.
<b>3. Requerimientos especiales</b>	N/A.
<b>4. Pre-Condiciones</b>	Se posee un binario para ejecutar dentro del emulador con un comportamiento conocido.
<b>5. Post-Condiciones</b>	Se tendrá una instancia del emulador con un binario cargado.