Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y Naturales y Agrimensura

Maestría en Tecnologías de la Información

Trabajo Final

Construcción de una metodología de gestión de requerimientos software y desarrollo de un ecosistema de herramientas de acuerdo con la norma EN-50128. Aplicación en el desarrollo de un prototipo para la Autoridad Ferroviaria Nacional

Autor: Cristian Pinto Luft

Director: Emanuel Irrazábal

Año 2019

Dedicado a:

Mi profesor orientador, Emanuel Irrazábal, quien siempre me brindó su ayuda, conocimiento, apoyo y aliento para poder concretar este Trabajo Final de Maestría.

A mi familia, quienes siempre estuvieron conmigo, en los momentos buenos y malos, y nunca dejaron de creer en mí.

A mis amigos, quienes me brindaron su apoyo y los momentos de ocio tan necesarios algunas veces durante el proceso.

Al grupo de investigación, con quienes compartimos proyectos y momentos, haciendo de este trabajo una experiencia muy enriquecedora.

Resumen

El objetivo principal del Trabajo Final de Maestría es desarrollar un procedimiento de gestión de requerimientos software para sistemas críticos ferroviarios que cumpla la norma europea EN-50128, integrarlo a un proyecto real de gestión de calidad del software y aplicarlo a un caso específico: un monitor de barreras ferroviarias. Para la creación del procedimiento se utilizó una metodología de gestión de proyectos evolutiva incremental, dividiéndola en tres etapas: primero se analizó el contexto y el estado del arte, se fundamentó y se definió el alcance del proyecto. Luego se desarrolló y mejoró el procedimiento en diversas etapas, se crearon los registros de implementación generados por el mismo, y se verificó su cumplimiento con la normativa mencionada. Posteriormente se realizaron implementaciones del procedimiento con requerimientos reales enviados por los stakeholders. Por último, se establecieron conclusiones y futuras líneas de trabajo.

Palabras claves

Ingeniería de software, sistemas críticos ferroviarios, EN-50128, requerimientos software, desarrollo de procesos.

Abstract

The main goal of this master's thesis is to develop a software requirements management procedure for critical railway systems, that meets the European standard EN-50128, integrate it into a real world software quality management project and apply it to a specific case: a rail barriers monitor. An incremental evolutionary project management methodology was used for the creation of the project, which was divided into three phases: firstly, the context and the state of the art were analyzed, and the scope of the project was founded and defined. Secondly, the procedure was developed and improved in several phases, the implementation registries generated by it were created, and compliance with the aforementioned standard was verified. After this, procedure implementations were made according to the stakeholders' requirements. Lastly, conclusions and future lines of action were established.

Keywords

Software engineering, safety-critical railway systems, EN-50128, software requirements, process development.

Agradecimientos

El financiamiento del Trabajo Final de Maestría fue realizado por el proyecto PI-F17-2017 "Análisis e implementación de tecnologías emergentes en sistemas computacionales de aplicación regional", acreditado por la Secretaría de Ciencia y Técnica de la Universidad Nacional del Nordeste (UNNE) para el periodo 2018-2021.

Índice de Contenidos

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1. Introducción

1.1. Fundamentación

El sistema público ferroviario argentino se encuentra centralizado, y aunque se percibe como poco importante constituye un eslabón fundamental para la industria. En Argentina cada día tres millones de personas viajan en tren o subte y el 10% del PBI se moviliza por ferrocarril [1]. Sin embargo, todos los sistemas electrónicos para la seguridad vial de trenes y subtes son importados y muy caros. Por ejemplo, un sistema de barrera automático cuesta hasta 200.000 dólares y un sistema de control de velocidad más de 100.000 dólares. Así, en muchos trenes, no hay sistemas de seguridad para pasajeros, conductores, peatones y automovilistas y en otros, se siguen usando tecnologías de hace más de 50 años, que en los países con alto desarrollo tecnológico han sido reemplazadas hace mucho tiempo [2]. Esta situación ha favorecido que ocurran terribles accidentes [3] y ha urgido al Estado a adquirir en el exterior trenes y sistemas de seguridad ferroviaria, lo que implica enormes gastos en dólares y dependencia de tecnología extranjera [4][5][6]. La mayoría de los accidentes se podrían haber evitado mediante el uso de sistemas electrónicos apropiados, que hoy en día son habituales en los países anteriormente mencionados. Sin embargo, como se mencionó, en la actualidad estos sistemas no se desarrollan en la Argentina.

Aun así, existen proyectos de investigación y de extensión que se encuentran actualmente trabajando en las problemáticas mencionadas. Un ejemplo es el Proyecto de Desarrollo Estratégico UBA Nº23 “Controlador electrónico para barreras automáticas ferroviarias con nivel de integridad de seguridad certificable hasta SIL4", desarrollado por el Dr. Ariel Lutenberg, director del Programa Computadora Industrial Abierta Argentina (CIAA) [7]. El objetivo de este proyecto ha sido desarrollar un prototipo de Monitor de Barrera ferroviaria construido a partir de normas internacionales y componentes electrónicos programables, en este caso la CIAA. El propósito del proyecto mencionado es desarrollar los sistemas de gestión y la construcción del ecosistema de herramientas que lo instrumenten, así como su verificación y validación en ensayos junto con la Autoridad Ferroviaria Nacional, valiéndose de las investigaciones realizadas en el marco del Grupo de Investigación en Calidad y Seguridad de las Aplicaciones Ferroviarias

(GICSAFe), del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) de la República Argentina [8].

En este marco es donde se inserta el presente Trabajo Final de Maestría, cuyo propósito es el desarrollo de un procedimiento para la gestión de requerimientos[1] software críticos ferroviarios producidos a nivel nacional, proponiendo una metodología de trabajo bien analizada, definida, y que reúna experiencias de proyectos de similar índole. Dicho procedimiento debe cumplir con los puntos definidos en la normativa europea EN-50128: "Aplicaciones ferroviarias. Sistemas de comunicación, señalización y procesamiento. Software para sistemas de control y protección del ferrocarril", la cual indica una forma de gestionar el software crítico ferroviario para cumplir con ciertos estándares que garanticen un determinado nivel de calidad y seguridad del producto [9].

1.2. Objetivo general

Desarrollar un procedimiento de gestión de requerimientos software para sistemas críticos ferroviarios, bajo la normativa EN-50128, que pueda ser aplicado a un ecosistema de gestión de calidad en sistemas ferroviarios.

1.3. Objetivos específicos

Para cumplimentar el objetivo general definido en el apartado anterior, se han propuesto la realización de determinados objetivos específicos, que lo componen y se indican a continuación.

1.3.1. (OE1): Investigación del marco teórico sobre la gestión de requerimientos software en sistemas críticos ferroviarios

Investigar y evaluar los métodos y las herramientas utilizadas actualmente a nivel mundial para la gestión de requerimientos software en sistemas críticos ferroviarios, mediante la realización de una Revisión Sistemática de la Literatura (RSL). En esta investigación se pretenden descubrir qué tipos o módulos de sistemas ferroviarios se implementan generalmente y por qué, qué

[1] En este documento y en el procedimiento realizado se utilizan indistintamente las palabras "requerimiento" y "requisito" cuando se hace referencia a los de software, distinguiendo entre los **funcionales** (aquellos que tratan sobre las funciones del software en sí) y los **no funcionales** (aquellos que tratan características del software). Además la palabra "requisito" se utiliza también haciendo referencia a los que impone la norma EN-50128.

metodologías y herramientas tecnológicas son usadas en este proceso, como se trata el concepto de seguridad y que normativas son las que se intentan cumplir.

1.3.2. (OE2): Desarrollo del procedimiento de gestión de requerimientos software en sistemas críticos ferroviarios

Desarrollar un procedimiento de gestión de requerimientos software en sistemas críticos ferroviarios que se adapte a la normativa EN-50128, y a niveles generales con las buenas prácticas de la EN-50126, e incluirlo en el proyecto CIAA, del GICSAFe. El mismo deberá constar de actividades que garanticen la seguridad y calidad del software a desarrollar, aplicando las técnicas, metodologías y herramientas obtenidas con los resultados del cumplimiento del OE1.

1.3.3. (OE3): Implementación del procedimiento desarrollado de gestión de requerimientos software en sistemas críticos ferroviarios

Aplicar el procedimiento desarrollado a un proyecto piloto, en el diseño de un monitor de barreras ferroviarias para la Autoridad Ferroviaria Nacional, enmarcado dentro de los límites del sistema de gestión de calidad, con la finalidad de probar su correctitud y madurez, dentro de un proceso de mejora continua. La implementación del procedimiento deberá ser validada por los stakeholders, con lo que se espera lograr un proceso de retroalimentación que permita la mejora de dicho procedimiento.

Capítulo 2

Metodología

2. Metodología

2.1. Metodología general

Para la construcción del procedimiento se utilizó un modelo de ciclo de vida evolutivo incremental [10], como se puede ver en la Fig. 1. Se tomó esta decisión porque los distintos aspectos considerados en la generación del procedimiento de gestión de requerimientos software se fueron incorporando gradualmente al mismo, incrementando su nivel de especificidad, eficiencia y seguridad. Las actividades que se realizaron fueron:

1. Análisis de la problemática en cuestión y del contexto.
2. Creación del procedimiento de gestión de requerimientos software.
3. Verificación de cumplimiento con la norma EN-50128.
4. Primera implementación del procedimiento con requerimientos reales.
5. Agregado de técnicas de seguridad al procedimiento.
6. Segunda implementación del procedimiento con requerimientos reales.
7. Agregado de técnicas de métodos semi formales y formales al procedimiento.
8. Tercera implementación del procedimiento con requerimientos reales.
9. Integración del procedimiento con Eclipse Process Framework (EPF).
10. Verificación del cumplimiento de la norma EN-50128 de los requerimientos tratados.

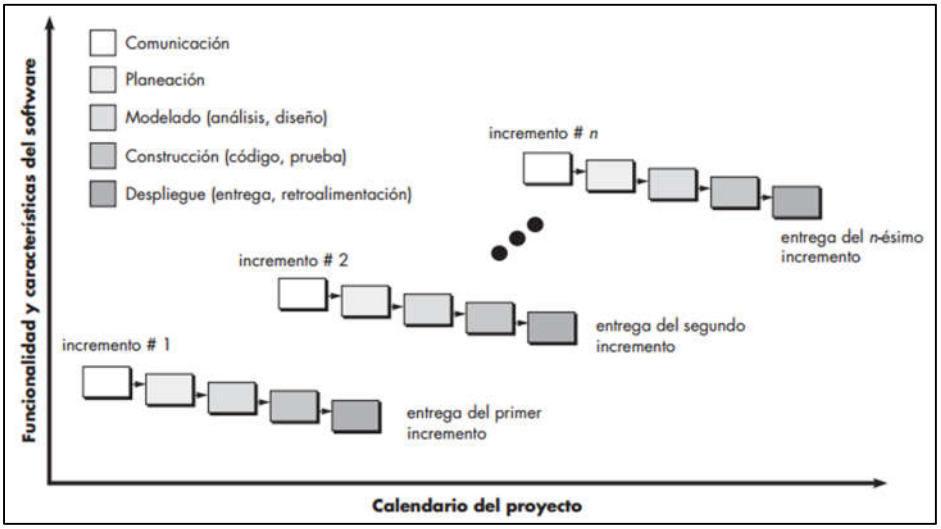


Fig. 1. Ciclo de vida evolutivo incremental. Fuente: [10]

En la Tabla 1 se indica el cronograma utilizado para la realización de las actividades anteriormente descritas, indicando en las columnas los meses y en las filas los números correspondientes a la numeración dada a estas actividades. Cabe aclarar que algunas de las mismas se solaparon, y que, por la naturaleza evolutiva incremental del proyecto, algunas se volvieron a modificar posteriormente a su concreción, adaptándolas a los nuevos cambios o avances que fueron surgiendo.

Tabla 1. Cronograma de actividades

Tarea/ Mes	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											

El método de investigación que se sigue en este trabajo es la adaptación del propuesto por Marcos & Marcos en [11] para la investigación en Ingeniería del Software. Dicho método se basa en el hipotético-deductivo propuesto por Bunge [12] y se compone de una serie de pasos lo suficientemente generales como para ser aplicado a cualquier proyecto de ingeniería del software, como se puede ver en la Fig. 2.

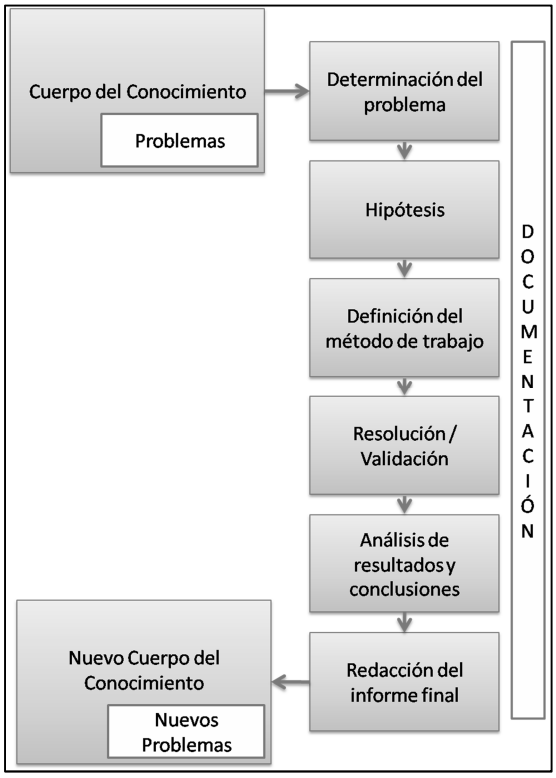


Fig. 2. Metodología. Fuente: [11]

A continuación, se brinda una descripción general de cada uno de estos pasos:

- **Cuerpo del conocimiento**: se introdujeron las bases teóricas en las que se enmarca este trabajo, es decir, el estado del arte, con base en la problemática existente al día del inicio de la misma. Para esto, se realizó una RSL con preguntas de investigación relacionadas al tema en cuestión.
- **Determinación del problema**: a partir de la información obtenida en el paso anterior y la recolección de la existente del problema puntual que se quiere abordar, se ha realizado la definición del problema de este trabajo. Para esto, se realizó un análisis del proyecto en el que el mismo se encuentra involucrado, con el fin de lograr un entendimiento general de las aristas que componen al mismo.
- **Hipótesis**: se ha definido una hipótesis como forma de abordaje de la solución a la problemática presentada y sus resultados esperados, intentando cumplir con el objetivo general, y por consiguiente, los específicos, propuestos en este documento. Esto es, la hipótesis de que, utilizando la información encontrada en la determinación del cuerpo del

conocimiento existente, se logre generar un procedimiento de gestión de requerimientos software para sistemas críticos ferroviarios, teniendo en cuenta los hallazgos más significativos resultantes. Además, dicho procedimiento debe cumplir con las restricciones encontradas en la determinación del problema puntual, es decir, utilizar la tecnología disponible (tanto a nivel hardware como software), adaptarse a la metodología de trabajo del proyecto en el que se encuentra y tratar de disminuir la problemática definida durante el desarrollo del presente Trabajo Final de Maestría.

- **Definición del método de trabajo**: Como se puede ver en la Fig. 2, este es un paso más del mismo método. Los autores [11] recomiendan definirlo de esta manera ya que, la naturaleza de cada investigación tiene sus propias características y por lo tanto, no sería conveniente aplicar un único método universal de investigación.
- **Resolución / Validación**: se dividieron conceptualmente ambos pasos en:
  - **Resolución**: compone a las actividades de desarrollo del procedimiento y su uso mediante implementaciones con requerimientos reales enviados por los stakeholders (usuarios finales).
  - **Validación / Verificación**: compone a las validaciones realizadas con el experto en el área del conocimiento (el profesor orientador) y los stakeholders. Además, se incluyen las verificaciones de cumplimiento del procedimiento con la normativa utilizada como referencia (EN-50128).
- **Análisis de resultados y conclusiones**: compone a las actividades de comparación de los objetivos propuestos, con las hipótesis planteadas y los resultados reales a los que se llegó con la realización del trabajo. Además, se informan los resultados parciales a los que se llegó mediante el desarrollo del Trabajo Final de Maestría, como ser, las publicaciones de información generada con respecto al tema en distintos congresos nacionales.
- **Redacción del informe final**: involucra la redacción del presente documento, siguiendo los lineamientos y revisiones dictados por la cátedra.
- **Nuevo cuerpo del conocimiento**: a partir de la realización del presente Trabajo Final de Maestría, se introdujo información al cuerpo de conocimiento existente, haciéndola pública mediante la publicación de trabajos generados a partir de este trabajo final. Esto

trajo aparejadas nuevas soluciones y casos de aplicación, y a su vez nuevos problemas, al introducir más complejidad en el dominio de la problemática estudiada y desarrollada.

Cabe destacar que cada paso fue debidamente documentado, generando evidencia empírica de su aplicación. Además, como se definió en un principio, al utilizarse un modelo de ciclo de vida evolutivo incremental, los mismos no se llevaron a cabo en un orden meramente secuencial, sino que existieron iteraciones: se realizaron tres incrementos (o "entregas"); para cada una de ellas se realizaron todos los pasos nuevamente desde el principio.

Durante la realización de este trabajo se generaron tres versiones distintas del procedimiento de gestión de requerimientos software, cuyas características principales se pueden resumir en:

- Versión inicial del procedimiento.
- Procedimiento con elementos de seguridad agregados.
- Procedimiento con métodos formales agregados.

Capítulo 3

Estado de la Cuestión

3. Estado del arte

3.1. Estado de la cuestión

Los sistemas ferroviarios son sistemas complejos, compuestos por distintos componentes software, hardware y humanos, que interactúan con su entorno de maneras muy variadas. Un fallo en uno de estos componentes o subsistemas puede llegar a tener asociados distintos niveles de peligros, pudiendo causar pérdidas financieras, daño al equipamiento, daños ambientales, lesiones a personas o en los peores casos pérdidas de vidas humanas. Es por estos motivos que estos sistemas se encuentran regulados por distintas leyes y normativas en distintos países, con el fin de preservar los recursos anteriormente mencionados [13]. Algunos de los principales organismos que regulan esta actividad son el Comité Européen de Normalisation Electrotechnique (CENELEC) en Europa o la International Electrotechnical Commission (IEC) en América.

Unas de las características más importantes de los sistemas que estas normas intentan reforzar durante todo su ciclo de vida son las de fiabilidad, disponibilidad, mantenibilidad y seguridad (RAMS por sus siglas en inglés).

Las principales normas propuestas por el CENELEC orientadas a la resolución de la problemática explicada anteriormente son las siguientes, cuyas relaciones se pueden observar en la Fig. 3:

- **UNE-EN-50126 [14]**: Aplicaciones ferroviarias. La especificación y demostración de Fiabilidad, Disponibilidad, Mantenibilidad y Seguridad (RAMS). Esta norma se orienta principalmente al cumplimiento de las características RAMS del sistema en general.
- **UNE-EN-50128 [9]**: Aplicaciones ferroviarias. Sistemas de comunicación, señalización y procesamiento. Software para sistemas de control y protección del ferrocarril. Esta norma se centra principalmente en la calidad de los aspectos software de los sistemas de ferrocarriles.
- **UNE-EN-50129 [15]**: Aplicaciones ferroviarias. Sistemas de comunicación, señalización y procesamiento. Sistemas electrónicos relacionados con la seguridad para la señalización. Esta norma se centra principalmente en los aspectos de calidad del hardware de los sistemas de ferrocarriles.

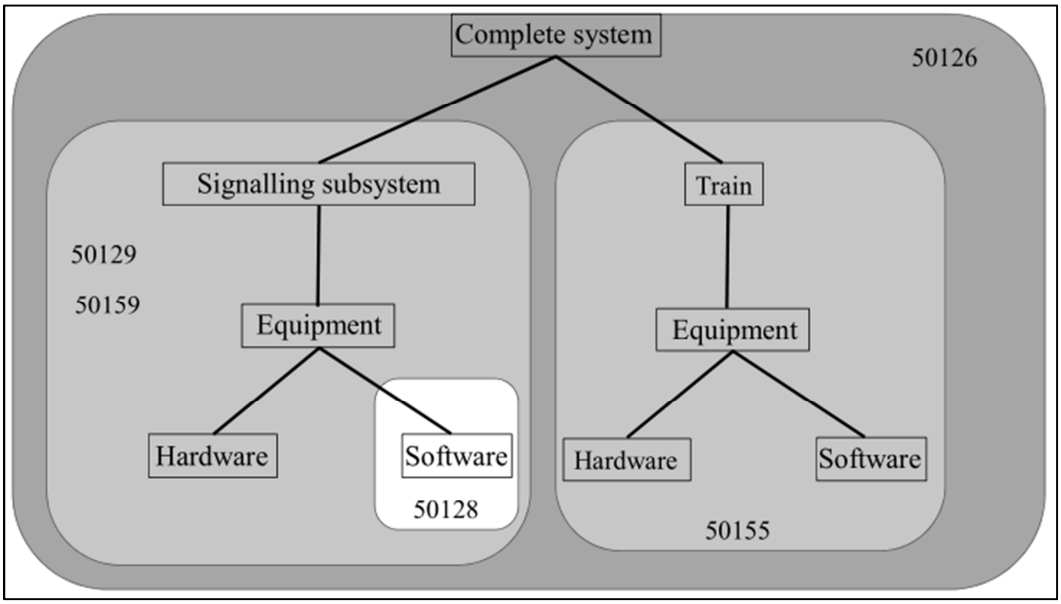


Fig. 3. Normas ferroviarias CENELEC. Fuente: [13]

Una de las características principales de los sistemas críticos es la seguridad de la que estos deben estar dotados por naturaleza, debido a las consecuencias que pueden provocar los fallos que pudieran llegar a ocurrir. Para dotarlos de seguridad, una de las metodologías utilizadas en su diseño es el aseguramiento de los mismos desde su concepción, es decir, desde el análisis y definición de sus requerimientos. Para esto se utilizan enfoques que integran las disciplinas de ingeniería de requerimientos con la ingeniería de seguridad, lo cual está comprobado que aumenta significativamente la seguridad del sistema en sí [16].

Para llegar a cumplir con los requerimientos de seguridad del sistema a niveles generales, se deben cumplir con los mismos en cada uno de los subsistemas que lo componen, aumentando el nivel de granularidad y disminuyendo la abstracción a medida que se va descendiendo hacia los niveles más bajos. Es decir, esta vinculación entre la ingeniería de requerimientos y la de seguridad también debe ser aplicada a los subsistemas que componen al sistema.

En la actualidad, grandes organizaciones como la NASA [17], Ansaldo Signal [18] o Siemens Rail Transportation [19] utilizan una combinación de metodologías y formas de trabajo provenientes de distintos campos del conocimiento para lograr dicha vinculación, y de esta

manera mejorar la calidad y seguridad de los sistemas críticos que desarrollan, dedicando tiempo, recursos y esfuerzo a esta tarea.

Este trabajo utiliza principalmente los enfoques propuestos por las normas EN-50126 y EN-50128, gestionando las políticas RAMS vinculadas a los requerimientos de los subsistemas software que componen a los sistemas ferroviarios, y haciendo hincapié en los aspectos de seguridad de las mismas. Dicho procedimiento es aplicado al diseño del monitor de barreras como prueba piloto de su funcionamiento y correctitud, y como experiencia para su mejora continua y mantenimiento. Además, mediante la generación de un procedimiento general de gestión de requerimientos software, se espera poder cubrir dichos aspectos en otros proyectos que se puedan presentar eventualmente en el marco de los sistemas ferroviarios argentinos.

3.2. Antecedentes del proyecto

Este trabajo fue realizado en el contexto del grupo de trabajo GICSAFe creado en el año 2017 por el CONICET [8]. Este organismo está conformado por investigadores, docentes, profesionales y estudiantes de las siguientes instituciones:

- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas) [20]
- CNEA (Comisión Nacional de Energía Atómica) [21]
- UBA (Universidad de Buenos Aires) [22]
- UNCA (Universidad Nacional de Catamarca) [23]
- UNNE (Universidad Nacional del Nordeste, Corrientes) [24]
- UNT (Universidad Nacional de Tucumán) [25]
- UTN-FRBB (Universidad Tecnológica Nacional - Facultad Regional Bahía Blanca) [26]
- UTN-FRH (Universidad Tecnológica Nacional - Facultad Regional Haedo) [27]
- Vortex (Emprendimiento Tecnológico, Mar del Plata)

Los principales objetivos del GICSAFe son:

- El desarrollo de sistemas electrónicos e informáticos para aplicaciones ferroviarias relacionadas con la seguridad.
- La instalación de prototipos de los sistemas desarrollados y elaboración de toda la documentación correspondiente.

- La transferencia del derecho de uso, fabricación y mantenimiento a sus clientes.

Para la concreción de estos objetivos, se aplican normas internacionales de seguridad en sistemas ferroviarios, como ser las EN-50126, EN-50128 y EN-50129, y la norma de gestión de calidad ISO 9001 [28], lo que la convierte en la primera organización en Argentina en aplicar este tipo de normas para el desarrollo de este tipo de sistemas. El cumplimiento de estas normas es necesario para el desarrollo y la implementación de este tipo de soluciones, y esto a su vez conlleva altos costos, dada la complejidad de dicha tarea y el valor agregado del que disponen los productos finales.

En el proyecto, para permitir auditar la calidad de la que dispone el mismo, y demostrar el cumplimiento de las RAMS, se utilizó la metodología de pirámide documental, como se puede ver en la Fig. 4.

Fig. 4. Pirámide documental

Como se puede apreciar, en cada nivel, en orden descendente:

- **Nivel 1: Manual de Calidad**. Se aplican los procesos definidos por la norma ISO 9001, indicando la evidencia de gestión de calidad del proyecto.
- **Nivel 2. Procedimientos de Soporte y Generales**. Se aplica la norma EN-50126,

indicando la evidencia de que se crearon procedimientos.

- **Nivel 3. Procedimientos específicos**. Aplicando los procedimientos específicos generados mediante la norma correspondiente (EN-50128 o EN-50129).
- **Nivel 4. Registros de Calidad**. Se evidencia la aplicación de los procedimientos definidos, mediante los registros generados en cada uno de ellos.

Actualmente casi todo lo que concierne a materia ferroviaria en el país, tanto a nivel de hardware como de software, son productos importados de empresas multinacionales, de otros países, con costos muy elevados de adquisición y de soporte. El hecho de comprar soluciones a proveedores extranjeros conlleva además a una dependencia del mismo de entre 20 y 60 años, incurriendo en inversiones de mantenimiento más costosas que la inicial (adquisición del producto). Una de las principales ventajas de la aplicación de este tipo de soluciones a nivel local, es la sustitución de estas importaciones, abaratando los costos de desarrollo y permitiendo la generación de trabajo a nivel regional con un alto valor agregado.

Se tienen registro de accidentes ferroviarios evitables en el país, causados por el mal diseño o la falta de controles y mantenimientos adecuados [3], como ser los que se muestran en la Tabla 2.

Tabla 2. Accidentes ferroviarios en Argentina. Fuente: [3]

Ciudad	Año	Fallecidos	Heridos	Descripción	¿Evitable?
Benavidez	1970	236	400	Choque de trenes	Evitable
Castelar	2013	3	315	Choque de trenes	Evitable
Flores	2011	11	228	Choque con ómnibus	Evitable
Once	2012	51	702	Choque con fin de vía	Evitable
Merlo	2017	2	14	No bajaron la barrera	Evitable

El presente Trabajo Final de Maestría tiene como principal stakeholder a la empresa Trenes Argentinos Operaciones (SOFSE) [29], la cual es una sociedad del Estado argentino creada en 2008, encargada de la prestación de los servicios de transporte ferroviario que le sean asignados, tanto de cargas como de pasajeros, en todas sus formas, incluyendo el mantenimiento del material

rodante. Integra al grupo Trenes Argentinos [30] junto a otras empresas estatales del sector ferroviario, y es la segunda mayor empresa pública del país, con 23000 empleados aproximadamente. La misma cuenta con cinco líneas metropolitanas, aproximadamente 7000 kilómetros de red ferroviaria en todo el país, 12 provincias conectadas, 439 estaciones, casi 2000 servicios diarios y 341 mil millones de pasajeros.

Durante la realización del proyecto, se lograron varios hitos intermedios, como ser:

- Obtención del premio ganador en el programa "Eureka, Desafío de Ideas II", en la categoría “Tecnología”, emitido en 2018 por Canal Encuentro y la TV Pública [31].
- Obtención del premio ganador en la categoría “Investigadores” del concurso INNOVAR 2018 y el premio general del mismo [32].
- Siete artículos publicados en distintos congresos [8].
- Implementación de sistemas monitores de barreras con los stakeholders [8].
- Divulgación del conocimiento obtenido en distintas universidades y por distintos medios, entre otros.

Por todo lo mencionado anteriormente, y teniendo en cuenta el plan de inversión en materia ferroviaria en el periodo 2016-2023 del país (14 mil millones de dólares) [33], el proyecto es considerado viable, innovador y presenta un triple impacto: económico, ambiental y social. Se intenta demostrar además que, de reemplazarse la compra de estas soluciones y producirse las mismas en el marco de este proyecto, el costo sería del 30% de la inversión planificada.

3.3. Monitor de barreras ferroviarias - Proyecto DIMBA

El grupo de investigación CONICET-GICSAFe tuvo como primer proyecto de aplicación un sistema de monitoreo automático de barreras ferroviarias, denominado Diseño e Implementación del Monitor de Barreras (DIMBA).

Una barrera ferroviaria es un elemento de seguridad vial que impide el paso de vehículos en caminos que intersectan una vía ferroviaria (o pasos a nivel, PaN), de manera temporal, cuando un tren está pasando por ese punto, para evitar siniestros viales. Las mismas pueden ser de tipo manual (Fig. 5) o automática (Fig. 6): las primeras se accionan mediante un operador humano y las segundas mediante un mecanismo automático, y su mal funcionamiento puede provocar

incidentes viales, pudiendo generar daños a las saludes de las personas, económicas o ambientales, por lo que son considerados sistemas críticos.

Fig. 5. Barrera ferroviaria manual. Fuente: [3]

Fig. 6. Barrera ferroviaria automática. Fuente: [3]

En la República Argentina hay 14.000 cruces ferroviarios con paso a nivel. De este total sólo 3% cuenta con un sistema automático de control de barreras, mientras que 7% es accionado manualmente y el 90% restante no cuenta con ningún tipo de barrera [1].

En el proyecto DIMBA se diseñó e implementó un monitor de datos para barreras ferroviarias automáticas, con la finalidad de poder adaptar este tipo de tecnología al entorno regional, disminuyendo los costos de compra y mantenimiento, mejorando la fiabilidad, disponibilidad, mantenibilidad y seguridad del sistema ferroviario y creando una base de conocimientos sólida para la realización de proyectos de similar índole.

Este monitor fue diseñado para ser ubicado en los PaN (nivel monitor), leer determinadas señales de entradas provenientes de la barrera y transferir dichos datos mediante una interfaz de salida a una central remota de monitoreo (nivel de gestión, cuyo sistema informático se encuentra fuera del alcance de este proyecto). De esta forma, desde dicha central se pueden conocer y procesar los datos de la barrera en tiempo real, incluyendo los fallos que la misma pudiera presentar, y actuar en consecuencia de manera inmediata.

El monitor de barreras consiste en un circuito electrónico con un procesador central denominado CIAA-NXP [34], parte del Proyecto CIAA [7], ubicado dentro de un contenedor de metal que lo protege de las inclemencias climáticas y del vandalismo, generalmente llamado *abrigo*, que va conectado a la barrera, como se ve en la Fig. 7.

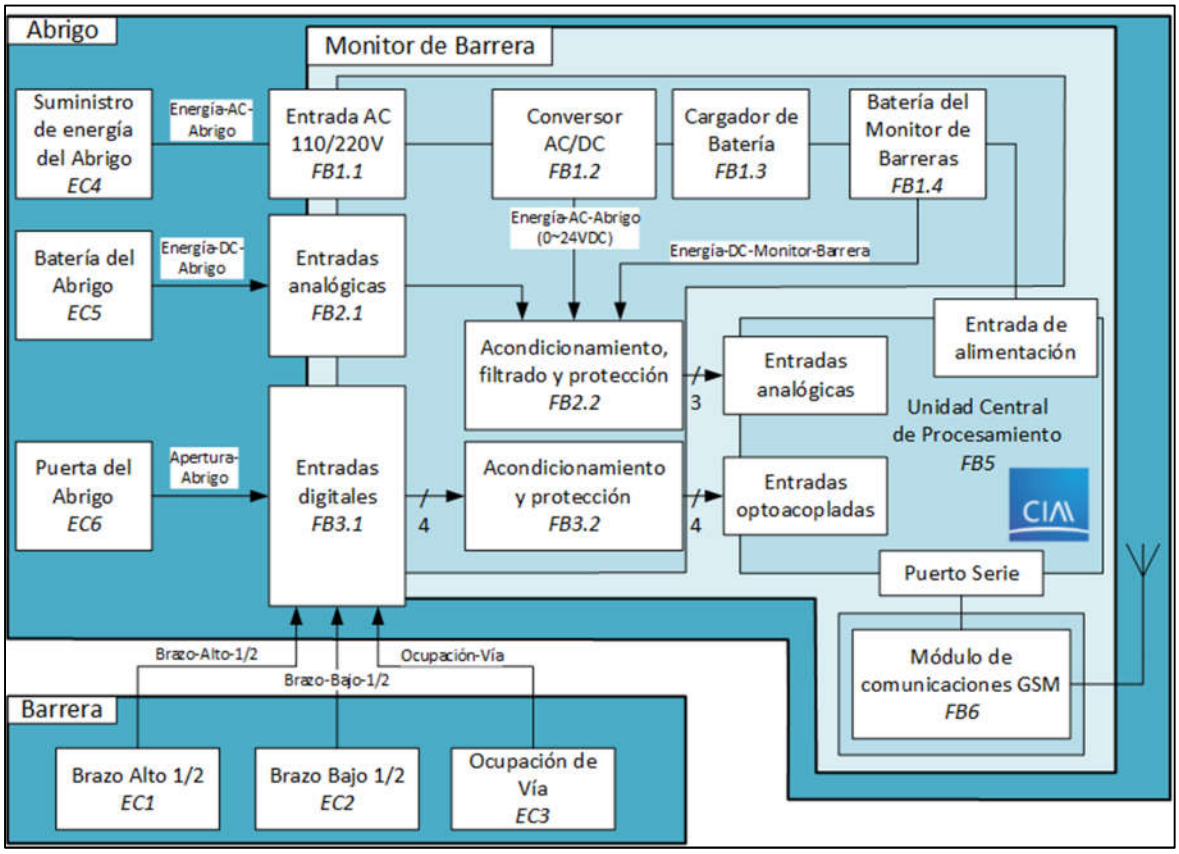


Fig. 7. Diagrama de bloques funcionales del monitor de barreras

Las señales a monitorear por el monitor de barreras se pueden ver en la

Tabla 3.

Tabla 3. Señales a monitorear

Nombre	Propósito	Punto de lectura
Brazo-Alto-1/2	Monitorear si el brazo está en la posición en alto	Contacto seco [a]
Brazo-Bajo-1/2	Monitorear si el brazo está en la posición en bajo	Contacto seco [b]
Ocupación-Vía	Monitorear si la vía está libre u ocupada	Contacto seco [c]
Apertura-Abrigo	Monitorear la apertura y	Contacto seco [d]

	cierre de la puerta de abrigo	
Energía-AC-Abrigo	Monitorear su valor en el rango 0 - 240 VAC	Conversor AC/DC en bloque FB1.2 [e]
Energía-DC-Abrigo	Monitorear su valor en el rango 0 - 24 VDC	Batería EC5 [e]
Energía-DC-Monitor-Barrera	Monitorear su valor en el rango 0 - 24 VDC	Batería en el bloque FB1.4 [e]

a. El contacto está abierto si el brazo está a menos de 83°, y cerrado si el brazo está más de 83°

b. El contacto está abierto si el brazo está a más de 5°, y cerrado si el brazo está a menos de 5°

c. El contacto está abierto si la vía está libre, y cerrado si la vía está ocupada

d. El contacto está abierto si la puerta del abrigo está abierta y cerrado si la puerta está cerrada

e. Los elementos FB1.2, EC5 y FB1.4 se presentan en la Fig. 7.

El diagrama de la arquitectura utilizado para construir el sistema se puede ver en la Fig. 8.

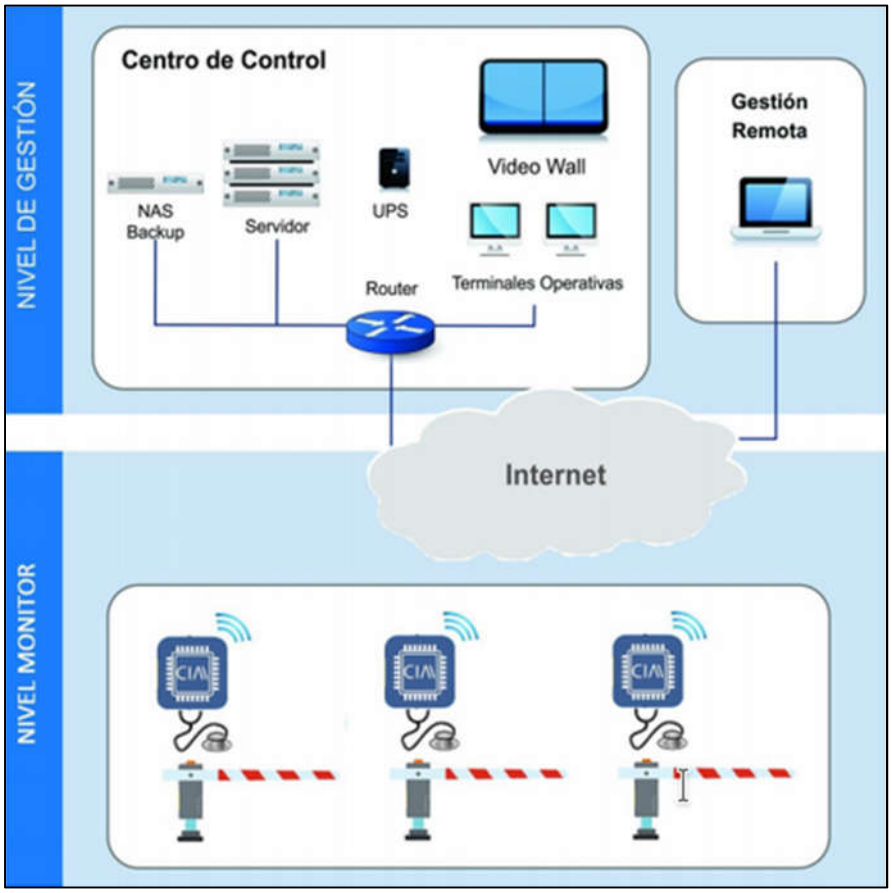


Fig. 8. Diagrama de arquitectura del sistema

El primer equipo monitor de barreras instalado se puede ver en la Fig. 9.

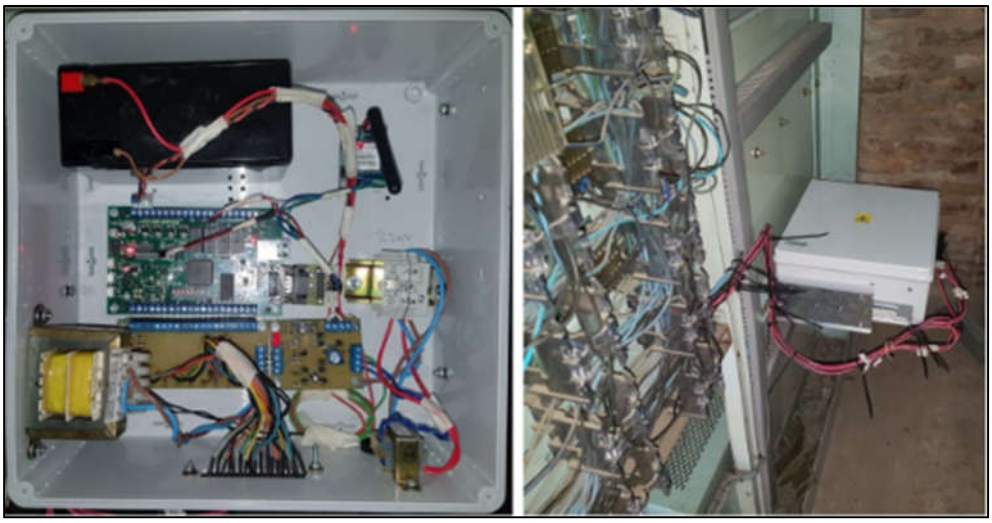


Fig. 9. Monitor de barreras instalado

A nivel software, se desarrolló el firmware necesario en lenguaje C para poder llevar a cabo la lectura y envío de datos de la barrera automática, en el nivel monitor, tal como se había descrito, en formato JSON y mediante GSM. La arquitectura del mismo con los componentes software usados y sus relaciones puede ser vista en la Fig. 10.

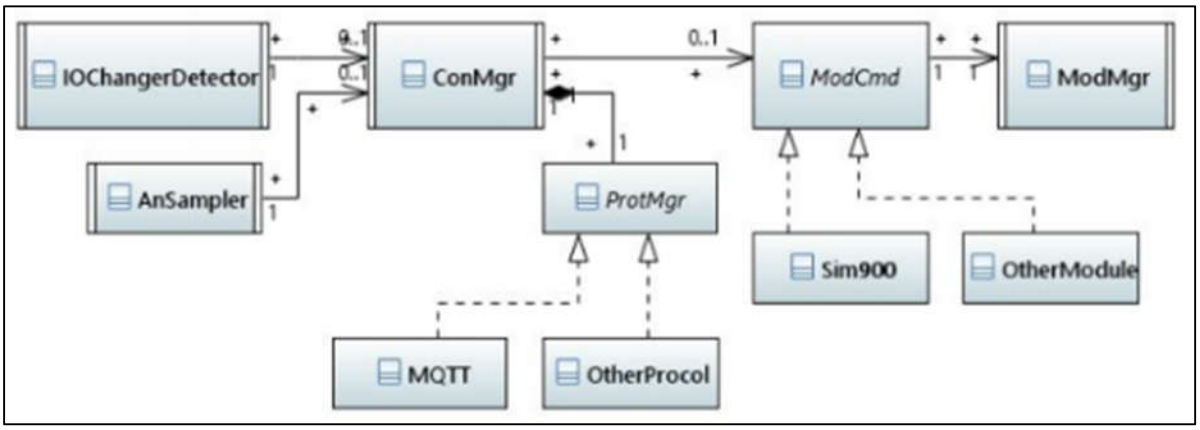


Fig. 10. Arquitectura del firmware

Como resultado de esta primera implementación, se logró colocar un monitor de barreras en el paso a nivel de la Línea Roca en Av. Espora, a 300 metros de la estación José Mármol, en el Partido de Almirante Brown, Buenos Aires, con un costo de 400 dólares. El mismo genera información que puede ser vista en la página web del CONICET-GICSAFe [35], como así también se puede ver en el nivel de gestión.

Se espera a futuro poder implementar junto a Trenes Argentinos este sistema monitor en otras barreras ferroviarias, además de incluir la lectura de otros dispositivos que componen a los PaN, como ser el motor, las lámparas y las campanas.

Con este Trabajo Final de Maestría se busca poder dar un paso más adelante con este monitor, como caso de aplicación puntual, gestionando los requisitos software mencionados mediante el Procedimiento de Gestión de Requisitos Software creado, de manera de poder elevar los estándares de calidad y seguridad de dicho monitor, cumpliendo lo indicado por la norma EN-50128.

3.4. Revisión Sistemática de la Literatura: aplicación de seguridad a requerimientos software de sistemas críticos ferroviarios

A continuación, se presenta una revisión sistemática de la literatura en la cual se realiza un análisis exhaustivo de la información existente en cuanto a la gestión de requerimientos software en sistemas críticos ferroviarios con la finalidad de conocer sus principales características de implementación. La revisión sistemática se realizó sobre cuatro repositorios académicos distintos (ACM, Science Direct, Springer e IEEE), obteniéndose un total de 23 publicaciones, las cuales han sido analizadas para obtener información categorizada en 5 dimensiones distintas para ayudar a su comprensión. Como conclusión, se informan los resultados de la aplicación de dicho procedimiento, indicando los principales hallazgos obtenidos de este análisis.

3.4.1 Introducción

Las RSL son herramientas fundamentales para la búsqueda, recolección, procesamiento y publicación del estado del arte de una temática particular, siguiendo una metodología sistémica, completa, explícita y reproducible, que puede ser utilizada con diversos fines. En lo que a esta RSL respecta, su foco principal se centra en investigar y apropiarse de los conocimientos más actualizados y relevantes relativos a la gestión de los requerimientos software de los sistemas críticos ferroviarios, centrándose principalmente en la seguridad. Esta última es uno de los atributos de calidad más importantes en estos tipos de sistemas, ya que de ellos pueden depender en gran medida la viabilidad económica y funcional del sistema, la conservación de su entorno e incluso la preservación de vidas humanas.

Si bien en el contexto de la Ingeniería de Requerimientos Software para sistemas críticos se puede encontrar gran cantidad de material de investigación y aplicación [36][37][38][39][40] , por la misma naturaleza crítica de estos sistemas, es importante conocer puntualmente el estado del arte para cada dominio antes de comenzar a definir un nuevo procedimiento de gestión. Esto es determinante en la seguridad (y en los aspectos RAMS en sí) del sistema software. Dichas particularidades incluyen aspectos como ser normativas y estándares [41][42] , técnicas de análisis y aseguramiento de la seguridad [43] , metodologías de análisis [44] , herramientas de soporte software [45], entre otras. En cada dominio de aplicación en particular, estos aspectos son

definidos empíricamente mediante el uso y aprendizaje histórico. Lo que es útil en un tipo dominio de aplicación (por ejemplo, en los sistemas aeroespaciales), puede no serlo en otros (como por ejemplo, en los sistemas ferroviarios), dadas las particularidades de cada dominio en sí [41].

Un aspecto importante a considerar es que, al tratarse de sistemas críticos, pueden llegarse a producir inconsistencias y llevar al sistema a posibles ocurrencias de fallas si los sucesos históricos no son analizados. Esta gestión de la seguridad de la que se debe dotar a los sistemas críticos es una de sus principales diferencias con otro tipo de sistemas, en donde su criticidad es menor.

El objetivo principal de la presente revisión es entonces establecer el estado del arte y marco teórico/práctico de la temática definida, con la finalidad de apoyar a la generación de un procedimiento general de gestión de requisitos software, cumpliendo con las buenas prácticas internacionales entre las que se pueden destacar la UNE-EN-50128:2012 o IEC 62279.

3.4.2. Trabajos relacionados

Anteriormente a la confección de la presente RSL, y conjuntamente con el estudio de diversos artículos de variada índole, se analizó el siguiente trabajo:

- En [A26] se presenta una RSL sobre la integración de la ingeniería de requerimientos y las técnicas de análisis de seguridad existentes. El objetivo principal de este trabajo es analizar los distintos enfoques de integración entre ambas metodologías utilizados en el mundo, con sus distintas variantes en cuanto a aplicabilidad.

En la RSL citada, se hace hincapié en la importancia de aplicar técnicas de análisis de seguridad en etapas tempranas del diseño de un sistema, integrándola a la gestión de sus requerimientos, para intentar disminuir la cantidad de errores posibles durante las siguientes etapas de su ciclo de vida. A diferencia de la anterior, la presente RSL intenta analizar específicamente la gestión de requerimientos software en sistemas críticos, centrándose principalmente en aquellos del tipo ferroviario y sus aspectos relativos a la seguridad, enfocándose en casos de aplicaciones exitosas de estos procedimientos.

3.4.3. Planificación de la RSL

El objetivo de esta sección es definir la metodología de trabajo a utilizarse para la realización de la presente RSL, especificando cada una de las etapas que la componen, y siguiendo el modelo propuesto por Barbara Kitchenham [46].

3.4.3.1. Elección de las preguntas de investigación

En la Tabla 4 se definieron las preguntas de investigación (PI) a formularse, para llevar a cabo la búsqueda de información relevante existente acerca de sistemas y metodologías de gestión de los requerimientos software en sistemas críticos ferroviarios, haciendo foco en la seguridad. Para puntualizar las respuestas a las interrogantes, se definieron para cada una de ellas una dimensión y un conjunto de atributos que corresponden a los aspectos más significativos a indagar en la construcción de la RSL.

Tabla 4. Preguntas de Investigación

PI	Descripción	Dimensiones
PI-1	¿Qué tipos o módulos de sistemas software se desarrollan utilizando metodologías de gestión de requerimientos software en sistemas críticos ferroviarios?	**Sistemas**: implementación, módulo, sistema, sub-sistema.
PI-2	¿Qué metodologías se utilizan para la gestión de requerimientos software en el desarrollo de sistemas críticos ferroviarios?	**Metodologías usadas**: métodos ágiles, formales, estructurados, mixtos.
PI-3	¿Qué software o herramientas se utilizan y cómo se logra la integración de las mismas en la gestión de requerimientos software en el desarrollo de sistemas críticos ferroviarios?	**Aplicaciones**: software, manual, integración, modelo.
PI-4	¿Cómo se gestiona en los sistemas de gestión de requerimientos software la seguridad de los sistemas críticos ferroviarios?	**Seguridad**: amenazas, peligros, fallas, nivel de seguridad.
PI-5	¿Bajo qué normativas se desarrollan sistemas críticos ferroviarios utilizando sistemas de gestión de requerimientos software?	**Normativas:** normativa, estándar, buenas prácticas, ley.

3.4.3.2. Formulación de cadenas de búsqueda

Para generar las cadenas de búsqueda, se definieron en la Tabla 5 las palabras claves y un conjunto de palabras relacionadas a cada una de ellas.

Tabla 5. Palabras clave

Palabras clave	**Palabras relacionadas**
Software	Application, firmware, program
Requirement	Safety-critical software requirement, requisite, SRS
Safety	Hazard, failure, fault, threat, risk, security level, SIL, SSIL, RAMS
Railway	Train, rails, metro, monorail, subway, tracks, 50128, 62279, 61508

Luego de haber definido las palabras claves y sus relaciones, se las unieron usando los conectores lógicos AND y OR, obteniendo la siguiente cadena de búsqueda:

(software OR application OR firmware OR program) AND (requirement OR "safety-critical software requirement" OR requisite OR SRS) AND (safety OR hazard OR failure OR fault OR threat OR risk OR "security level" OR SIL OR SSIL OR RAMS) AND (railway OR train OR rails OR metro OR monorail OR subway OR tracks OR 50128 OR 62279 OR 61508)

Como aclaración: la cadena de búsqueda tuvo que ser refinada para los distintos repositorios, debido a las restricciones que poseen, quedando:

- **IEEE** (15 términos como máximo): (software OR application) AND (requirement OR "safety-critical software requirement" OR requisite OR SRS) AND (safety OR hazard OR fault OR SIL OR RAMS) AND (railway OR 50128 OR 62279 OR 61508)
- **Science Direct** (250 caracteres como máximo): (software OR application OR program OR firmware) AND (requirement OR "safety-critical software requirement" OR requisite) AND (safety OR hazard OR fault OR risk OR SIL OR SSIL OR RAMS) AND (railway OR train OR rails OR 50128 OR 62279 OR 61508)
- **Springer** (computer science and software engineering): (software OR application OR firmware) AND (requirement OR "safety-critical software requirement" OR requisite OR SRS) AND (safety OR hazard OR failure OR fault OR "security level" OR SIL OR SSIL OR RAMS) AND (railway OR subway OR 50128 OR 62279 OR 61508)

3.4.3.3. Selección de fuentes de búsqueda

La búsqueda de información fue realizada sobre las fuentes electrónicas que se mencionan a continuación, usando las siguientes funcionalidades/secciones:

- Science Direct (advanced search - title, abstract, keywords)
- ACM (advanced search)
- Springer
- IEEE (command search - metadata only - Metadata Includes the abstract, index terms, and bibliographic citation data (such as document title, publication title, author, etc.)).

3.4.3.4. Determinación de criterios de selección

Los criterios de inclusión de la información encontrada en la confección de la RSL fueron: artículos relacionados con la gestión de requerimientos software en sistemas críticos ferroviarios, en inglés y publicados en congresos, workshops, revistas, libros y/o capítulos.

El criterio de exclusión definido fue: artículos repetidos en otras fuentes de búsqueda.

3.4.4. Ejecución de la RSL

Para llevar a cabo la presente RSL, por cada fuente de búsqueda definida se llevaron a cabo los siguientes pasos, con tal de obtener los estudios primarios que se analizaron:

1. Realizar la búsqueda utilizando la cadena de búsqueda definida.
2. En los artículos resultantes, decidir cuáles incluir y cuáles excluir leyendo el título y el abstract.
3. De los artículos resultantes del paso anterior, decidir cuáles incluir y cuáles excluir leyendo el texto completo.
4. Los artículos resultantes fueron indicados y clasificados de acuerdo a las dimensiones definidas, colocados en el *Anexo Artículos seleccionados RSL* y numerados con referencias desde [A1] hasta [A23] para mejorar su lectura. En la Tabla 6 se indicó la distribución de los artículos incluidos de acuerdo a sus fuentes.

Tabla 6. Distribución de artículos

Fuente	Artículos encontrados	Filtrados por título	Filtrados por abstract	Filtrados por texto	Porcentaje por fuente
Science Direct	3196	17	8	1	4
ACM	506	9	3	3	13
Springer	5039	38	19	11	48
IEEE	368	20	13	8	35
Total	**9109**	**84**	**43**	**23**	**100%**

Además de los artículos encontrados en la búsqueda definida, se habían analizado 3 más de especial importancia para el estado del arte, por lo que se los incluye en la presente RSL. Los mismos son:

- CENELEC 50128 and IEC 62279 Standards [A24]
- NASA Software Safety Guidebook [A25]
- Integration between Requirements Engineering and Safety Analysis: A Systematic Literature Review [A26]

3.4.5. Reporte de resultados

En la presente sección se muestran los resultados obtenidos del análisis de los estudios primarios obtenidos, teniendo en cuenta las distintas PI a las que responden y se referencian los artículos en los que se encuentran.

3.4.5.1. PI-1 - ¿Qué tipos o módulos de sistemas software se desarrollan utilizando metodologías de gestión de requerimientos software en sistemas críticos ferroviarios?

En la Fig. 11 se muestra la distribución de aplicación de gestión de requerimientos software a distintos sistemas, módulos o subsistemas ferroviarios. Estos se corresponden con los principales sistemas encontrados en la RSL realizada, siendo en los que más se trabaja con estos aspectos.

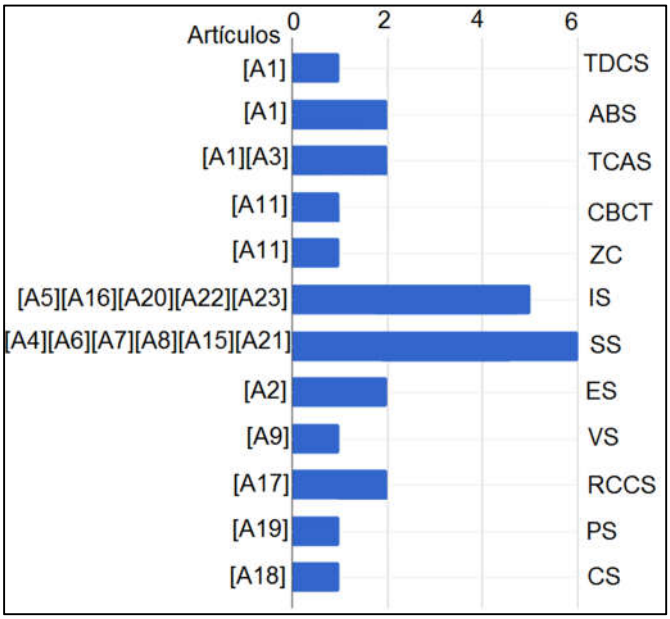


Fig. 11. Distribución por tipo de sistema

Los tipos de sistemas encontrados fueron:

- **TDCS - Train Door Control System**: sistemas de control de las puertas del tren.
- **ABS - Antiblockiersystem**: sistemas anti-bloqueo de frenos, diseñados exclusivamente para que las ruedas del tren no se bloqueen al frenar y causen que el mismo se desestabilice.
- **TCAS - Traffic Collision and Avoidance System**: sistemas de control de tráfico y evasión de colisiones, diseñados para calcular las rutas seguras de los trenes en cada momento del tiempo.
- **CBCT - Communications Based Train Control**: sistemas de control de trenes basados en comunicaciones, diseñados para realizar comunicaciones bidireccionales entre el equipamiento del tren y los componentes instalados en las vías ferroviarias.
- **ZC - Zone Controller**: controlador de zonas, es un sub módulo de los sistemas CBCT.
- **IS - Interlocking System**: sistemas de interbloqueo, diseñados para evitar cruces no deseados entre carriles en los puntos de conjunción entre los mismos, de tal manera de habilitarlos solamente cuando su uso es considerado seguro para la circulación de los trenes.
- **SS - Signalling Systems**: sistemas de señalamiento, diseñados para indicar mediante

distintos tipos de señales (como ser visuales o auditivas) los distintos tipos de eventos que pueden ocurrir durante el tráfico ferroviario.

- **ES - Entire System**: sistema completo, hace referencia a todo el sistema en sí que compone a la actividad ferroviaria. La integración de este tipo de metodologías a nivel de sistema completo es sumamente compleja y costosa en cuestión de recursos.
- **VS - Vigilance System**: sistemas de vigilancia, también conocidos como Dead-man's vigilance devices (dispositivos de vigilancia de hombre muerto), diseñados para la detección y toma de medidas preventivas/correctivas al momento de detección de incapacidad del operario del tren.
- **RCCS - Railroad Crossing Control System**: sistema de control de cruces ferroviarios, diseñados para manejar el cruce de vehículos y peatones por una vía ferroviaria, generalmente por medio de barreras y señales.
- **PS - Platform System**: sistemas de plataforma, diseñados para gestionar la seguridad de las plataformas ferroviarias.
- **CS - Control System**: sistemas de control, engloban distintos subsistemas o módulos de control ferroviarios, como ser control de temperatura, tracción, peso, entre otros.

Mediante la lectura de los artículos seleccionados y el conteo de los sistemas tratados en los mismos, se puede observar la prevalencia de integración de métodos de ingeniería del software principalmente en la definición de sistema de señalamiento (SS) y de interbloqueo (IS). Se hace mayor énfasis en los SS y en los IS porque son sistemas sumamente complejos, integran hardware y software, y generalmente son de tipo SIL 4. Para los primeros se utilizan técnicas de análisis de seguridad, como ser SFMEA, y métodos formales, como ser lenguaje B, lenguaje Z, la suite SCADE, métodos semi formales, como ser diagramas de estado o UML, y lenguajes de programación de bajo nivel, como ser C. Para los segundos se realizan controles de diseño mediante chequeo de modelos, y se hace un gran énfasis también en el uso de métodos formales, de manera similar a los primeros.

3.4.5.2. PI-2 - ¿Qué metodologías se utilizan para la gestión de requerimientos software en el desarrollo de sistemas críticos ferroviarios?

Los principales tipos de metodologías utilizadas fueron: los métodos formales (53.1%), métodos estructurados (34.4%) y métodos mixtos (12.5%). Cabe apreciar que, a pesar de su auge en la actualidad, no se encontraron metodologías ágiles para esto, y prevalecieron los métodos formales por sobre los demás. Esto se puede deber principalmente a la naturaleza intrínsecamente compleja de estos sistemas, que requieren una gran cantidad de técnicas de análisis de seguridad principalmente, siendo más importante este aspecto que los cortos tiempos de entrega (en los artículos encontrados, generalmente se tratan de proyectos que llevan años para su concreción).

De los métodos formales, los más mencionados fueron: métodos integrados, método formal B (lenguaje B), método formal Z (lenguaje Z), entre otros, como se puede observar en la Fig. 12. En cuanto a los primeros, se clasificó en esta categoría a aquellos que se valen del producto software con el que se implementan para definir las especificaciones de los requerimientos software. Se puede apreciar una prevalencia de estos, ya que las herramientas que los soportan ayudan a gestionar la complejidad del uso de dichos métodos, a la vez que brindan otras funcionalidades, como ser la generación automática de diagramas y código, que aportan al análisis de los requerimientos y su documentación de manera automatizada y organizada.

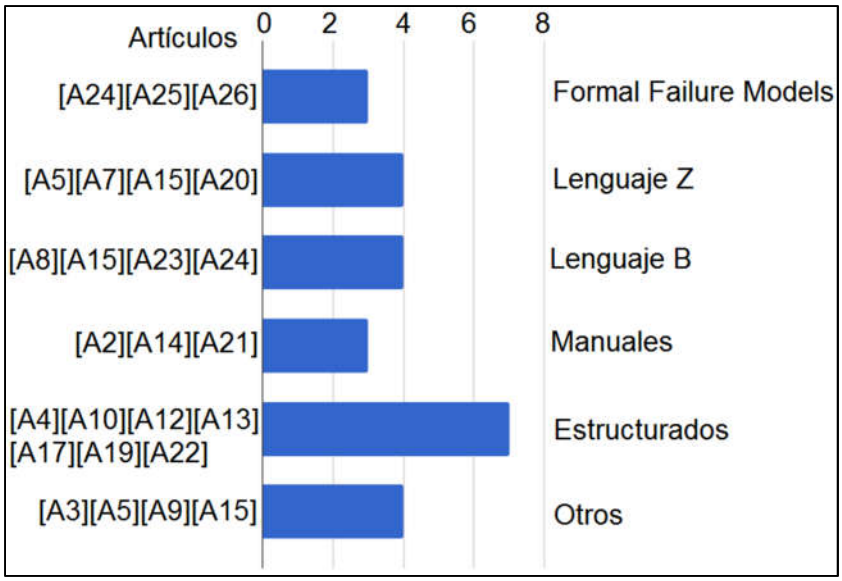


Fig. 12. Métodos formales usados

Los métodos formales encontrados fueron:

- **Formal Failure Models**: modelos de fallas formales, son métodos que realizan análisis de seguridad formales sobre modelos del sistema generados, con tal de hallar posibles fallas y peligros del software.
- **Lenguaje Z**: es un lenguaje formal usado en la ingeniería del software para realizar especificaciones formales del sistema software. Se basa principalmente en teoría de conjuntos, cálculo lambda y lógica de primer orden.
- **Lenguaje B**: o método B, es un método de especificación formal de software basado en notación de máquina abstracta, que brinda herramientas para, además de la especificación formal, las fases de diseño, pruebas y generación de código. En comparación al lenguaje Z, B está más enfocado al refinamiento de código que a la especificación formal (aunque también es muy útil para esto último), y tiene características de un lenguaje de más bajo nivel.

Además, se encontraron los siguientes métodos:

- **Manuales**: hace referencia a métodos manuales de especificación semi formal, especificados mediante modelos y fórmulas matemáticas, con demostraciones formales o semi formales hechas sobre las mismas.
- **Estructurados**: métodos estructurados, llevados a cabo mediante la creación de procedimientos, procesos, actividades y tareas, indicando un orden y una secuencia de gestión.
- **Otros**: combinaciones de las metodologías anteriores, además de otras como ser Redes de Petri, Abstract Interpretation (interpretación abstracta), entre otras.

Cabe destacar que la implementación de métodos formales se dio casi siempre de manera parcial, integrando otras metodologías y herramientas a las mismas, debido a que de esta forma se logra una comprensión más completa del problema, y su uso es altamente recomendado por los distintos estándares a cumplir (como EN-50128). Este tipo de integraciones parciales también se debe a la naturaleza de cada caso de aplicación, que, aunque puedan tratar el mismo dominio de problema, varían en detalles (incluso en la *experticia* de los participantes de los proyectos en las

distintas herramientas, técnicas y metodologías), lo que conlleva naturalmente a realizar análisis de distinta índole.

3.4.5.3. PI-3 - ¿Qué software o herramientas se utilizan y cómo se logra la integración de las mismas en la gestión de requerimientos software en el desarrollo de sistemas críticos ferroviarios?

Los principales tipos de herramientas utilizadas fueron: software (52,6%), modelos (26,3%) e integración de herramientas varias (21,1%). Cabe resaltar que no se encontró ninguna propuesta que haga referencia a métodos manuales de gestión, es decir que se tiende hacia una gestión automatizada por medio de herramientas software especialmente diseñadas para estos fines. Esto se debe principalmente a que por causa de la altísima complejidad que presentan estos sistemas, a la hora de trabajar sobre procedimientos de gestión de requerimientos, el uso de herramientas manuales no automatizadas se vuelve inviable muy rápidamente: se genera una gran cantidad de documentación de manera muy veloz, con altos niveles de complejidad, y que va cambiando (evolucionando) a medida que se realizan más y más análisis.

Una de las herramientas software más utilizadas para el uso de métodos formales es SCADE Suite (usada en cuatro trabajos), la cual brinda un entorno de desarrollo basado en modelos para sistemas críticos embebidos de software, e incluye un lenguaje formal propio, integrando todas las etapas del ciclo de vida de un sistema de dicha índole.

Otra herramienta ampliamente usada es Matlab (usada en dos trabajos) con sus herramientas Simulink y Stateflow. Matlab es una herramienta software que permite el desarrollo de cálculos matemáticos mediante su lenguaje propio (lenguaje M). Simulink es una herramienta componente del paquete Matlab que permite la creación de modelos y simulaciones sobre los mismos, y es utilizada para realizar chequeos de dichos modelos (model checking). Stateflow es otra herramienta de Matlab para modelar y simular lógica de decisión combinatoria y secuencial basado en máquinas de estado y diagramas de flujo, permitiendo realizar análisis semi formales. Cabe aclarar que para el modelado semi formal, las herramientas más utilizadas fueron aquellas que permiten realizar diagramas UML, como ser diagramas de secuencia, de estados, de clases, entre otras.

Otra herramienta utilizada es Frama-C (usada en un trabajo), la cual brinda un framework que permite realizar análisis estático de código fuente escrito en lenguaje C, además de realizar especificaciones formales y validaciones sobre el mismo, entre otras funcionalidades. Para realizar dichas especificaciones formales, la herramienta se basa en el lenguaje ACSL (ANSI/ISO-C Specification Language).

Cabe destacar además el uso de herramientas software propias, confeccionadas de manera personalizada para cada proyecto puntual; la ventaja de las mismas sobre las otras, es la integración con las metodologías y técnicas utilizadas en cada caso puntual.

3.4.5.4. PI-4 - ¿Cómo se gestiona en los sistemas de gestión de requerimientos software la seguridad de los sistemas críticos ferroviarios?

Las técnicas de seguridad aplicadas a los requerimientos software para gestionar la seguridad de sistemas críticos ferroviarios encontradas fueron:

- **SFTA - Software Failure Tree Analysis**: análisis del árbol de fallas del software, es un análisis deductivo de arriba hacia abajo (de manera descendente), en el que se descubren los posibles eventos desencadenantes de una falla del software mediante el uso de lógica booleana y símbolos básicos. Su complemento es denominado SSTA (Software Success Tree Analysis, o análisis del árbol de éxitos del software), y permite conocer los eventos necesarios para que el software realice la función para la que está diseñado.
- **SFMEA - Software Failure Modes and Effect Analysis**: análisis de modos de fallas y efectos del software, es un análisis inductivo de abajo hacia arriba (ascendente), en el que se analizan los efectos de una falla del software. Esta técnica se suele utilizar como complemento de SFTA.
- **PHA - Preliminary Hazard Analysis**: análisis de peligros preliminares, es una técnica de análisis de peligros que se lleva a cabo a nivel de sistema en general (no únicamente del sub-sistema software). Generalmente se lleva a cabo mediante el control de listas pre-armadas de posibles peligros para distintos dominios de aplicación, dependiendo del propio del proyecto a tratar.

- **HAZOP - Hazard and Operability study**: estudio de peligros y operabilidad, es una técnica de identificación de riesgos inductiva, que intenta descubrir y analizar dichos riesgos mediante las posibles desviaciones operativas (o del funcionamiento) de un sistema.
- **SFMECA - Software Effect and Criticality Analysis**: análisis de criticidad, efectos y modos de fallas del software, es una técnica de análisis de abajo hacia arriba (inductiva), que extiende a SFMEA agregándole un análisis de criticidad a la misma. Es decir, luego de realizar un análisis SFMEA, se determina para cada uno de los modos de fallos encontrados, la criticidad de los mismos, analizando además otras cuestiones, como ser sus efectos, causas, consecuencias, modos de prevención y corrección, entre más atributos de las mismas.
- **STPA - Systems-Theoretic Processes Analysis**: análisis de procesos teóricos de sistemas, es una técnica de análisis de seguridad basada en un modelo y proceso de accidentes teóricos de sistemas (STAMP, Systems-Theoretic Accident Model and Process (STAMP)) de sistemas grandes y complejos de seguridad crítica, que permite identificar los factores causales de posibles peligros. Es una técnica que se aplica a sistemas en general, no solamente a sub-sistemas software.
- **CEA - Critical Event Analysis**: análisis de eventos críticos, es un análisis deductivo, similar al FTA, utilizado para identificar las causas de ocurrencia de eventos críticos a nivel de sistemas.
- **SSQI - Safety Specification Quality Index**: índice de calidad de especificaciones de seguridad, es un análisis que brinda un índice que indica la cantidad de requerimientos de seguridad de una especificación, para estimar el grado de seguridad que se le está brindando a un sistema.
- **SSHA - Software Subsystem Hazard Analysis**: análisis de peligros del sub-sistema software, es un análisis que identifica y lista los peligros relacionados a componentes software, de entre los obtenidos en el análisis de peligros a nivel de sistema (PHA). El sistema y las especificaciones software son revisadas luego para verificar que las funciones software incluidas en la lista de peligros, estén incluidas como requerimientos

de seguridad crítica.

En la Fig. 13 se puede observar la cantidad de veces que se menciona cada una de ellas en los artículos analizados.

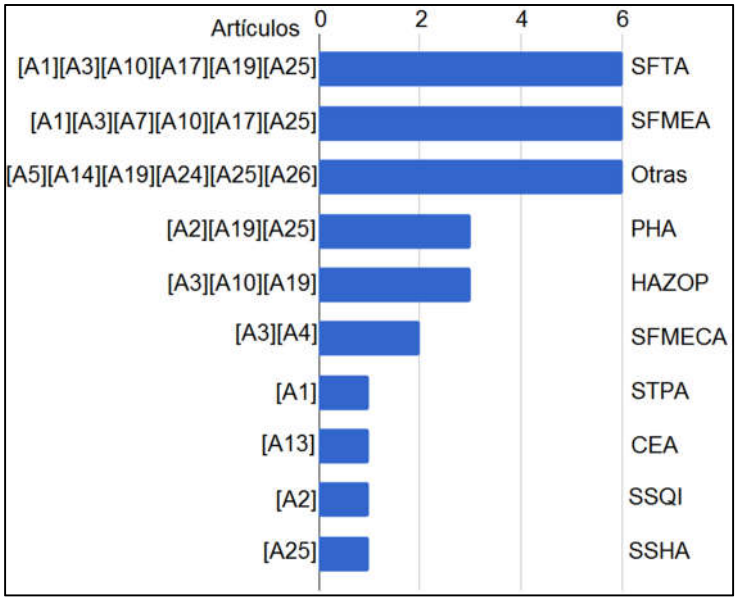


Fig. 13. Técnicas usadas

Se puede apreciar puntualmente el predominio de las técnicas SFTA y SFMEA, debido a que las mismas son técnicas complementarias: las primeras permitiendo descubrir las posibles causas de las fallas del software, y la segunda analizar los efectos, causas, detección, mitigación, prevención y demás características de dichas fallas.

Otras técnicas de análisis son utilizadas a nivel de sistema completo, pero pueden derivar en nuevos requerimientos del sub-sistema software, como ser PHA o HAZOP.

Mediante el análisis de los artículos resultantes de la RSL se halló que las características de seguridad de un sistema crítico complejo (como los ferroviarios) se deben empezar a definir desde las primeras etapas del ciclo de vida del mismo. Actualmente muchas funciones de dichos sistemas son delegadas al sub-sistema software, por lo que este se vuelve un factor crítico en cuanto a la seguridad del sistema en general. Es por esto que es necesaria la realización de estos análisis de seguridad del software a partir de sus primeras fases de vida: los requerimientos.

De esta manera se logra securizar los requerimientos e identificar requerimientos de seguridad del software, pudiendo identificarse posibles modos de fallo del mismo, modos de funcionamiento excepcional, consecuencias e impacto en la seguridad con la implementación de nuevas funcionalidades, entre otras características, desde antes de su integración al sistema en sí.

3.4.5.5. PI-5 - ¿Bajo qué normativas se desarrollan sistemas críticos ferroviarios utilizando sistemas de gestión de requerimientos software?

Las normativas encontradas en los artículos seleccionados fueron:

- **EN-50128**: titulada "Aplicaciones ferroviarias. Sistemas de comunicación, señalización y procesamiento. Software de para sistemas de control y protección del ferrocarril.", es una norma europea elaborada por el CENELEC (en francés *Comité Européen de Normalisation Electrotechnique*, Comité Europeo de Normalización Electrotécnica). Esta norma europea especifica los procedimientos y requisitos técnicos para el desarrollo de software para sistemas electrónicos programables para su uso en aplicaciones de control y protección del ferrocarril. Se puede aplicar en cualquier área del ferrocarril que tenga relación con la seguridad.
- **EN-50657**: titulada "Aplicaciones ferroviarias. Aplicaciones del material rodante. Software a bordo del material rodante.", es una norma europea también elaborada por el CENELEC. Esta norma europea especifica el proceso y los requerimientos técnicos para el desarrollo de software para sistemas electrónicos programables para el uso en aplicaciones de material rodante. Es aplicable a software relacionado a la seguridad como también al no relacionado a la seguridad. Se considera que si el software cumple con una versión válida de la EN-50128, también cumple con esta norma.
- **IEC-61508**: titulada "Seguridad funcional de sistemas electrónicos relacionados a la seguridad eléctricos/electrónicos y programables", es una norma internacional elaborada por el IEC (International Electrotechnical Commission, o Comisión Electrotécnica Internacional). Esta norma consiste en métodos de cómo aplicar, diseñar, desplegar y mantener sistemas de protección automática, denominados sistemas relacionados a la seguridad.

- **IEC-62279**: titulada "Aplicaciones ferroviarias. Sistemas de señalamiento, comunicación y procesamiento. Software para control ferroviario y sistemas de protección", es una norma internacional también elaborada por el IEC. Esta norma especifica los procesos y requerimientos técnicos para el desarrollo de software para sistemas electrónicos programables para el uso en aplicaciones de control y protección ferroviario. El objetivo es que sea usada en cualquier área en donde haya implicaciones de seguridad.

En la Fig. 14 se pueden observar las normas mencionadas.

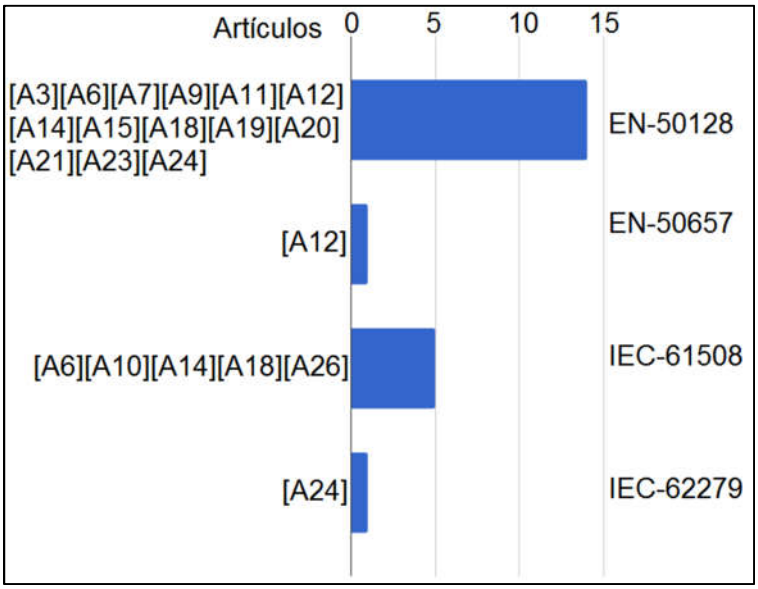


Fig. 14. Normativas para el desarrollo de software crítico ferroviario

Como se puede observar en la Fig. 14, se destaca la prevalencia de artículos mencionando a la normativa EN-50128, por sobre otras más genéricas, como ser la IEC-61508, debido a que la RSL se realizó enfocándose exclusivamente en sistemas ferroviarios, acotando el dominio del problema.

Los artículos analizados mencionan principalmente a la norma EN-50128 a modo de referencia, usándola como guía para el desarrollo de los distintos proyectos, de acuerdo a las metodologías, técnicas y procedimientos propuestos por la misma. En otras ocasiones, los desarrollos se restringen a lo mencionado en dicha norma, con tal de cumplir con la misma, aunque los casos en que sucede esto último son menos.

Es importante destacar el alto grado de influencia que ha tenido la norma al momento del desarrollo de aplicaciones para este dominio del problema, encontrándose referenciada en la gran mayoría de los artículos.

3.4.5.6. Resultados adicionales

En la Fig. 15 se puede observar la evolución cronológica de las publicaciones encontradas, mostrándose la cantidad de publicaciones realizadas por año.

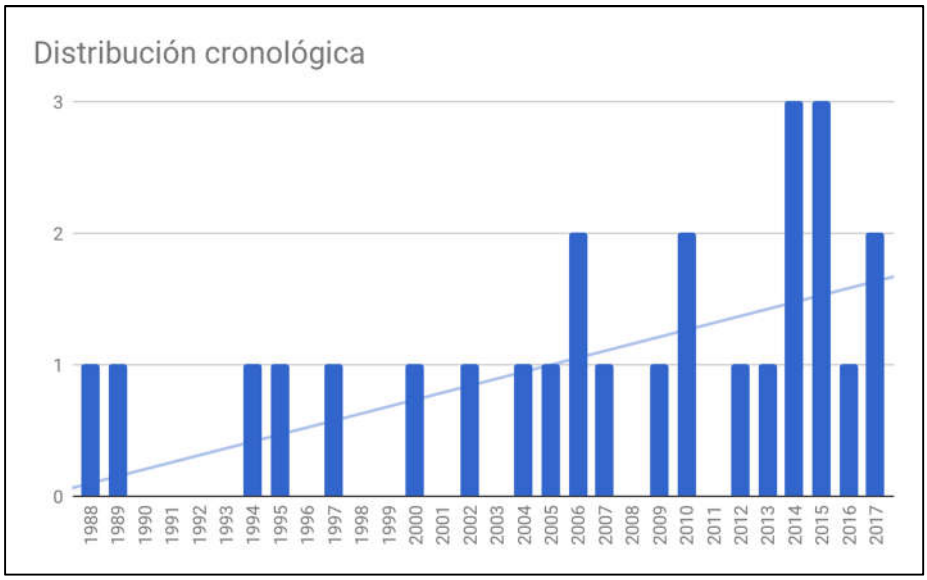


Fig. 15. Distribución por año de publicación

Además del aumento en la cantidad de artículos publicados para este dominio del problema, se pueden apreciar claramente el aumento en la especificidad de los mismos: entre los artículos más antiguos se tratan temas más conceptuales, generales y abarcativos, que hacen mención principalmente a aspectos de la ingeniería del software para sistemas críticos ferroviarios; pero conforme van avanzando los años, los artículos se vuelven más técnicos y se tocan más temas relativos a cuestiones tecnológicas, metodologías, técnicas, frameworks, modelos de análisis, entre otros, a medida que se van puntualizando y disgregando los objetos de estudio: ya no se estudia solamente al sistema ferroviario como un todo, sino también a sus partes (sub-sistemas, módulos y demás) que lo componen. A su vez, las normativas que se intentan cumplir o que se utilizan como marcos de referencia en los distintos trabajos, van avanzando en versiones, volviéndose cada vez más complejas, completas y agrupándose con otras normativas, a medida que evolucionan con el cuerpo de conocimiento de la época correspondiente.

3.4.6. Conclusiones

Se han encontrado un total de 23 publicaciones de carácter relevante sobre esta temática (más las otras 3 mencionadas), las cuales han sido clasificadas y analizadas, encontrando como resultados principales los siguientes:

- El software de sistemas críticos ferroviarios que más se investiga con respecto a sus requerimientos son los de señalamiento e interbloqueo.
- Los métodos más utilizados para la especificación de requerimientos software para sistemas críticos ferroviarios suelen ser métodos formales, con el uso del lenguaje B o Z.
- Muchas publicaciones mencionan al menos una herramienta software para la gestión de este tipo de procedimientos (como SCADE), o la integración de varias con distintas metodologías.
- Las principales técnicas de aseguramiento de la seguridad de los requerimientos software encontradas fueron SFTA y SFMEA, junto con otras que integran el análisis de seguridad del sistema entero
- La mayoría del software crítico ferroviario se desarrolla bajo la norma EN-50128.
- En los últimos siete años se ha incrementado significativamente el estudio de este tipo de problemáticas.

Capítulo 4

Resultados y Verificación

4. Resultados y verificación

4.1. Resultados

En este apartado se describen los resultados del desarrollo de las actividades mencionadas en el capítulo 2. Metodología.

4.1.1 Análisis de la problemática en cuestión y del contexto

Para llevar a cabo el análisis del dominio del problema, se inició la lectura, análisis y comprensión de las normas EN-50128, EN-50126 e ISO 9001, teniendo en cuenta las directivas del libro "CENELEC 50128 and IEC 62279 Standards", el cual brindó el apoyo necesario para comprenderlas. Una vez comprendidas las normativas con las que se estaba desarrollando el proyecto, el siguiente paso fue analizar el contexto dentro del que se encontraba sumido el mismo, es decir, el Sistema de Gestión de Calidad armado por el grupo de trabajo [8]. Para esto se analizaron los siguientes documentos creados dentro del contexto del proyecto:

- **MC_MGC_10 de Gestión de Calidad**: Este manual describe el Sistema de Gestión de la Calidad, perfila los campos de autoridad, las relaciones y los deberes del personal responsable del desempeño de la organización.
  El manual está dividido en 10 secciones que están directamente relacionadas con los requisitos de la norma ISO 9001:2015. Cada sección comienza con una declaración que expresa el deber de la organización de implementar y satisfacer los requisitos básicos de la norma a la que se hace referencia. Después de cada declaración se aporta información específica acerca de los procedimientos que describen los métodos usados para implementar los requerimientos pertinentes. Este manual se utiliza internamente para orientar a los participantes de la organización con respecto a los diversos requisitos de la norma ISO 9001:2015 que deben ser cumplidos y mantenidos para asegurar la satisfacción de los stakeholders, la mejora continua y brindar las directivas necesarias que generen una fuerza laboral dotada de poder, autoridad y responsabilidad en el desarrollo de aplicaciones ferroviarias de calidad.
  Finalmente, la organización desarrolló, implementó y formalizó el Sistema de Gestión de

la Calidad con el fin de:

- Satisfacer los requisitos de la norma internacional ISO 9001:2015
- Documentar las mejores prácticas de negocio de la empresa.
- Entender y satisfacer más adecuadamente las necesidades y las expectativas de sus clientes.
- Desarrollar un marco general para el cumplimiento de normativas ferroviarias.

- **PG_DCP_10 Definición Conceptual del Proyecto**: Este documento tiene como objeto establecer un procedimiento general para realizar la definición conceptual en cada proyecto que se realice bajo el presente sistema de Gestión de la Calidad.
- **PG_DSA_10 Definición del sistema y condiciones de aplicación**: Este documento tiene como objeto establecer un procedimiento general para realizar la definición del sistema y sus condiciones de aplicación en cada proyecto que se realice bajo el presente sistema de Gestión de la Calidad.
- **PG_ARI_10 Análisis de Riesgos**: Este documento tiene como objeto establecer un procedimiento general para realizar el análisis de riesgos que corresponde a cada proyecto que se realice bajo el presente sistema de Gestión de la Calidad.
- **PG_REQ_10 Requisitos del sistema**: Este documento tiene como objeto establecer un procedimiento general para especificar los requisitos globales RAMS correspondientes al sistema total en cada proyecto que se realice bajo el presente sistema de Gestión de la Calidad.
- **PG_PSG_10 Plan de Seguridad**: Este documento tiene como objeto establecer un procedimiento para la confección de un Plan de Seguridad, consistente en un conjunto de actividades programadas temporalmente, recursos y supuestos que sirven para poner en práctica la estructura organizativa, mejorar la eficiencia operativa y la relación con las autoridades competentes y con la población en general.
- **PG_CSE_10 Caso de Seguridad**: El caso de seguridad es un documento estructurado para demostrar o evidenciar la seguridad de un producto genérico, una aplicación genérica o específica.

Un Caso de Seguridad es necesario cuando un peligro no puede ser corregido de inmediato y un análisis deberá demostrar si el riesgo es aceptable. En caso contrario, se demostrará que con una solución alternativa se mantendrá el nivel de seguridad operacional dentro de los límites aceptables y no afectará excesivamente la capacidad operacional del sistema o producto.

En base a las consideraciones previas, este documento tiene como objeto establecer un procedimiento para la confección de un Caso de Seguridad, abarcando los siguientes aspectos:

- Evidencia de la gestión de la calidad.
- Evidencia de la gestión de la seguridad.
- Evidencia de la seguridad técnica y funcional.

- **PG_RAM_10 Políticas RAMS**: Este documento tiene como objeto establecer un procedimiento general para establecer las políticas RAMS del sistema evaluando los parámetros de confiabilidad (fiabilidad), disponibilidad, mantenibilidad y seguridad de los distintos equipos que forman parte del sistema. Con esto se espera optimizar el rendimiento, minimizar la pérdida debida a fallas y requerimientos de mantenimiento e inspección, e identificar los equipos más críticos para el funcionamiento óptimo.
- **PG_DRQ_10 Distribución de los requisitos del sistema**: Este documento tiene como objeto establecer un procedimiento general para realizar la Distribución de los requisitos del sistema en cada proyecto que se realice bajo el presente sistema de Gestión de la Calidad.
- **PG_DEI_10 Diseño e Implementación**: Este documento tiene como objeto establecer un procedimiento general para realizar el diseño y la implementación del sistema, según los siguientes objetivos:
    - Crear los subsistemas e identificar los componentes que se ajusten a los requisitos RAMS.
    - Demostrar que los componentes y subsistemas se ajustan a los requisitos RAMS.
    - Establecer planes para las futuras fases de desarrollo.

- **PG_FAB_10 Fabricación**: Este documento tiene como objeto verificar y poner en práctica la fabricación de los subsistemas y componentes, estableciendo planes de formación de recursos humanos, con ajuste a los requisitos RAMS establecidos en la norma EN-50126 / EN-50128 y EN-50129.
- **PG_INS_10 Instalación**: Este documento tiene como objeto verificar y poner en práctica el montaje y la instalación de los subsistemas y componentes que se requieren para formar el sistema completo. Poner en marcha los planes de apoyo al sistema, con ajuste a los requisitos RAMS establecidos en la norma EN-50126 / EN-50128 y EN-50129.

La presencia del profesor orientador de este trabajo y de los demás miembros del grupo de investigación fue fundamental para la comprensión de muchos de estos conceptos, realizándose reuniones con cierta periodicidad al principio del proyecto para ayudar a transmitir estos conocimientos.

Además de esto, se realizaron búsquedas y análisis de artículos relativos a estos temas para lograr indagar más acerca del estado de la cuestión, encontrándose principalmente los siguientes:

- *Integration between Requirements Engineering and Safety Analysis: A Systematic Literature Review* [16]: en esta RSL se evidencia la integración entre ingeniería de requerimientos y de seguridad y los beneficios que esto conlleva.
- *NASA Software Safety Guidebook* [17]: es una guía desarrollada por la NASA que da a conocer los métodos que utilizan para dotar de seguridad al software en sus proyectos.

4.1.2 Creación y descripción del procedimiento de gestión de requerimientos software

Luego de haber tenido una primera aproximación a la problemática a resolver, se creó la primera versión del Procedimiento General de Gestión de Requisitos Software (PG_RS_01 Requisitos del Software). El mismo se adaptó al formato y lógica del resto de documentación ya generada dentro del proyecto de gestión de calidad del GICSAFe, indicándose objetivos, alcance, referencias y responsabilidades, su cuerpo principal y las entregas a generar, lográndose así una integración completa con los demás procedimientos que se encuentran dentro del proyecto mencionado. El uso correcto de este procedimiento generaría el documento de Especificación de Requisitos Software, garantizando el cumplimiento del punto 7.2.1.1 de la norma EN-50128.

Se definieron las pautas generales del procedimiento, sub-dividiéndolo a su vez en 3 distintos procesos: Obtención, Especificación y Verificación, como se puede ver en la Fig. 16.

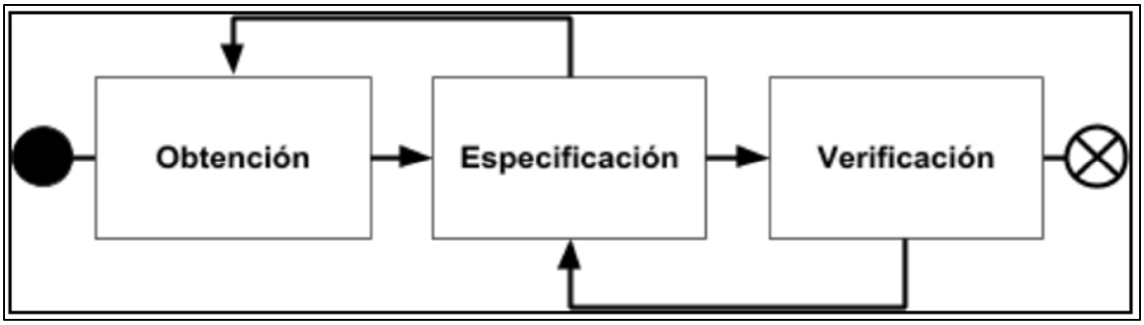


Fig. 16. Procesos del procedimiento

Cada uno de estos procesos tiene como objetivo:

- **Obtención**: obtener, documentar, identificar, clarificar y justificar los requerimientos software.
- **Especificación**: analizar los requerimientos software obtenidos con el proceso anterior, definirlos y especificar los cambios que estos provocarán en el sistema en caso de implementarse.
- **Verificación**: este proceso está compuesto a su vez por 2 sub-procesos: el de verificación y el de validación, cuyos objetivos son:
    - *Verificación:* llevar a cabo los distintos tipos de controles y pruebas técnicas a llevarse a cabo sobre los requerimientos especificados, para mantener el nivel de seguridad requerido del software.
    - *Validación:* validar las especificaciones de requerimientos generadas en el proceso anterior con los stakeholders, modificarlos y redefinirlos en un proceso iterativo de ser necesario, definir los ensayos a realizarse sobre los mismos

Como se puede apreciar más adelante en este documento, en el capítulo 4.1.5 Agregado de técnicas de seguridad al procedimiento, se cambió el nombre de este último proceso por el de "Verificación y validación". El nombre "Verificación" que se le puso en un principio se debió a un error conceptual, que fue corregido en instancias posteriores del procedimiento, como se indica en el capítulo mencionado.

A su vez, para cada uno de los procesos se definieron, mediante una tabla de información específica del proceso: su responsable, su objetivo, los resultados esperados de haberlo concretado, su alcance, los errores que se pretenden evitar y el marco normativo bajo el que se diseña.

Cada proceso se compone a su vez de actividades a llevar a cabo para su concreción, y en la misma tabla, para cada actividad se indican: un orden, la descripción de la actividad en sí, su responsable y los registros que se deberán generar al realizarse.[2] Además, para algunos de estos registros mencionados, se definieron códigos identificadores, para poder ser rastreados a lo largo del ciclo de vida del proyecto, siguiendo una determinada nomenclatura, indicada en la información de la actividad.

Para el proceso de **Obtención**, las actividades a realizar para cumplimentarlo se pueden observar en la Fig. 17.

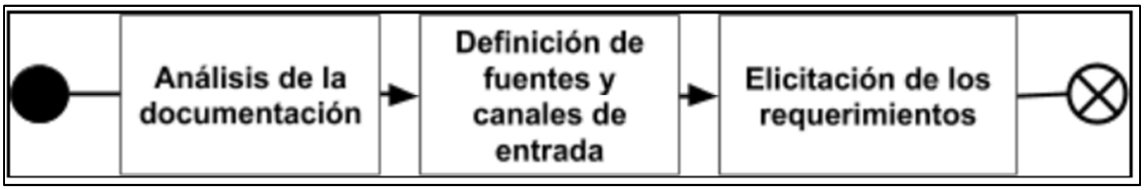


Fig. 17. Proceso de obtención

En la

Tabla 7 se puede ver la información específica del proceso en esta primer etapa.

La actividad 1 tiene como objetivo realizar el análisis sistemático de la documentación de entrada relacionada al requisito software que se esté por tratar, indicada en el capítulo 7.2.2 de la norma EN-50128, indicando el documento que se analizó, la fecha de análisis, su completitud y correctitud. Esta información tiene que haber sido analizada, satisfaciendo los niveles de completitud y correctitud requeridos en cada caso, y generando una tabla como registro de salida con dicha información, como se puede observar a continuación a modo de ejemplo en la Tabla 8.

[2] Los números de tablas, figuras y capítulos expresados en los campos de registros de las actividades, corresponden a los que componen al procedimiento en sí, no a los de este documento. Dichos objetos se pueden encontrar en los Anexos de las distintas versiones del procedimiento.

Tabla 7. Información específica del Proceso de Obtención

Nombre del proceso: Obtención[3]

Responsable: Gestor de Requerimientos.

Objetivo: Obtener los requerimientos del software del sistema.[4]

Resultados esperados: Requerimientos del sistema.

Alcance: El Sistema de Gestión de Requerimientos.

Errores a evitar: Fuentes de entradas de requerimientos no identificadas, canales de comunicación no definidos y requerimientos mal obtenidos.

Marco normativo: Norma UNE-EN 50128.

Orden	Actividad	Responsable de ejecución	Registro
1	Realizar un análisis sistemático de la documentación de entrada.	Gestor de Requerimientos	Filas 2, 3, 4, 5, 6 y 7 de la Tabla 2
2	Identificar los stakeholders y canales de entrada de los requerimientos. Los stakeholders serán identificados mediante un valor con el formato SH-XXXXX. Los grupos de stakeholders serán identificados mediante un valor con el formato GSH-XXX.	Gestor de Requerimientos	Filas 2 de la Tabla 3 y 5
3	Definir el formato y las técnicas de elicitación a utilizar, al igual que la planificación.	Gestor de Requerimientos	Fila 2 de la Tabla 7
4	Realizar la elicitación de los requerimientos. Los requerimientos serán identificados mediante un valor con el formato RQ-XXXXX.	Gestor de Requerimientos	Columna 2 de la Tabla 9

La actividad 2 tiene como objetivo identificar los stakeholders (individuales o en grupo) y canales de entrada de los requerimientos, dejándolos registrados de manera explícita, pudiendo así obtenerse un historial y trazabilidad correctos de los inicios de cada requerimiento software. Esta identificación se realizó mediante la generación de dos tablas; la primera para la identificación individual de los stakeholders, con un formato que se puede observar a

[3] El campo "Nombre del proceso" no se encuentra originalmente en el procedimiento. Se lo agregó solamente dentro de este documento, en esta tabla de información específica del proceso y en todas las demás.
[4] El campo "Objetivo" fue modificado en este y los demás procesos dentro de este documento, por lo que puede diferir del procedimiento.

continuación a modo de ejemplo en la Tabla 9, y la segunda para la identificación de grupos de stakeholders, como se puede ver en la Tabla 10.

Tabla 8. Análisis de documentación de entrada

Documento	Fecha de análisis	Completitud	Correctitud
Especificación de Requisitos del Sistema	10/07/2017	Completa	Correcta
Especificación de Requisitos de Seguridad del Sistema	11/07/2017	Completa	Correcta
Descripción de la Arquitectura del sistema	13/07/2017	Completa	Correcta
Especificaciones de la Interfaz Externa	13/07/2017	Completa	Correcta
Plan de Aseguramiento de Calidad del Software	13/07/2017	Completa	Correcta
Plan de Validación del Software	15/07/2017	Completa	Correcta
Observaciones	La documentación de entrada es la adecuada para el desarrollo del requerimiento.		

Tabla 9. Stakeholders

Identificador	Nombre y apellido	Puesto o función	Canales de comunicación
SH-00001	Mariano Moreno	Maquinista del tren	Email: marianomoreno@ejemplo.com.ar Dirección: Ciudad, Calle, Altura Teléfono: 555- 4444
SH-00002	Roberto Perez	Operador de sistemas	Email: robertoperez@ejemplo.com.ar Teléfono: 555- 4443

Tabla 10. Grupos de stakeholders

Identificador de grupo	Nombre de grupo	Descripción de grupo	Miembros del grupo
GSH-001	Operarios de trenes	Personas vinculadas a la operación física y lógica de los trenes	SH-00001 SH-00002

La actividad 3 tiene como objetivo definir el formato y las técnicas de elicitación a utilizar (entrevistas, encuestas, cuestionarios, formularios, entre otros), al igual que la planificación. Es decir, como se realizará la elicitación de requerimientos, indicándose el medio (se pueden incluir

domicilios por ejemplo), la duración aproximada de la actividad y los objetivos de la misma, como se ve en la Tabla 11 a modo de ejemplo.

Tabla 11. Técnicas de elicitación

Técnica	Duración aproximada	Objetivos
Entrevista	3 horas aprox.	Obtener información sensible y puntual sobre un tema en específico, difícil de hallar mediante comunicación no presencial.
Cuestionario	1 hora aprox.	Obtener información general sobre temas específicos.

Por último, la actividad 4 tiene como objetivo realizar la elicitación de los requerimientos en sí, dejando plasmado el mismo de manera explícita, y dándole una identidad propia dentro del proyecto. Será necesario, por cada uno, indicar su fecha de elicitación, la fuente, la necesidad que se desea satisfacer con su concreción, el motivo que lo origina, el objetivo de la realización del requerimiento, y la verificación que se debe llevar a cabo para comprobar que el mismo ha sido cumplido de manera correcta. Esta actividad cumple con lo establecido en el punto 7.2.4.4 de la norma EN-50128, en donde se indica que la especificación de requisitos de software debe expresarse y estructurarse de forma que sea completa, clara, precisa, inequívoca, verificable, que se pueda someter a ensayo, que se pueda mantener y sea realizable, además de que cada requisito debe ser trazable hasta los documentos de entrada (esto último se cumple utilizando el identificador único definido para cada requisito software). El cumplimiento de estas verificaciones estará a cargo del responsable de la actividad.

Esta información deberá ser registrada en una tabla, como la que se indica a modo de ejemplo en la Tabla 12.

Tabla 12. Elicitación de los requerimientos

RQ-ID	RQ-00001	Fecha	25/03/2017	Fuente	GSH-00001
Necesidad	El monitor de barrera debe informar su estado cada 5 segundos				
Motivo	Necesidad de contar con información actualizada del dispositivo				
Objetivo	Reducir las probabilidades de un accidente				
Verificación	El monitor de barreras informa su estado cada 5 segundos				

Para el proceso de **Especificación**, las actividades a realizar para cumplimentarlo se pueden observar en la Fig. 18.

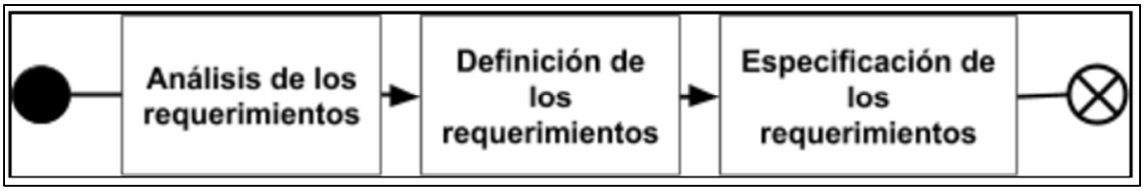


Fig. 18. Proceso de especificación

Este proceso posee restricciones que deben ser respetadas para su cumplimiento, las mismas son:

- *7.2.2 El nivel de integridad de seguridad del software debe registrarse en la Especificación de Requisitos del Software*. Este nivel de integridad (o SIL) debe ser deducido siguiendo una lógica preestablecida, indicada en el capítulo 4 de la norma EN-50128. Es necesario establecer dicho valor ya que de acuerdo al mismo la norma aplica ciertos controles al software a desarrollarse.
- *7.2.3 Los requerimientos de tipo no funcional pueden tener incluidos, entre otros, los subtipos: robustez, mantenibilidad, seguridad, eficiencia, usabilidad y portabilidad*. De acuerdo al subtipo que posean, se deberá hacer hincapié en dicho atributo del software a desarrollar, pero sin descuidar los demás aspectos relacionados al subtipo. Esto así se indica en el capítulo 7.2.4.2 de la norma EN-50128. Este análisis y especificación pueden realizarse a la hora de hacer el análisis sistémico del requerimiento.
- *7.2.4 El otro tipo de requerimientos, además de los no funcionales, son los funcionales, los cuales hacen referencia a cualidades o funcionalidades del software*. Relacionado al punto anterior, para estos requerimientos también se deben especificar las cualidades mencionadas anteriormente. Este análisis y especificación pueden realizarse a la hora de hacer el análisis sistémico del requerimiento.
- *7.2.5 Se deben explicitar los requisitos utilizados para los ensayos periódicos de funciones, indicando que es un requisito de tipo no funcional y de subtipo ensayo*. Esta restricción ayuda a cumplir parcialmente los puntos 7.2.4.11 y 7.2.4.12 de la norma EN-50128, en la que se indica que en el documento de Especificación de Requisitos Software

se deben incluir los ensayos o pruebas que deben ser llevados periódicamente a cabo para las funciones comunes y de seguridad del software, dentro de lo establecido en el documento global de Especificación de Requisitos de Seguridad del Sistema.

- *7.2.6 Los distintos estados por los que puede pasar el software son modelados mediante un modelo de máquina de estados finitos*. Esto es, realizar un diagrama de transición de estados, partiendo de los modos de comportamiento que se definen inicialmente.
- *7.2.7. Se deben explicitar los requisitos relacionados a la autocomprobación del hardware y del software, especificando la detección y el envío de informes de fallos y errores por parte de los mismos. Los mismos serán definidos como requisitos de tipo funcional y subtipo autocomprobación*. De acuerdo al punto 7.2.4.10, el software debe informar de alguna manera de sus propios fallos y del fallo del hardware que tiene a su cargo.

En la Tabla 13 se puede ver la información específica del proceso en esta primer etapa.

Tabla 13. Información específica del Proceso de Especificación

Nombre del proceso: Especificación

Responsable: Gestor de Requerimientos.

Objetivo: Analizar, definir y especificar los requerimientos obtenidos en el proceso de Obtención.

Resultados esperados: Especificación de requisitos del software.

Alcance: El Sistema de Gestión de Requerimientos.

Errores a evitar: Requerimientos contradictorios, mal interpretados o mal definidos.

Marco normativo: Norma UNE-EN 50128.

Orden	Actividad	Responsable de ejecución	Registro
1	Realizar un análisis operacional de los requerimientos.	Gestor de Requerimientos	Fila 2 y 3 de la Tabla 12
2	Realizar un análisis sistémico de los requerimientos.	Gestor de Requerimientos	Fila 4 y 5 de la Tabla 12
3	Definir los distintos modos de comportamiento del software.	Gestor de Requerimientos	Tabla 14
4	Definir atributos característicos de los requerimientos	Gestor de Requerimientos	Fila 2 de la Tabla 16
5	Construir un glosario de términos	Gestor de Requerimientos	Tabla 18
6	En caso de generarse nuevos requerimientos a partir de la especificación de los actuales,	Gestor de Requerimientos	7.1 Proceso de Obtención

	educirlos.		

La actividad 1 tiene como objetivo realizar un análisis operacional de los requerimientos, y la 2 realizar un análisis sistémico de los mismos. Es decir, se pretenden analizar e indicar los cambios a nivel funcional (operativo) e informático (en el sistema software en el que se incluiría dicho requerimiento) en caso de llevarse a cabo. Un ejemplo de la información que se produce como registro de salida de dichas actividades se puede observar en la Tabla 14.

Tabla 14. Análisis operacional y sistémico

RQ-ID	RQ-00001			
Análisis operacional	Fecha	24/03/2017	Responsable	Juan Ramirez
Resultados	La implementación del requerimiento permitiría tener información actualizada del estado del monitor de barreras, a un intervalo de tiempo razonable, según [referencia]			
Análisis sistémico	Fecha	25/03/2017	Responsable	Pedro Pujol
Resultados	El requerimiento puede ser implementado mediante la creación de una tabla en la base de datos que registre los estados del monitor de barreras, manteniendo un cierto número de registros del mismo.			

La actividad 3 tiene como objetivo definir los distintos modos de comportamiento del software, es decir, realizar un análisis inicial de los estados iniciales y finales del software en caso de implementarse el requerimiento. Esto se podría pensar como una primera aproximación a un análisis más completo de estados a realizarse posteriormente en el procedimiento. Esto cumple con los puntos 7.2.4.7. y 7.2.4.8 de la norma EN-50128, en donde se establece que estos modos de comportamiento deben estar detallados, particularmente el comportamiento en caso de fallos, aludiendo a la seguridad. Los registros de aplicación de esta actividad se deberán plasmar en una tabla, como la Tabla 15.

Tabla 15. Modos de comportamiento del software

RQ-ID	RQ-00001
Estado inicial	Funcionamiento normal
Evento	Transcurrido lapso de 5 segundos
Estado final	Transmitiendo datos
Resultado	El monitor de barreras ha transmitido sus datos al centro de mandos

La actividad 4 tiene como objetivo definir atributos característicos de los requerimientos, indicando para cada uno ciertos indicadores, como ser:

- *Versión del requerimiento*: ya que el mismo puede evolucionar en versiones en cambio de solicitarse una modificación al mismo.
- *Responsable*: indicando al responsable de gestionar dicho requerimiento
- *Tipo*: indicando el tipo de requerimiento (funcional o no funcional, o la clasificación que corresponda).
- *Subtipo*: indicando el subtipo del tipo anterior (robustez, mantenibilidad, seguridad, eficiencia, usabilidad, portabilidad, entre otros).
- *Estado*: el estado del requerimiento dentro de su ciclo de vida (inicial, especificado, en desarrollo, en testing, entregado, entre otros).
- *Prioridad*: indicando la prioridad que tiene el requerimiento.
- *Esfuerzo*: el esfuerzo estimado para la realización del requerimiento.
- *Impacto en la seguridad*: el nivel de relación que tiene el requerimiento con la seguridad del sistema en general.
- *Restricciones de HW*: restricciones de hardware a la hora de realizar el requerimiento.
- *Restricciones de SW*: restricciones de software a la hora de realizar el requerimiento.

Estos últimos dos atributos cumplen lo indicado en el punto 7.2.4.9 en la norma EN-50128, en donde se especifica que se deben explicitar dichas restricciones HW y SW, en caso de existir.

Un ejemplo de información generada por la realización de esta actividad puede ser visto en la Tabla 16.

Tabla 16. Atributos de los requerimientos

RQ-ID	RQ-00001	Versión	1.0	Responsable	Pedro Pujol
Tipo	No funcional	Subtipo	Seguridad	Estado	Aprobado
Prioridad	8	Esfuerzo	5	Impacto en la seguridad	9
Restricciones de HW		Congestión de canales de comunicación.			
Restricciones de SW		Dificultad para adaptar las tablas involucradas en el proceso.			

La actividad 5 tiene como objetivo construir un glosario de términos, indicando en el mismo las palabras o frases propias del dominio técnico de aplicación, de comprensión necesarias por parte del equipo de trabajo para la resolución del requerimiento. Esto cumple con lo establecido en el punto 7.2.4.5 en la norma EN-50128, en donde se indica que en la Especificación de Requisitos Software se deben incluir modos de expresión y descripciones entendibles por todo el personal involucrado en el ciclo de vida del software.

Un ejemplo de información generada por la realización de esta actividad puede ser visto en la Tabla 17 a modo de ejemplo.

Tabla 17. Glosario de términos

Término	Definición
Stakeholder	Personas u organizaciones que afectan o son afectadas por el proyecto.

Por último, la actividad 6 tiene como objetivo la identificación y educción de nuevos requerimientos, en caso de generarse durante el desarrollo de este proceso, generándolos mediante el proceso de Obtención, de manera iterativa.

Para el proceso de **Verificación**, las actividades a realizar para cumplimentarlo se pueden observar en la Fig. 19.

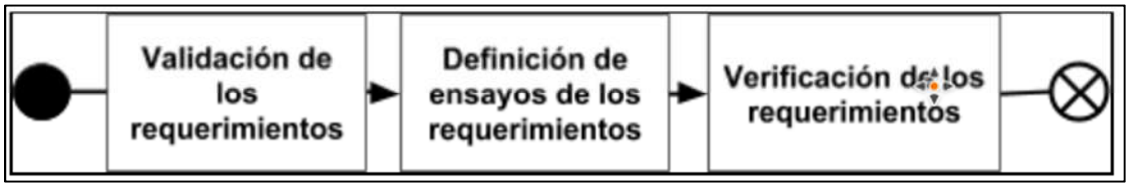


Fig. 19. Proceso de verificación

Este proceso posee una restricción que debe ser respetada para su cumplimiento:

- *7.3.2 Los requerimientos pueden ser verificados con stakeholders individuales o grupos identificados*. Esto es, teniendo en cuenta las personas o grupos de personas.

En la Tabla 18 se puede ver la información específica del proceso en esta primer etapa.

Tabla 18. Información específica del Proceso de Verificación

Nombre del proceso: Verificación **Responsable:** Verificador de Requerimientos. **Objetivo**: Validar los requerimientos especificados con los stakeholders, definir sus modos de comportamiento, los ensayos, controles y pruebas técnicas a realizarse sobre los mismos. **Resultados esperados**: Requerimientos del sistema verificados. Especificación de ensayos del software en conjunto. Informe de verificación de los requisitos del software. **Alcance:** El Sistema de Gestión de Requerimientos. **Errores a evitar:** Requerimientos mal especificados o no verificados. **Marco normativo:** Norma UNE-EN 50128.			
Orden	Actividad	Responsable de ejecución	Registro
1	Verificar con los stakeholders los requerimientos especificados mediante el uso de modelos.	Verificador de Requerimientos	Fila 2 de la Tabla 21
2	Definir los distintos ensayos del software a realizar.	Verificador de Requerimientos, Encargado de Ensayos del Software	Tabla 23
3	Verificar la no contradicción de los requerimientos.	Verificador de Requerimientos	Fila 2 de la Tabla 25
4	Verificar el cumplimiento de los requerimientos con los niveles de seguridad especificados para el software.	Verificador de Requerimientos	Fila 2 de la Tabla 25
5	Verificar los requisitos del software	Verificador de Requerimientos	Tabla 27
6	En caso de generarse modificaciones en los requerimientos durante el proceso, modificar las especificaciones.	Gestor de Requerimientos	7.2 Proceso de Especificación

La actividad 1 tiene como objetivo validar con los stakeholders los requerimientos especificados mediante el uso de modelos, para corroborar que el equipo involucrado en el desarrollo del software haya logrado un entendimiento correcto de lo requerido por los mismos, antes de avanzar con las fases posteriores de su ciclo de vida. Se deben indicar a los stakeholders las herramientas utilizadas al momento de dar a conocer lo comprendido en cuanto al requerimiento, es decir, el tipo de modelos o la forma en que se presentará la información generada al mismo. Además, se deben indicar los stakeholders con quienes se validará, la fecha en que se realizará la actividad y el resultado de la misma, con las observaciones pertinentes, en caso de ser necesario.

Los registros de información que genera esta actividad se pueden observar en la Tabla 19 a modo de ejemplo.

Tabla 19. Validación de requerimientos

RQ-ID	Herramientas de validación	Stakeholders	Fecha	Resultado
RQ-00001	Prototipo de alto nivel Presentaciones Documentación	GSH-00001	27/03/2017	Aprobado
RQ-00010	Prototipo de bajo nivel	SH-00009	30/03/2017	Rechazado. Volver a especificar.

La actividad 2 tiene como objetivo definir los distintos ensayos del software a realizar, es decir, las pruebas que se deberá realizar a cada requerimiento software para verificar su correcto funcionamiento, definiendo el tipo de prueba a realizar sobre el mismo, las señales de entrada, las señales de salida y los criterios de éxito. Solamente habiendo pasado todos estos ensayos (pruebas), el requerimiento software se dará como verificado. Esta actividad busca cumplir los puntos 7.2.4.11, 7.2.4.12, 7.2.4.17 y 7.2.4.19 de la norma EN-50128, en donde, para cada uno se indica:

- 7.2.4.11 y 7.2.4.12: como ya se indicó anteriormente, en la ERS se deben incluir los requisitos software para llevar a cabo ensayos periódicos de las funciones comunes y de seguridad del mismo, según lo definido en el documento de Especificación de Requisitos de Seguridad del Sistema.
- 7.2.4.17: se deben indicar los ensayos a realizar al software terminado.
- 7.2.4.19: se deben indicar las señales de entrada, de salida y los casos de éxito, incluyendo las prestaciones y calidad que debe cumplir el software para pasar la verificación con éxito.

La información generada por esta actividad se puede ver en los registros de la Tabla 20 que se indica a modo de ejemplo.

Tabla 20. Ensayos del software

RQ-ID	RQ-00001	Tipo de ensayo	Análisis estático de software
Señales de entrada		Contador de 5 segundos del reloj interno del monitor de barreras.	
Señales de salida		Datos del monitor de barreras: ID, fecha, hora y estado actual.	
Criterios de éxito		El monitor de barreras envía sus datos de manera correcta al centro de control. El tiempo de recepción de los mismos es como máximo de un segundo después del envío.	

Las actividades 3 y 4 tienen como objetivos verificar la no contradicción de los requerimientos, entendiendo que se puede llegar a presentar la situación en que un requerimiento interfiera o anule a lo solicitado en otro, y verificar el cumplimiento de los requerimientos con los niveles de seguridad especificados para el software, respectivamente, debiendo definirse qué medidas tomar en los casos en que alguna de las verificaciones no pase las pruebas. A modo de ejemplo, en la Tabla 21 se pueden observar los registros generados tras el desarrollo de estas actividades.

Tabla 21. Contradicciones - Cumplimiento SIL

RQ-ID	Contradicción con RQ-ID	Cumplimiento con nivel de seguridad
RQ-00001	-	Cumple
RQ-00010	RQ-00009	No cumple

La actividad 5 tiene como objetivo verificar los requisitos del software, lo que ayuda a cumplir parcialmente con los puntos 7.2.4.11 y 7.2.4.12 mencionados anteriormente. Para esto, se establece para cada requerimiento especificado una valoración en cuanto a distintos aspectos, definidos en los procesos anteriores. Los mismos son:

- *Adecuación*: indica la adecuación del requerimiento con las especificaciones del sistema.
- *Legibilidad*: indica el grado de comprensión que se logra obtener del mismo al leer la documentación generada.
- *Trazabilidad*: indicador de si el requerimiento puede ser trazado por todos los documentos generados durante su tratamiento.
- *Ensayos*: indica el grado de correctitud y completitud de los ensayos (pruebas) definidos para el requerimiento.
- *Coherencia interna*: indica el grado de correctitud semántica del requerimiento, sin contradecirse o poseer faltas de integridad.
- *Restricciones HW y SW*: indicador de si se definieron correctamente dichas restricciones del requerimiento.
- *Observaciones*: en caso de ser necesario, se pueden indicar correcciones o mejoras al requerimiento para poder ser verificado correctamente.
- *Resultados*: indicador de si el requerimiento pasa o no la verificación realizada.

Los registros de información generados tras la concreción de esta actividad pueden verse en la Tabla 22, a modo de ejemplo:

Tabla 22. Verificación de requerimientos

RQ-ID	RQ-00001	Fecha	25/01/2017	Responsable	Jorge Lopez
Adecuación	10	Legibilidad	8	Trazabilidad	Cumple
Ensayos	8	Coherencia interna	10	Restricciones HW y SW	Cumple
Observaciones	La redacción del requisito no es del todo clara. No se encuentran definidos todos los ensayos del software que involucra este requisito, de acuerdo a los modos de comportamiento definidos sobre el mismo.				
Resultados	Corregir los puntos especificados en las observaciones.				

Por último, la actividad 6 tiene como objetivo la modificación de las especificaciones generadas en el proceso de Especificación, en caso de que en esta fase se llegara a encontrar un motivo para hacerlo, es decir, readecuar la especificación en base a lo verificado o validado, de manera iterativa.

Las actividades definidas en cada uno de estos procesos fueron evolucionando paulatinamente a lo largo del ciclo de vida del proyecto, agregándose algunas nuevas y modificando otras ya existentes, como se puede observar en los capítulos siguientes, con el fin de optimizar el procedimiento final resultante.

Además, se definió el capítulo 8 Entregas, en donde se indica la documentación resultante de la aplicación de este procedimiento, compuesta por los distintos elementos generados durante la implementación del mismo en durante su uso. Dicha documentación consiste en:

- **Especificación de Requisitos Software (ERS)**: compuesta por los elementos resultantes del primer y segundo proceso (Obtención y Especificación). Genera el documento nombrado como “F_ERS_01 Especificación de Requisitos Software - Proyecto XX”, como se puede ver en el *Anexo F_ERS_01 - Primer versión*.
- **Especificación de Ensayos del Software en Conjunto (ESC)**: contiene los resultados de la definición de ensayos del software, presente en el proceso de Verificación y

Validación. Genera el documento nombrado como "F_ESC_01 Especificación de Ensayos del Software en Conjunto - Proyecto XX", como se puede ver en el *Anexo F_ESC_01*.

- **Informe de Validación y verificación de los Requisitos del Software (VRS)**: contiene los elementos que componen al proceso de Verificación y Validación, exceptuando la definición de ensayos de software en conjunto. Genera el documento nombrado como "F_VRS_01 Informe de Verificación de los Requisitos del Software - Proyecto XX", como se puede ver en el *Anexo F_VRS_01*.

La aplicación de este procedimiento, procesos y actividades, y generación de dicha documentación por medio de formularios, es definida en capítulos posteriores.

Esta primera versión del procedimiento, hasta este punto, se puede observar en el *Anexo Primer versión del PG*.

4.1.3 Verificación de cumplimiento con la norma EN-50128

Para garantizar que el procedimiento generado cumpla con la norma EN-50128, teniendo en cuenta los aspectos que la misma menciona, se definió el Anexo 1 - Cumplimiento de UNE-EN-50128:2012, dentro del procedimiento (*Anexo Primer versión del PG*). En el mismo se generó una tabla (que se puede ver en la Tabla 23), con dos columnas: la primera indica el punto de la norma a cumplimentar, es decir, el capítulo e ítem de la norma a verificar, y la segunda el elemento del procedimiento que garantiza el cumplimiento del mismo.

Tabla 23. Cumplimiento con la norma EN-50128

Ítem de UNE-EN 50128	Elemento del procedimiento definido
7.2.1.1	Todo el procedimiento
7.2.1.2	Resultado del Proceso 7.3
7.2.2	Tabla 2
7.2.3	Resultados de los Procesos 7.2 y 7.3
7.2.4.1	Resultado del Proceso 7.2
7.2.4.2	Punto 7.2.3
7.2.4.3	Punto 7.2.2
7.2.4.4 a	Tabla 9
7.2.4.4 b	Identificador de los requerimientos definido en el punto 4 de la Tabla 1
7.2.4.5	Tabla 18: glosario para unificar términos
7.2.4.6	Fila 5 de la Tabla 2
7.2.4.7	Figuras 4 y 6
7.2.4.8	Tabla 16
7.2.4.9	Filas 4 y 5 de la Tabla 14
7.2.4.10	Punto 7.2.7
7.2.4.11	Punto 7.2.5, Tabla 23 y fila 3, columna 2 de la Tabla 27
7.2.4.12	Punto 7.2.5, Tabla 23 y fila 3, columna 2 de la Tabla 27
7.2.4.13	Punto 7.2.2
7.2.4.14	
7.2.4.15	
7.2.4.16	Resultado del Proceso 7.3
7.2.4.17	Tabla 23
7.2.4.18	
7.2.4.19	Tabla 23
7.2.4.20	Resultado del Proceso 7.3
7.4.21	
7.4.22	

Como se puede observar en esta tabla, en esta instancia el procedimiento aún no cumplía ciertos puntos de la norma; los mismos se fueron cumplimentando a medida que evolucionaba e

incrementaba el mismo, y otros se fueron modificando, y pueden ser vistos en capítulos posteriores.

4.1.4 Primera implementación del procedimiento con requerimientos reales

Luego de tener desarrollada la primera versión del procedimiento, se realizó una implementación inicial del mismo mediante requerimientos software reales obtenidos del proyecto del prototipo Monitor de Barreras anteriormente mencionado. Estos requerimientos fueron brindados por Representantes del desarrollo del monitor de barreras para Trenes Argentinos Operaciones (SOFSE) [29], mediante un documento denominado "Prototipo del Monitor de Barreras a ensayar en el INTI" que describe las características que debe tener el mismo, y que puede ser visto en el *Anexo Primeros Requerimientos*, para ensayar en el Instituto Nacional de Tecnología Industrial (INTI) [47]. En el mismo se indican las características de funcionamiento de dicho prototipo, a nivel hardware y software, obteniéndose así las distintas señales de entrada, salidas, interfaces de comunicación y requisitos funcionales del mismo (como por ejemplo, los distintos estados en los que se puede encontrar).

Con la implementación del procedimiento con estos requerimientos preliminares se generaron los siguientes documentos:

- **R_ERS_01 Especificación de Requisitos Software - Monitor de Barreras**. El mismo se puede observar en el *Anexo R_ERS_01 Monitor de Barreras - Primer implementación*.
- **R_ESC_01 Especificación de Ensayos del Software en Conjunto - Monitor de Barreras**. El mismo se puede observar en el *Anexo R_ESC_01 Monitor de Barreras - Primer implementación*.
- **R_VRS_01 Informe de Verificación de los Requisitos del Software - Monitor de Barreras**. El mismo se puede observar en el *Anexo R_VRS_01 Monitor de Barreras - Primer implementación*.

Los requisitos funcionales solicitados por los stakeholders en la primera educción se pueden ver a continuación:

El Monitor de Barreras podrá estar en uno de estados que se muestran en la Fig. 20.

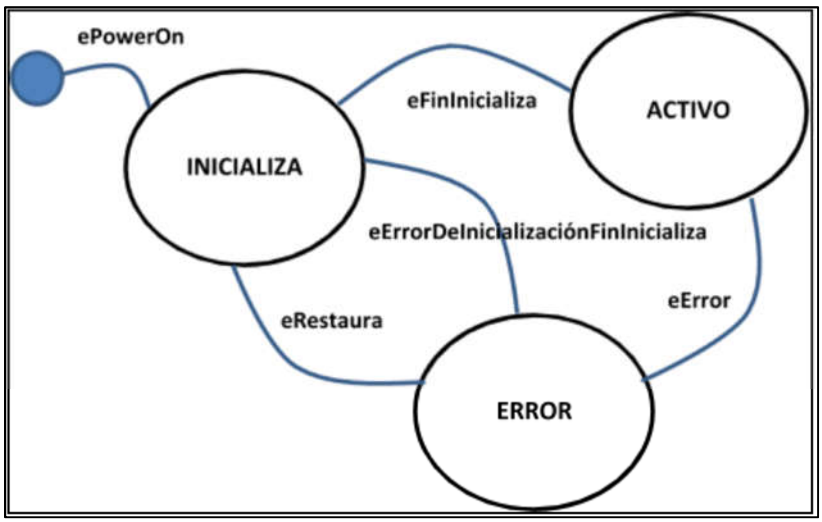


Fig. 20. Diagrama de estados del monitor de barreras

Al energizar el sistema se inicia en el estado INICIALIZA. En este estado se realiza una comprobación general del sistema.

Finalizado con éxito el proceso de inicialización el sistema evoluciona al estado ACTIVO. Este estado se monitorea el estado de los sensores y se activa las señales luminosas de indicación de estado. De no existir fallas el sistema transmite cada 10 minutos el estado del sistema de barreras y ante la aparición de un evento también transmite.

En estado ACTIVO el sistema debe comprobar que al activarse la señal de ocupación de vía la barrera se baja en un tiempo acorde al esperado (lo que se determina a partir de las señales de barrera arriba y barrera abajo) y que se activan la sirena y las luces.

Al desactivarse la señal de ocupación de vía el sistema debe comprobar que la barrera sube y que se desactiva la sirena y las luces.

Al estado de ERROR se ingresa ante un evento de error relacionado con temperatura, tensión de batería, etc.

A modo ejemplificativo, se muestra uno de los requerimientos, con su respectiva documentación generada a lo largo del procedimiento.

Se toma como ejemplo el tratamiento del requerimiento funcional que indica "*... Al energizar el sistema se inicia en el estado INICIALIZA. En este estado se realiza una comprobación general del sistema. ...*"

Primero se analizó la documentación de entrada, generándose la Tabla 24 indicada en la actividad 1 del proceso de Obtención: "*Realizar un análisis sistemático de la documentación de entrada.*". Los documentos se encontraban en distintas etapas de desarrollo, encontrándose uno solo completo, dos en versión preliminar (es decir, en proceso de desarrollo) y otros dos a ser definidos (To Be Defined, TBD).

Tabla 24. Análisis de la documentación de entrada

Documento	Fecha de análisis	Completitud	Correctitud
Especificación de Requisitos del Sistema	09/08/2017	Versión preliminar	Versión preliminar
Especificación de Requisitos de Seguridad del Sistema	09/08/2017	De acuerdo con UNE-EN 50578	De acuerdo con UNE-EN 50578
Descripción de la Arquitectura del sistema	09/08/2017	Versión preliminar	Versión preliminar
Especificaciones de la Interfaz Externa	09/08/2017	Versión preliminar	Versión preliminar
Plan de Aseguramiento de Calidad del Software	09/08/2017	TBD	TBD
Plan de Validación del Software	09/08/2017	TBD	TBD
Observaciones	Sin cambios mayores desde la redacción del documento R_ERS_01 Especificación de Requisitos Software - Probador de relé		

Luego de esto, se realizó lo indicado en la actividad 2 del proceso de Obtención: "*Identificar los stakeholders y canales de entrada de los requerimientos. Los stakeholders serán identificados mediante un valor con el formato SH-XXXXX. Los grupos de stakeholders serán identificados mediante un valor con el formato GSH-XXX.*", generándose la Tabla 25 y la

Tabla 26. El SH-0001 y el SH-0002 fueron agrupados como representantes del desarrollo del proyecto, creándose el GH-01. En este caso, el primero fue el stakeholder encargado de enviar el documento de requerimientos mencionado.

Tabla 25. Identificación de stakeholders y canales de comunicación

Identificador	Nombre y apellido	Puesto o función	Canales de comunicación
SH-0001	Ariel Lutenberg	Director GICSAFE	email: alutenberg@gmail.com tél: +54 9 11 5844-3749
SH-0002	Martín Harris	Coordinador de Desarrollos	Indirecto a partir de Ariel Lutenberg email: martin.harris@sofse.gob.ar
SH-0003	Mariano Soler	Subgerente Desarrollo y Normas Técnicas	Indirecto a partir de Ariel Lutenberg / Martín Harris email: mariano.fernandez@sofse.gob.ar
SH-0004	Adrián Laiuppa	Director del Proyecto Probador de Relé y desarrollador del sistema	email: alaiuppa@gmail.com tél: +54 9 291 643-4357

Tabla 26. Grupos de stakeholders

Identificador de grupo	Nombre de grupo	Descripción de grupo	Miembros del grupo
GH-01	SOFSE	Representantes del desarrollo del monitor de barreras para SOFSE	SH-0001 y SH-0002

Se definieron las técnicas de elicitación, tal como se menciona en la actividad 3 del proceso de Obtención: "*Definir el formato y las técnicas de elicitación a utilizar, al igual que la planificación.*", generándose la Tabla 27. Se decidió utilizar diagramas UML de transición de estados, generando elementos de especificación semi formal que sirviera para la comunicación entre los stakeholders y los miembros del grupo de proyecto. Estos iniciarían su proceso desde la actividad de análisis y definición de los modos de comportamiento del software.

Tabla 27. Técnicas de elicitación

Técnica	Duración aproximada	Objetivos
Prototipado mediante diagrama de estados	10 horas.	Realizar el análisis y la especificación de los requerimientos funcionales del monitor de barreras.

Se realizó la actividad 4 del proceso de Obtención, que indica: "*Realizar la elicitación de los requerimientos. Los requerimientos serán identificados mediante un valor con el formato RQ-XXXXX.*", generándose la Tabla 28. Para esto, se tomó como base el documento enviado por los stakeholders. Las dudas que existieron se consultaron vía email o de manera telefónica, hasta entender concretamente el pedido.

Tabla 28. Elicitación de requerimientos

RQ-ID	000001	Fecha	09/08/2017	Fuente	GH-01
Necesidad	El monitor de barreras debe realizar una comprobación general del sistema al inicializarse.				
Motivo	Controlar que los distintos componentes del sistema se encuentren funcionando correctamente al iniciar su operación.				
Objetivo	Inicializar el sistema en un estado seguro, sin fallas internas de los componentes que puedan conducir a peligros.				
Verificación	● Se enciende el indicador luminoso correspondiente. ● El monitor de barreras envía información relativa al estado general del sistema al centro de operaciones y la misma se recibe de manera correcta El sistema entonces pasa a estado ACTIVO.				

Posteriormente se procedió con el desarrollo de las actividades 1 y 2 del proceso de Especificación, que indican "*Realizar un análisis operacional de los requerimientos.*" y "*Realizar un análisis sistémico de los requerimientos.*" respectivamente, plasmándose los resultados como se puede ver en la Tabla 29. Se tuvieron en cuenta la especificación de los atributos o cualidades del software a desarrollarse a tener en cuenta, como se mencionó en la restricción de este proceso.

Tabla 29. Análisis de requerimientos

RQ-ID	000001			
Análisis operacional	Fecha	09/08/2017	Responsable	Cristian Pinto Luft
Resultados	Se comprobarán la barrera, los brazos, el semáforo PaN, la puerta del gabinete (abrigo), los relés de accionamiento, la energía y la temperatura del habitáculo del monitor de barreras, el suministro de energía primario, los rectificadores, los motores, la lámpara y la campana. Las comprobaciones hechas serán enviadas mediante una interfaz de comunicaciones Bluetooth o WiFi hacia el centro de operaciones.			
Análisis sistémico	Fecha	09/08/2017	Responsable	Cristian Pinto Luft
Resultados	● La comprobación se llevará a cabo cuando el sistema se encuentre en estado INICIALIZA. De finalizarse con éxito, el estado cambiará a ACTIVO. ● Los resultados se representarán con el formato JSON y serán enviados			

	mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa correspondiente.

Se realizó la actividad 3 del proceso de Especificación, que indica: "*Definir los distintos modos de comportamiento del software.*", obteniéndose los resultados de la Tabla 30.

Tabla 30. Modos de comportamiento del software

RQ-ID	RQ-000001
Estado inicial	Apagado
Evento	Se enciende el monitor de barreras
Estado final	Encendido - ACTIVO
Resultado	El monitor de barreras se enciende, realiza las comprobaciones correspondientes y emite una señal lumínica local y una señal digital con datos del estado al centro de control.

Luego se realizó la actividad 4 del proceso de Especificación, que indica: "*Definir atributos característicos de los requerimientos*", obteniéndose la Tabla 31. La prioridad y el impacto en la seguridad se definieron como "altas" debido a la restricción que menciona la necesidad de la autocomprobación periódica del SW y el HW en estos sistemas en la norma. El esfuerzo se estimó con base en la cantidad de personas involucradas en el proyecto y la carga de trabajo de los mismos.

Tabla 31. Atributos de los requerimientos

RQ-ID	000001	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	40 horas	Impacto en la seguridad	9
Restricciones de HW		● El sistema debe ser alimentado con energía de manera contínua durante el proceso de comprobación. ● Los distintos componentes a controlarse deben estar correctamente conectados y configurados en el sistema para su comprobación. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.			
Restricciones de SW		● El sistema debe encontrarse en estado INICIALIZA.			

Los resultados de la actividad 5 del proceso de Especificación, que indica "*Construir un glosario de términos*", pueden verse en la Tabla 32. Estos términos fueron identificados como de

conocimiento necesario por los stakeholders y los miembros del equipo de trabajo para lograr una comprensión mutua de lo que se está realizando.

Tabla 32. Glosario de términos

Término	Definición
TBD	To Be Defined, a definir
HW/SW	Hardware/Software
Timestamp	Marca de tiempo
Rollback	Volver un objeto a un estado anterior al actual
JSON	JavaScript Object Notation, Notación de Objetos de JavaScript
BD	Base de Datos
SQLite	Sistema de gestión de BD relacional

La actividad 6 del proceso de Especificación, que indica "*En caso de generarse nuevos requerimientos a partir de la especificación de los actuales, educirlos.*", no fue llevada a cabo, ya que no se generaron nuevos requerimientos, producto del análisis del presente.

Se pasó al proceso de Verificación, con la actividad 1, que indica "*Verificar con los stakeholders los requerimientos especificados mediante el uso de modelos.*", generándose la Tabla 33 para corroborar lo mismo. Para esto, se utilizarían los diagramas UML de transición de estados y la misma documentación generada por el procedimiento al momento de realizar dicha validación.

Tabla 33. Validación de requerimientos con stakeholders

RQ-ID	Modelo de validación	Stakeholders	Fecha	Resultado
RQ-000001	Diagramas, documentación	GH-01	01/09/2017	A validar

Luego se pasó a la actividad 2 del proceso de Verificación: "*Definir los distintos ensayos del software a realizar.*", generando la Tabla 34. El lapso de tiempo establecido es configurable para el sistema del nivel monitor.

Tabla 34. Ensayos del software

RQ-ID	RQ-000001	Tipo de ensayo	Funcional
Señales de entrada		Entrada de corriente en el monitor de barreras	

Señales de salida	Informe de comprobación del estado de los componentes
Criterios de éxito	El monitor de barreras envía un informe de comprobación de componentes a la central de control dentro del lapso de tiempo establecido.

Se realizaron las actividades 3 y 4 del proceso de Verificación: "*Verificar la no contradicción de los requerimientos.*" y "*Verificar el cumplimiento de los requerimientos con los niveles de seguridad especificados para el software.*", respectivamente, generándose la Tabla 35. En un principio, no existe contradicción con ningún otro requerimiento, ya que este es el primero generado con el procedimiento; más adelante algún requerimiento nuevo podría entrar en conflicto con la especificación de este.

Tabla 35. Verificación de contradicciones y seguridad de los requisitos

RQ-ID	Contradicción con RQ-ID	Cumplimiento con nivel de seguridad
RQ-000001	-	Cumple

Luego se realizó la actividad 5 del proceso de Verificación, que indica "*Verificar los requisitos del software*", generándose la Tabla 36. Para esto se analizaron todos los atributos definidos, tal como se indicó anteriormente, realizando la verificación con éxito. El único inconveniente fue que, a falta de un rol de Verificador de Requerimientos, el mismo Gestor de Requerimientos fue quien llevó a cabo la actividad, teniendo que solventarse esto en el futuro.

Tabla 36. Verificación de los requisitos del software

RQ-ID	RQ-000001	Fecha	10/08/2017	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	10	Trazabilidad	10
Ensayos	9	Coherencia interna	9	Restricciones HW y SW	9
Observaciones	Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	El requerimiento cumple con todos los puntos indicados de manera correcta				

La última actividad, la 6, que indica "*En caso de generarse modificaciones en los requerimientos durante el proceso, modificar las especificaciones.*", no fue llevada a cabo, ya que el requisito no sufrió modificaciones durante la ejecución del proceso.

Todos estos registros fueron colocados en sus respectivos formularios (o documentos), tal como se indicó anteriormente.

4.1.5 Agregado de técnicas de seguridad al procedimiento

Luego de la primera implementación piloto del procedimiento, se detectó la falta de técnicas de análisis para los aspectos relacionados a la seguridad en los requisitos software especificados, que garanticen el cumplimiento de dichos aspectos en relación a la normativa utilizada, y la generación de software seguro, requisito fundamental para la generación de software crítico. Se identificaron como unas de las técnicas principales utilizadas para este tipo de análisis a [17][48]:

- **Software Failure Tree Analysis (SFTA)**: análisis del árbol de fallas del software. Este es un análisis de fallas deductivo descendente, realizado con lógica Booleana, en el que se toma como evento principal un peligro presente en el software, y se realiza un análisis de sus eventos desencadenantes, hasta llegar a eventos básicos (o fallas, irreducibles en la práctica o para los fines deseados).
- **Software Failure Modes and Effect Analysis (SFMEA)**: análisis de modos de fallas y efectos del software. Este es un análisis estructurado en el que se analiza cada una de las posibles fallas del software, definiendo sus posibles causas, modos de fallos y efectos. Además, se pueden identificar sus formas de detección, mitigación, prevención, severidad y frecuencia (de estas últimas dos características se puede derivar el riesgo). Este análisis puede permitir detectar nuevos peligros.

Cabe destacar que la técnica Preliminary Hazard Analysis (PHA, o análisis preliminar de peligros) [49] que se menciona entre los resultados de la RSL mencionada, debería ser realizada a nivel del sistema en general, no solamente del subsistema software, en etapas iniciales del análisis del sistema, por lo que no se la incluyó en el procedimiento generado. Esta técnica si podría generar requerimientos software de seguridad en caso de realizarse correctamente.

Además de estas dos técnicas, realizando una búsqueda adicional de información acerca de estas técnicas, se encontró la siguiente [50]:

- **Software Success Tree Analysis (SSTA)**: análisis del árbol de éxito del software. Este es un análisis de casos de éxito deductivo descendente, realizado con lógica Booleana, en el que se toma como evento principal un caso de éxito de una funcionalidad del software, y se realiza un análisis de sus eventos desencadenantes, hasta llegar a eventos básicos. Es decir, es similar al SFTA, con la diferencia de que en el SSTA se analiza la cadena de eventos que provoca que el software cumpla correctamente su función, siendo uno el complemento del otro (SSTA=SFTA').

Estas tres técnicas encontradas se añadieron al procedimiento generado, vinculándolas para obtener un resultado, de la siguiente manera:

1. Primero se realiza el análisis SSTA del requerimiento en cuestión, obteniéndose el árbol de casos de éxito del software.
2. Luego se realiza el análisis SFTA, mediante el complemento del árbol SSTA obtenido en el punto anterior, obteniéndose el árbol de fallas del software, como se indica en [50].
3. Por último, se realiza el análisis SFMEA, mediante el análisis de cada una de las fallas detectadas en el punto anterior, obteniéndose información relativa a la seguridad del software que retroalimentará el diseño del mismo, o incluso podrá generar nuevos requerimientos software, como se indica en [48].

Esta vinculación de técnicas se puede observar en la Fig. 21, en la que se puede ver que:

- A partir de la aplicación de SSTA a requerimientos software (RQ) se pueden obtener las condiciones (funcionalidades) necesarias para que el subsistema alcance el éxito en sus funciones.
- A partir del complemento del SSTA se obtiene el SFTA (SSTA'=SFTA), con el que se pueden obtener las causas de falla de los distintos posibles peligros del software involucrados en el requerimiento (CN y DN). Dichos RQ pueden pertenecer a distintos sub-sistemas software (como ser, de frenos, de aceleración, de señalización, etc.).
    - A partir de dicho análisis se pueden descubrir causas de fallas externas al sistema software, como ser causas pertenecientes al hardware (EN). Dichos descubrimientos se deben documentar e informar pertinentemente, para complementar los análisis de seguridad llevados a cabo sobre el sistema

correspondiente.

- A partir de la aplicación de SFMEA a las causas detectadas anteriormente se puede obtener la información de seguridad generada por esta técnica.
    - Además, en este paso se pueden llegar a generar nuevos requerimientos relativos a la seguridad, ya sea del subsistema software o de otro (RQ-M).

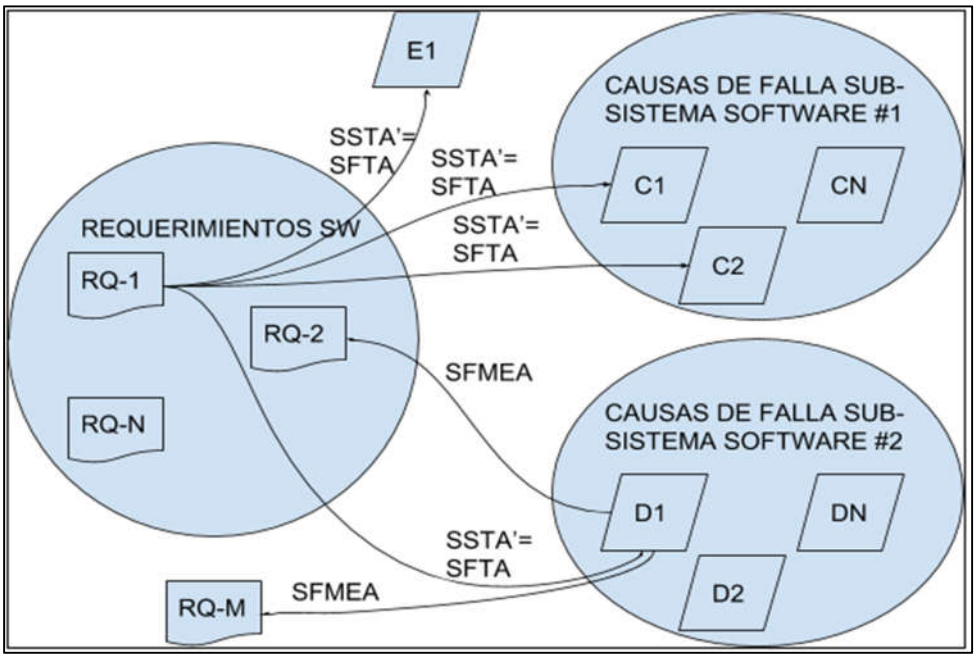


Fig. 21. Análisis de seguridad de sistemas software de seguridad crítica

Además del agregado de técnicas de seguridad, con la continuación del análisis y comprensión de la norma EN-50128, y la documentación utilizada en el proyecto, se detectaron mejoras y correcciones a aplicar al procedimiento. Se procedió a incluir estas nuevas técnicas y cambios al procedimiento ya generado, obteniéndose la segunda versión (o incremento) del mismo. Los principales cambios que se realizaron fueron:

En el proceso de **Obtención**:

- Se agregó a la descripción de la actividad 4: "*Los requerimientos de seguridad serán identificados mediante un valor con el formato RQS-XXXXX.*". Esto es, para identificar correctamente los requerimientos software de seguridad de los demás, pudiendo detectarlos rápidamente al leer la ERS.

- Se agregó la restricción 7.1.3: "*Una de las fuentes de entrada de los requerimientos del software son los requerimientos de seguridad del software identificados en el 7.2 Proceso de Identificación de secuencias de acontecimiento de peligros del PG_ ARI_10 Análisis de Riesgos.*". Estos riesgos detectados afectan directamente al software, por lo que se deben gestionar mediante este procedimiento.

En el proceso de **Especificación**:

- Se agregó la restricción 7.2.6: "*Cuando sea necesario, se deben expresar los requerimientos mediante métodos formales.*". Cumpliendo con el punto 7.2.4.15 de la norma EN-50128, en la que se indica que se debe utilizar un tipo u otro de técnica de especificación de acuerdo al nivel de SIL del software; por ejemplo: métodos formales, modelado, metodología estructurada o tablas de decisión.
- Se modificó la restricción 7.2.7 a: "*Los distintos estados por los que puede pasar el software serán modelados mediante un árbol SSTA. El evento iniciador del árbol será el estado descrito por la verificación del requerimiento ingresado.*". Se conjugan así el análisis de los estados con la seguridad (estados "seguros"). La aplicación de SFTA produciría los estados "inseguros" del software.
- Se agregó la restricción 7.2.9: "*Se deben realizar análisis del impacto de la seguridad de cada uno de los requerimientos software especificados, para detectar los peligros que podrían llegar a provocar. Esta tarea se realizará a nivel funcional mediante el uso complementario de las técnicas de análisis de seguridad SFMEA y SFTA.*". Esto es, la aplicación de las técnicas de seguridad descritas anteriormente.
- Se modificó la actividad 3, convirtiéndola en la 4 y cambiando su registro a: "*Árbol SSTA generado con la Figura 4.*". Es decir, se agregó la técnica SSTA y se la utilizó como registro de esta actividad, reemplazando la tabla que generaba anteriormente (la de Modos de comportamiento del software). Los símbolos básicos para la construcción del mismo se ven en la Fig. 22.

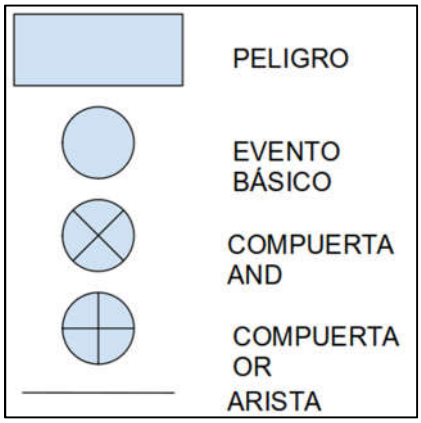


Fig. 22. Símbolos básicos SSTA

- Se agregó la actividad 5: "*Definir los distintos modos de fallo del software.*", a cargo del Gestor de Requerimientos, que genera el registro "*Árbol SFTA generado con el complemento del árbol SSTA*". Esto cumple con el punto 7.2.4.8 de la norma EN-50128, en donde se hace hincapié en el análisis y la documentación de todos los modos de fallo.
- Se agregó la actividad 6: "*Realizar el análisis de seguridad SFMEA funcional de los requerimientos especificados.*", a cargo del Gestor de Requerimientos. La misma está destinada también a cumplir el punto 7.2.4.8 de la norma EN-50128, y a dotar de mayor seguridad a la ERS. Genera como resultado la Tabla 37.

Tabla 37. SFMEA funcional de los requerimientos

RQ-ID/SFMEA-ID					
Modo de fallo					
Efecto	**Local**				
	Subsistema				
	Sistema				
Causas					
Detección					
Mitigación					
Prevención					
Severidad		**Frecuencia**		**Riesgo**	

- Se agregó la restricción 7.2.12: "*Cada caso puede generar un nuevo requerimiento de seguridad del software o del sistema, y se pueden descubrir nuevos peligros.*". En caso de generarse, se debe introducir el requerimiento descubierto al procedimiento desde el proceso de Obtención y analizarse.
- Se agregó la restricción 7.2.13: "*Por cada evento básico detectado en el árbol SFTA anteriormente generado se realiza un análisis SFMEA.*". Como se explicó anteriormente, cada evento básico del árbol SFTA representa un fallo, que debe ser analizado mediante SFMEA.
- Se agregó la restricción 7.2.14: "*Los grados de severidad, en orden ascendente pueden ser: menor, mayor, crítico o catastrófico.*". Esto hace referencia al daño que puede llegar a causar en el sistema la falla en caso de ocurrir, como se indica en [13].
- Se agregó la restricción 7.2.15: "*La frecuencia de ocurrencia, en orden ascendente puede ser: muy rara, remota, ocasional, probable o frecuente.*". Esto hace referencia a la probabilidad de ocurrencia de la falla en el sistema, como se indica en [13].

Se agregó la restricción 7.2.16: "*El grado de riesgo, en orden ascendente puede ser: aceptable, semi aceptable o no aceptable. Para evaluar este último se debe utilizar la Tabla de Riesgos que se propone a continuación.*". Se debe definir el grado de riesgo al realizar el análisis SFMEA para cada requerimiento, buscando la intersección de frecuencia y severidad [13] en la Tabla 38 de riesgos, como se puede observar a continuación.

Tabla 38. Tabla de riesgos

Frecuencia/ severidad	**Muy rara**	**Remota**	**Ocasional**	**Probable**	**Frecuente**
Catastrófico	S-Aceptable	N-Aceptable	N-Aceptable	N-Aceptable	N-Aceptable
Crítico	Aceptable	S-Aceptable	S-Aceptable	N-Aceptable	N-Aceptable
Mayor	Aceptable	Aceptable	S-Aceptable	S-Aceptable	N-Aceptable
Menor	Aceptable	Aceptable	Aceptable	S-Aceptable	S-Aceptable

En el proceso de **Verificación**:

- Se cambió su nombre a: "***Verificación y validación***". Esto es, debido a que se diferenciaron correctamente las actividades que hacen referencia a verificar el

cumplimiento de las restricciones y propiedades definidas para cada requerimiento, y las validaciones que se realizan de los mismos con los stakeholders.

- Se modificó la restricción 7.3.1 a: "*El proceso de validación de requerimientos del software indica las pautas a seguir para validar las especificaciones de requerimientos generadas en el proceso anterior con los stakeholders, modificarlos y redefinirlos en un proceso iterativo de ser necesario, definir los ensayos a realizarse sobre los mismos.*". De acuerdo a lo expresado anteriormente.
- Se modificó la restricción 7.3.2 a: "*Los requerimientos pueden ser validados con stakeholders individuales o grupos identificados.*". Esto es, dependiendo del stakeholder o grupo de stakeholders que hayan iniciado el requerimiento.
- Se agregó la restricción 7.3.3: "*El proceso de verificación de requerimientos del software indica las pautas para llevar a cabo los distintos tipos de controles y pruebas técnicas a llevarse a cabo sobre los requerimientos especificados, para mantener el nivel de seguridad requerido del software.*". De acuerdo a lo expresado anteriormente.
- Se agregó la restricción 7.3.5: "*Los tipos de Ensayos del software a utilizar pueden ser: ensayos de las prestaciones, funcionales/de caja negra o modelado.*". Esto cumple con el punto 7.2.4.18 de la norma EN-50128, que indica las principales técnicas que se pueden utilizar para realizar dichos ensayos, de acuerdo al nivel de SIL del software.

En las **Entregas**:

- Se modificó el punto 1.g a: "*Definición de modos de comportamiento del software mediante los diagramas SSTA y SFTA generados con la Figura 4.*". Como se expresó anteriormente, dichos modos de comportamiento se empezaron a modelar con estos diagramas en vez de con una tabla.
- Se agregó el punto 1.i: "*Análisis SFMEA funcional de los requerimientos de la Tabla 16.*". De esta forma, se deja registro de dicho análisis en el correspondiente formulario.

Se completaron los puntos faltantes en la tabla del Anexo 1 - Cumplimiento de UNE-EN 50128:2012, dentro del procedimiento (*Anexo Segunda versión del PG*) quedando como se puede ver a continuación en la Tabla 39.

Tabla 39. Cumplimiento de UNE-EN 50128:2012

Ítem de UNE-EN 50128	Elemento del procedimiento definido
7.2.1.1	Resultado del Proceso 7.1 y del Proceso 7.2
7.2.1.2	Resultado del Proceso 7.3
7.2.2	Tabla 2
7.2.3	Resultados de los Procesos 7.2 y 7.3
7.2.4.1	Resultado del Proceso 7.2
7.2.4.2	Punto 7.2.3 y Fila 2 de la Tabla 14
7.2.4.3	Punto 7.2.2
7.2.4.4 a	Tabla 9
7.2.4.4 b	Identificador de los requerimientos definido en el punto 4 de la Tabla 1
7.2.4.5	Tabla 18: glosario para unificar términos
7.2.4.6	Fila 5 de la Tabla 2 y Fila 4 y 5 de la Tabla 12
7.2.4.7	Tabla 14
7.2.4.8	Tabla 16
7.2.4.9	Filas 4 y 5 de la Tabla 16
7.2.4.10	Punto 7.2.7
7.2.4.11	Punto 7.2.5, Tabla 23 y fila 3, columna 2 de la Tabla 27
7.2.4.12	Punto 7.2.5, Tabla 23 y fila 3, columna 2 de la Tabla 27
7.2.4.13	Punto 7.2.2, Tabla 12 y Fila 2 de la Tabla 25
7.2.4.14	Cuando el impacto en la seguridad de la Tabla 16 es cero
7.2.4.15	Punto 7.2.6
7.2.4.16	Resultado del Proceso 7.3
7.2.4.17	Tabla 23
7.2.4.18	Punto 7.3.5
7.2.4.19	Tabla 23
7.2.4.20	Resultado del Proceso 7.3
7.2.4.21	Tabla 27
7.2.4.22	Tabla 27

Esta segunda versión del procedimiento, hasta este punto, se puede observar en el *Anexo Segunda versión del PG*, y el documento de registro de la ERS modificado en el *Anexo F_ERS_01 - Segunda versión*.

4.1.6 Segunda implementación del procedimiento con requerimientos reales

Luego de haber agregado las distintas técnicas para realizar análisis de seguridad a los requerimientos, se procedió a implementarlas en los requerimientos ya especificados con anterioridad. De esta forma, por cada uno de estos requerimientos, se generó su correspondiente SSTA y SFTA, como se puede ver en las Fig. 23 y Fig. 24, correspondientes al primer requerimiento tomado a modo de ejemplo en este desarrollo.

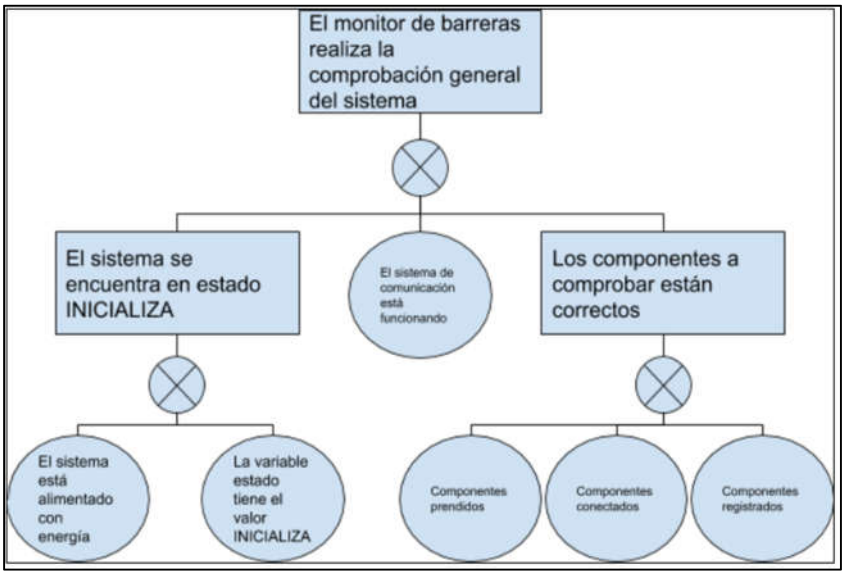


Fig. 23. Árbol SSTA

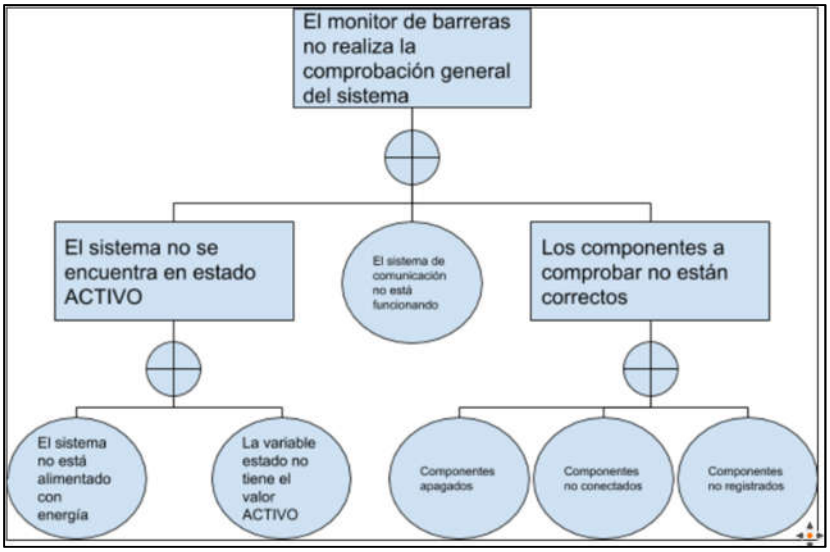


Fig. 24. Árbol SFTA

Como se puede visualizar, y como se mencionó anteriormente, el árbol SFTA es el complemento del SSTA (SFTA'=SSTA). En este caso, el árbol SFTA se encontraba compuesto por seis eventos básicos (o posibles fallas del software), y por cada uno de estos se generó su correspondiente análisis SFMEA, obteniéndose nuevos aspectos de seguridad del software y del hardware a ser considerados en el diseño del sistema, como se puede observar en las Tabla 40, Tabla 41,

Tabla 42, Tabla 43,

Tabla 44 y Tabla 45.

Tabla 40. SFMEA-000001

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000001				
Modo de fallo		El sistema no está alimentado con energía				
Efecto	**Local**	No se pueden realizar comprobaciones				
	Subsistema	El sistema de monitor de barreras no funciona				
	Sistema	Peligro de falta de señalización en el PaN				
Causas		• El sistema fué apagado intencionalmente • El sistema fué apagado no intencionalmente				
Detección		No se reciben señales del monitor de barreras desde el centro de control				
Mitigación		Si fué apagado no intencionalmente, se intenta prender el sistema de manera remota, y de no funcionar se envía a un operario a que lo prenda manualmente				
Prevención		• Se mantiene al monitor de barreras con redundancia de fuentes de energía. • Se utilizan baterías con reserva de energía para suministrar al monitor de barreras hasta un día.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

Tabla 41. SFMEA-000002

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000002				
Modo de fallo		La variable estado no tiene el valor ACTIVO ni su estado interno OK				
Efecto	**Local**	No se realiza la comprobación inicial				
	Subsistema	El sistema de monitor de barreras se encuentra en estado EN CONFIGURACIÓN o apagado.				
	Sistema	-				
Causas		• El sistema está apagado • El sistema se encuentra en estado EN CONFIGURACIÓN				
Detección		Chequeo del estado del monitor de barreras desde el centro de control				
Mitigación		-				
Prevención		-				
Severidad		Menor	**Frecuencia**	Ocasional	**Riesgo**	Aceptable

Tabla 42. SFMEA-000003

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000003				
Modo de fallo		El sistema de comunicación no está funcionando				
Efecto	**Local**	No se pueden comunicar los resultados de las comprobaciones ni se pueden recibir órdenes del centro de operaciones				
	Subsistema	El sistema de monitor de barreras no puede comunicar sus datos ni recibir órdenes				
	Sistema	Desconocimiento del estado de la barrera y sus componentes				
Causas		Falla del sistema de comunicación o de uno de sus componentes (HW/SW)				
Detección		No se reciben señales del monitor de barreras desde el centro de control				
Mitigación		Se intenta restablecer el sistema de comunicación a distancia, y de no ser posible se envía personal especializado hasta el lugar.				
Prevención		Se mantiene al monitor de barreras con redundancia de canales de comunicación				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

Tabla 43. SFMEA-000004

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000004				
Modo de fallo		Componentes apagados				
Efecto	**Local**	No se pueden comprobar los componentes apagados				
	Subsistema	No se pueden utilizar de los componentes apagados				
	Sistema	Peligro de falta de señalización				
Causas		• Falla de HW/SW del componente que no le permite arrancar • Apagado intencional de un componente por mantenimiento				
Detección		• Al realizar la comprobación, luego de N segundos el monitor de barreras no recibe respuesta del estado del componente. • El monitor de barreras informa al centro de control sobre los componentes no disponibles.				
Mitigación		Se intenta prender el componente a distancia, y de no ser posible se envía personal especializado hasta el lugar.				
Prevención		Se mantiene al monitor de barreras con redundancia de componentes críticos que no se deberían apagar				
Severidad		Crítico	**Frecuencia**	Ocasional	**Riesgo**	S-Aceptable

Tabla 44. SFMEA-000005

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000005				
Modo de fallo		Componentes no conectados				
Efecto	**Local**	No se pueden comprobar los componentes desconectados				
	Subsistema	No se pueden utilizar de los componentes desconectados				
	Sistema	Peligro de falta de señalización				
Causas		• Desconexión intencional de un componente por mantenimiento • Desconexión intencional de un componente por vandalismo				
Detección		• Al realizar la comprobación, luego de 2 segundos el monitor de barreras no recibe respuesta del estado del componente. • El monitor de barreras informa al centro de control sobre los componentes no disponibles.				
Mitigación		Se envía personal especializado hasta el lugar para conectar los componentes desconectados.				
Prevención		• Cada vez que el personal técnico realiza un mantenimiento, llena una planilla de comprobación de conexión correcta de los componentes principales. • Se mantienen los componentes del monitor de barreras bien protegidos, sin acceso por parte del público a los mismos.				
Severidad		Crítico	**Frecuencia**	Remoto	**Riesgo**	S-Aceptable

Tabla 45. SFMEA-000006

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000006				
Modo de fallo		Componentes no registrados				
Efecto	**Local**	No se pueden comprobar los componentes no registrados				
	Subsistema	No se conoce el estado de los componentes no registrados				
	Sistema	Peligro de falta de señalización				
Causas		Falta de registración en el software del monitor de barreras de un nuevo componente luego de haberlo conectado y prendido.				
Detección		El monitor de barreras no envía ningún tipo de información del componente en la señal de comprobación enviada al centro de control.				
Mitigación		Se registra correctamente el componente en el software del monitor de barreras mediante el software definido.				
Prevención		Cada vez que se añade, se reemplaza o se conecta un nuevo componente, se debe verificar que el mismo haya sido registrado correctamente.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

Además, se crearon nuevos requerimientos y se completaron o modificaron algunos ya existentes, debido a que los stakeholders enviaron información más detallada de los posibles estados del software, mediante un nuevo diagrama de estados, como se puede ver en la Fig. 25. Estos nuevos requerimientos fueron sometidos al procedimiento actualizado con sus correspondientes análisis de seguridad, e integrados a los ya existentes.

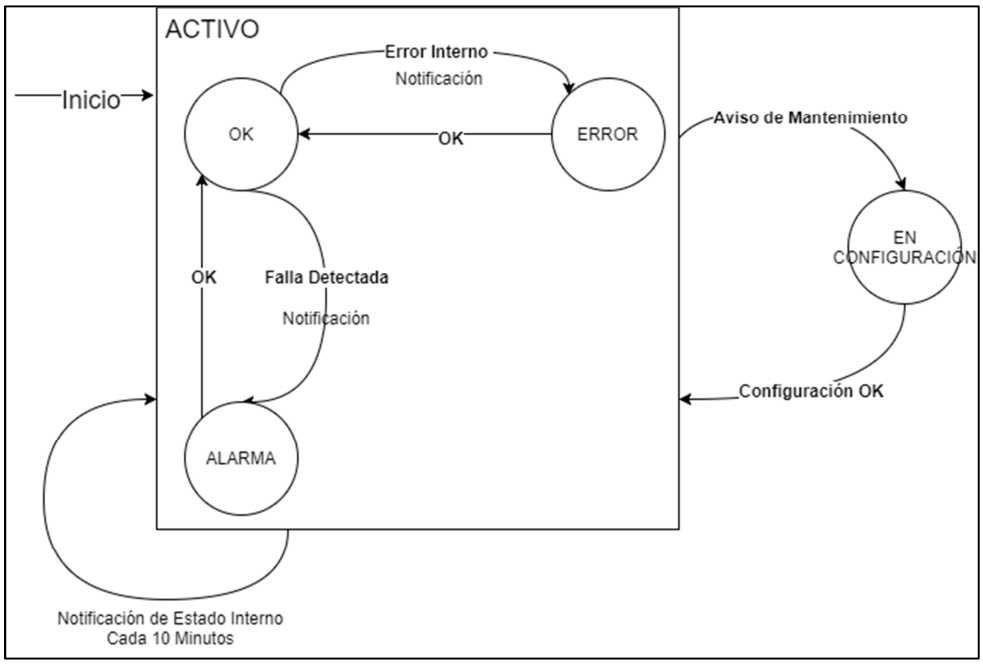


Fig. 25. Diagrama de estados detallado

El requerimiento aquí mostrado a modo de ejemplo sufrió modificaciones, ya que se añadió el sub-estado interno OK al estado ACTIVO anteriormente definido, reemplazando al estado INICIALIZA, quedando documentado como se puede ver en la

Tabla 46, Tabla 47 y

Tabla 48.

Tabla 46. Especificación de requerimientos

RQ-ID	000001	Fecha	09/08/2017	Fuente	GH-01
Necesidad	El monitor de barreras se inicia en un estado ACTIVO y se realiza la comprobación de componentes de manera periódica. Mientras no se detecten fallas en sistema se mantendrá en estado ACTIVO con estado interno OK.				
Motivo	Controlar que los distintos componentes del sistema se encuentren funcionando correctamente durante su operación.				
Objetivo	Comprobar que el sistema se encuentre en un estado seguro, sin fallas internas de los componentes que puedan conducir a peligros.				
Verificación	● Se enciende el indicador luminoso correspondiente. ● El monitor de barreras envía información relativa al estado general del sistema al centro de operaciones y la misma se recibe de manera correcta. ● El sistema se encuentra en estado ACTIVO~~.~~ con estado interno OK.				

Tabla 47. Análisis de requerimientos

RQ-ID	000001			
Análisis operacional	Fecha	09/08/2017		Cristian Pinto Luft
Resultados	Se comprobarán la barrera, los brazos, el semáforo PaN, la puerta del gabinete (abrigo), los relés de accionamiento, la energía y la temperatura del habitáculo del monitor de barreras, el suministro de energía primario, los rectificadores, los motores, la lámpara y la campana. Las comprobaciones hechas serán enviadas mediante una interfaz de comunicaciones Bluetooth o WiFi hacia el centro de operaciones.			
Análisis sistémico	Fecha			
Resultados	● La comprobación se llevará a cabo cuando el sistema se encuentre en estado ~~INICIALIZA~~ACTIVO. De finalizarse con éxito, el estado cambiará a ACTIVO con estado interno OK. ● Los resultados se representarán con el formato JSON y serán enviados mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa correspondiente.			

Tabla 48. Atributos de los requerimientos

RQ-ID	000001	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	40 horas	Impacto en la seguridad	9
Restricciones de HW		• El sistema debe ser alimentado con energía de manera contínua durante el proceso de comprobación. • Los distintos componentes a controlarse deben estar correctamente conectados y configurados en el sistema para su comprobación. • Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.			
Restricciones de SW		• El sistema debe encontrarse en estado ~~INICIALIZA~~ACTIVO con estado interno OK.			

Esta segunda implementación generó la segunda versión del documento **R_ERS_01 Especificación de Requisitos Software - Monitor de Barreras**. El mismo se puede observar en el *Anexo R_ERS_01 Monitor de Barreras - Segunda implementación*; además se generaron los documentos *Anexo R_ESC_01 Monitor de Barreras - Segunda implementación y Anexo R_VRS_01 Monitor de Barreras - Segunda implementación*, con las actualizaciones correspondientes de los requerimientos.

4.1.7. Agregado de técnicas de métodos semi formales y formales al procedimiento

Luego de la segunda implementación del procedimiento con requerimientos de los stakeholders, se decidió analizar e implementar métodos formales al mismo, siguiendo las pautas dictadas por la norma EN-50128 y lo expuesto en el Capítulo 3.4.5.2 de este documento.

Se optó por realizar implementaciones de métodos semi formales y formales para los requerimientos más críticos del sistema a desarrollar, de acuerdo a su nivel de SIL, disminuyendo el nivel de abstracción de las especificaciones de manera progresiva (informal → semi formal → formal), hasta el punto que sea necesario para cada requerimiento, de acuerdo a este último valor. Para esto, se propuso el uso de:

- **Análisis semi formal**: mediante la generación de diagramas UML de clases (opcionales pero sugeridos) y de estados, utilizando la información obtenida de los análisis llevados a

cabo sobre los requerimientos especificados [51].
La manera de operar con esta metodología, idealmente inicia con la construcción de los diagramas de clases necesarios para representar los objetos que componen el problema a resolver, sus atributos, comportamientos y relaciones. Posteriormente, se deben crear los diagramas de estado necesarios para poder representar las transiciones de estados de estos objetos, pudiendo llegar a elaborarse esquemas con varios niveles de detalles, mediante el uso de, por ejemplo, sub-máquinas de estados.

- **Análisis formal**: mediante la generación de especificaciones formales utilizando el lenguaje ACSL. Esta decisión se tomó debido a que dicho lenguaje está pensado específicamente para proyectos programados en lenguaje de programación C (lenguaje a utilizar en la programación en este proyecto), y además permite generar casos de verificación de los requerimientos, a la vez que se realiza la especificación formal: es decir, permite cumplir con dos objetivos de una sola vez, como se puede apreciar en [52]. Para su implementación, primero se realiza el análisis de requerimientos del subsistema que se esté tratando, luego se realiza el diseño y especificación semi formal y la formal. Para esta última se toman como entradas los elementos generados mediante el análisis semi formal, así como toda la información que pueda servir de contexto en cada uno de los requerimientos (principalmente la obtenida en los análisis SFMEA llevados a cabo anteriormente). Luego de realizar la implementación (codificación) de la funcionalidad realizada, este modo de trabajo permite realizar inmediatamente después las pruebas unitarias, y poder pasar a realizar la integración de este nuevo subsistema, como se ve en la Fig. 26.

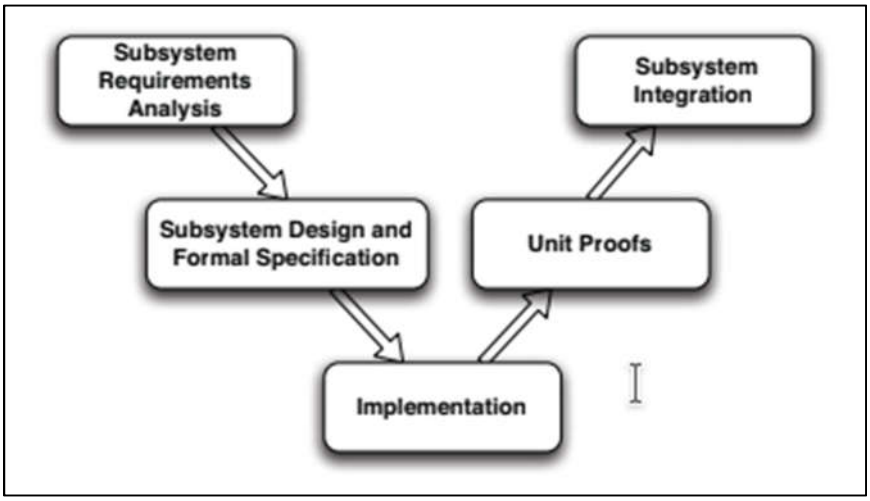


Fig. 26. Ciclo de vida con ACSL. Fuente: [52]

Se procedió entonces a incluir estas nuevas técnicas y cambios al procedimiento ya generado, obteniéndose la segunda versión (o incremento) del mismo. Los principales cambios que se realizaron fueron:

En el proceso de **Especificación**:

- Se modificó la restricción 7.2.4 a: "*Los requerimientos funcionales hacen referencia a cualidades o funcionalidades del software.*", indicando el rol de los mismos más claramente.
- Se modificó la restricción 7.2.6 a: "*Cuando sea necesario, se deben expresar los requerimientos mediante métodos formales o semi formales.*". Es decir, se agregaron los requerimientos semi formales al procedimiento.
- Se agregó la actividad 7: "*Realizar las especificaciones semi formales y formales.*", a cargo del gestor de requerimientos, incorporando estos nuevos cambios.
- Se agregó la restricción 7.2.18: "*En cuanto a las especificaciones formales y semi formales, se deberán llevar a cabo mediante lenguaje ACSL y diagramas de estado respectivamente.*".
- Se agregó la restricción 7.2.19: "*Para realizar la especificación semi formal, por cada requerimiento se deberá graficar su diagrama de estados, indicando como mínimo los distintos estados por los que puede pasar el sistema (dentro del contexto del requerimiento), y los eventos que los desencadenan, como se puede ver en la Figura 7.*

Alternativamente se pueden realizar diagramas de clases que ayuden a comprender el contexto del requerimiento y a la creación de su diagrama de estados correspondiente.". Los elementos del diagrama de estados se pueden ver en la Fig. 27.

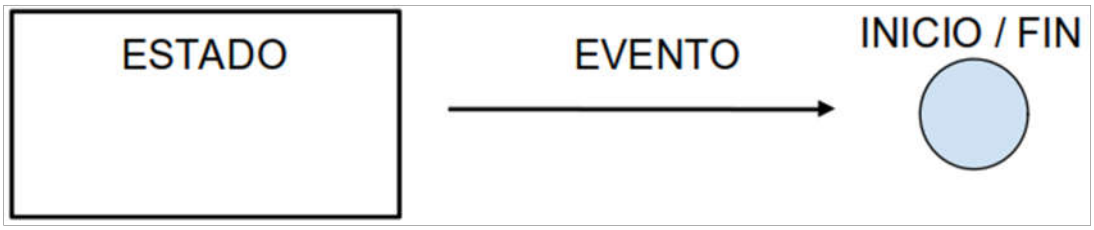


Fig. 27. Elementos del diagrama de estados

- Se agregó la restricción 7.2.20: "*Para realizar la especificación formal, por cada especificación semi formal se deberán crear los contratos de funciones en los comentarios del código utilizando el lenguaje de especificación formal ACSL, indicando como mínimo las pre y post condiciones de los mismos. Solamente se deberán crear los contratos de funciones, no las funciones en sí, ya que las mismas se crearán en una etapa posterior del desarrollo. Esto permitirá a su vez la realización de la verificación del código a generar.*"

  Un ejemplo del código ACSL a generar se puede ver en el siguiente contrato de función, en donde se especifica que las pre condiciones son que la variable A y la variable B sean dos enteros, y además se define la post condición, que indica que, al finalizar, la variable C deberá contener el número entero resultante de la suma de las dos anteriores:

```
/*@    requires variableA == (integer) && variableB == (integer);
       ensures variableC == (integer) variableA + variableB;
*/
```

En las **Entregas**:

- Se agregó el punto 1.i: "*Especificaciones semi formales y formales generadas con la Figura 7 y formales generadas en el lenguaje ACSL.*". Como se expresó anteriormente,

dichos análisis se deberán generar para los requerimientos dependiendo de su nivel de SIL.

Los cambios llevados a cabo en el procedimiento general, pueden ser vistos en la tercera versión de dicho procedimiento, disponible en el *Anexo Tercer versión del PG*, y el documento de registro de la ERS modificado en el *Anexo F_ERS_01 - Tercer versión*.

4.1.8 Tercera implementación del procedimiento con requerimientos reales

Luego de haber agregado al procedimiento las técnicas elegidas para el análisis semi formal y formal de los requerimientos más críticos, se procedió a realizar el análisis del primer requerimiento especificado (RQ-000001) a modo de prueba de su uso, generándose las especificaciones que se adjuntan a continuación para el mismo. Como aclaración, se decidió generar solamente las especificaciones del primer requerimiento debido a que era necesario instruir a los validadores en las técnicas utilizadas, e interiorizar más al especificador en el lenguaje ACSL, ya que él mismo era desconocido por ambas partes y presentaba un alto grado de complejidad: el tiempo insumido en la comprensión a profundidad de su utilización, y en la transferencia de conocimiento al validador hubiera retrasado la finalización del proyecto, incumpliendo con los plazos de tiempo planificados. Además, todos los requerimientos del monitor de barreras son de SIL 0, siendo alternativo el uso de estas técnicas.

Primero se generó el diagrama de clases del requerimiento, como se ve en la Fig. 28.

Para este caso en particular, se modelaron dos clases: *Monitor de barreras* y *Centro de control*, cada una con sus correspondientes atributos y métodos. Para identificar dichos objetos, se analizó la documentación generada hasta el momento para este requerimiento, siendo de especial importancia la generada mediante el análisis SFMEA, que permitió detectar posibles peligros, como ser el caso en que el sistema de comunicaciones utilizado no esté disponible, o que un componente hardware no se encuentre correctamente registrado en el sistema. Entonces, las salidas generadas por el uso de la técnica SFMEA sirvieron de entrada para la creación de este diagrama UML.

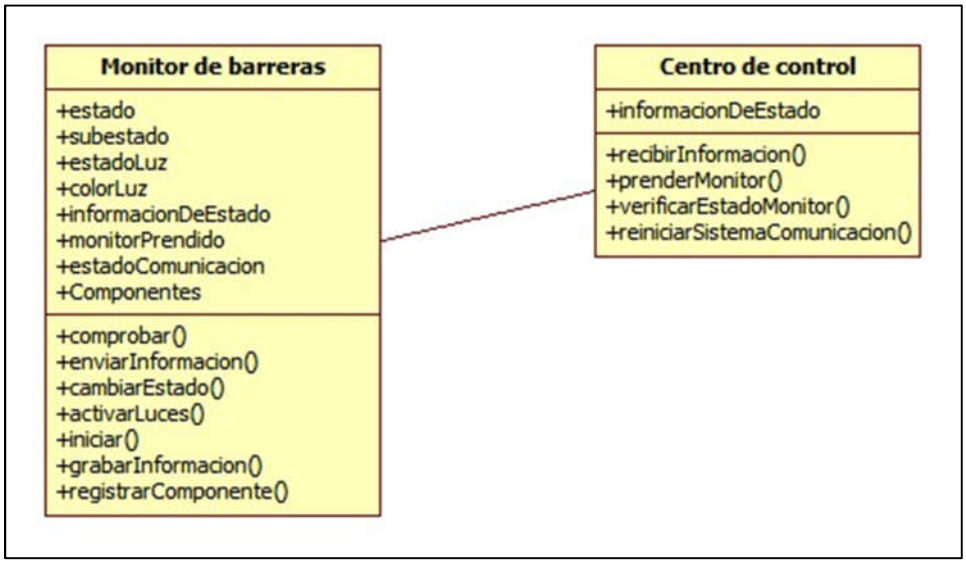


Fig. 28. Diagrama de clases

Luego de esto, y teniendo en cuenta el diagrama de estados informado por los stakeholders, los modos de comportamiento del software, los ensayos del software y en parte la información brindada por el árbol SFTA del requerimiento, se procedió a confeccionar su correspondiente diagrama de estados, como se ve en la Fig. 29.

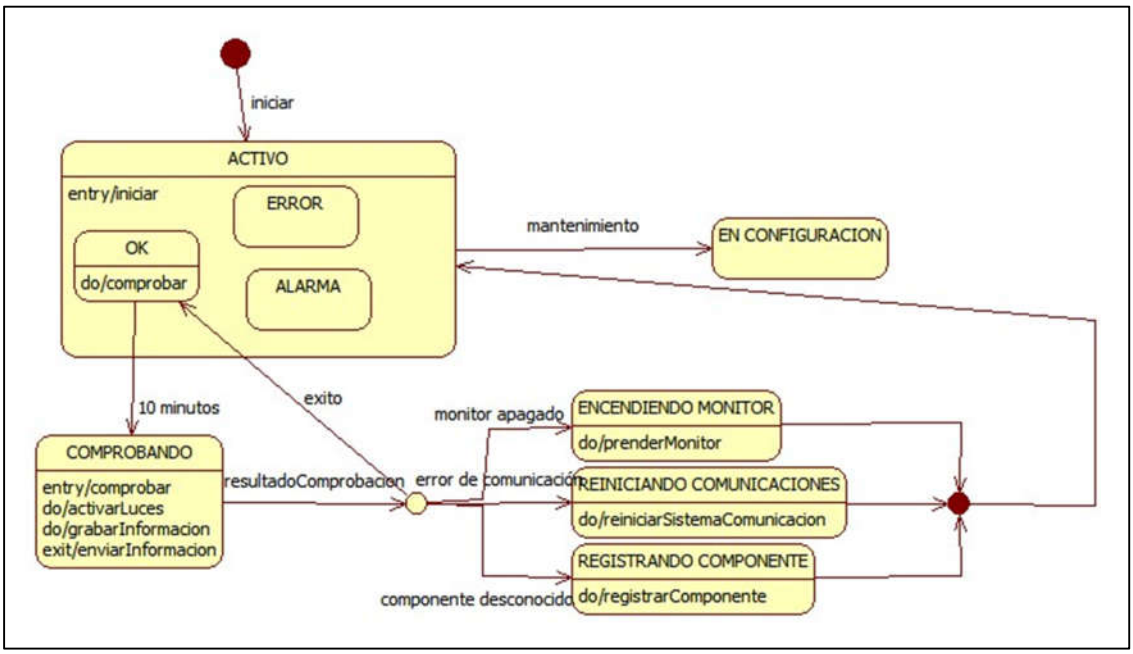


Fig. 29. Diagrama de estados

Se pueden apreciar seis estados en total, con sus correspondientes sub-estados cuando corresponde (los sub-estados del estado ACTIVO) y los eventos que provocan dichas transiciones entre los mismos. Los estados ENCENDIENDO MONITOR, REINICIANDO COMUNICACIONES y REGISTRANDO COMPONENTE hacen referencia a nuevos posibles "estados internos" del sistema, identificados mediante el análisis de seguridad SFMEA, que describen las acciones que debe realizar el mismo en caso de generarse alguna de las posibles fallas detectadas de acuerdo al resultado de la comprobación: monitor apagado, error de comunicación o componente desconocido. Realizar un análisis de seguridad más exhaustivo y detallado, de acuerdo al nivel de granularidad utilizada, podría provocar la aparición de nuevas posibles fallas que podrían también reflejarse en este diagrama.

Por último, con los resultados de los modelos UML de clases y de transición de estados, se procedió a generar la especificación formal en lenguaje ACSL, como se puede ver a continuación. Para esto se generaron los contratos de las funciones que serán necesarias de implementar a la hora de generar el código, describiendo siete en total. Las mismas se caracterizan principalmente por describir las pre y post-condiciones de las funciones al momento de generarse, es decir, las características que debe tener el sistema al momento previo a su ejecución, y como debe quedar al haber finalizado dicha ejecución. Los nombres propuestos para las funciones se pusieron como comentarios (entre los símbolos /* y */), ya que no es obligatorio que se llamen de esta manera, y se deberán definir concretamente al momento de la programación de las mismas. A continuación, se describen los contratos de funciones generadas de manera informal:

Primero, al estar apagado el sistema, el mismo no presenta ningún valor en la variable *estado*, como se describe en la cláusula *requires*. Al finalizar su inicialización, las variables deben quedar como se describen en la cláusula *ensures*.

```
/*@    requires estado == '';
 *   ensures estado == 'ACTIVO' && subestado == 'OK' && estadoLuz == 'on' && colorLuz = 'verde';
*/
```

```
/*iniciar()*/
```

Al realizarse la comprobación, el sistema debe estar en estado ACTIVO, con sub-estado OK. Pueden ocurrir dos cosas: que la comprobación se pase con éxito, quedando el sub-estado en OK, o que se produzca una falla. En este último caso, la misma puede deberse a que el monitor se encuentre apagado, se haya producido un error de comunicación o se haya monitoreado un componente desconocido (de acuerdo a lo encontrado al realizar el análisis SFMEA).

```
/*@    requires estado == 'ACTIVO' && subestado == 'OK';
   ensures estado == 'ACTIVO';
   behavior exito:
         assumes resultadoComprobacion = 'exito';
         ensures subestado == 'OK';
   behavior falla:
         assumes     resultadoComprobacion = 'monitor apagado' ||
                                 resultadoComprobacion = 'error de comunicacion' ||
                                 resultadoComprobacion = 'componente desconocido';
         ensures subestado == 'ERROR';
   complete behaviors exito, falla;
   disjoint behaviors exito, falla;
*/
```

```
/*comprobar()*/
```

Al activar las luces, el sistema debe encontrarse si o si en estado ACTIVO, y se debe comprobar que las luces se prendan. En caso de que el sub-estado del sistema sea OK, las luces deben ser de color verde, y en caso de encontrarse en sub-estado ERROR, las mismas deben ser de color rojo.

```
/*@    requires estado == 'ACTIVO';
```

```
    ensures estadoLuz = 'on';
    behavior OK:
        assumes subestado == 'OK';
        ensures colorLuz == 'verde';
    behavior ERROR:
        assumes subestado == 'ERROR';
        ensures colorLuz == 'rojo';
    complete behavior OK, ERROR;
    disjoint behavior OK, ERROR;
*/
/*activarLuces()*/
```

Al grabar la información temporal en memoria cache, el sistema debe encontrarse en estado ACTIVO y asegurarse de que la variable informacionDeEstado no esté vacía: debe haber grabado algo si o sí.

```
/*@ requires estado == 'ACTIVO';
    ensures informacionDeEstado != '';
*/
/*grabarInformacion()*/
```

Al enviar información el monitor de barreras (nivel monitor) al centro de control (nivel gestión), el sistema debe asegurarse de que la variable informacionDeEstado no esté vacía: debe enviar algo si o si.

```
/*@ requires informacionDeEstado != '';
    ensures informacionDeEstado == '';
*/
```

/*enviarInformacion()*/

Al prender el monitor de barreras, el mismo debe encontrarse obviamente apagado, y fuera del sub-estado error (bajo el cual no deberá prenderse). Al finalizar su inicialización, la variable *monitorPrendido* debe quedar instanciada en "si".

```
/*@ requires informacionDeEstado.monitorPrendido == 'no' && subestado != 'ERROR';
   ensures monitorPrendido == 'si';
*/
```

/*prenderMonitor()*/

Al ejecutarse el procedimiento de contingencia de reinicio del sistema de comunicación, el estado de la comunicación debe ser erróneo, y el sistema debe presentar sub-estado ERROR. Al finalizar dicho reinicio, el estado del sistema de comunicación debe ser correcto.

```
/*@ requires informacionDeEstado.estadoComunicacion == 'erroneo' && subestado == 'ERROR';
   ensures estadoComunicacion == 'correcto';
*/
```

/*reiniciarSistemaComunicacion()*/

Al ejecutar el procedimiento de contingencia de registración de componentes, se debe comprobar que la variable componentesRegistrados posea el valor "no", que indicará que al menos uno de los mismos no se encuentra correctamente registrado, y el sub-estado del monitor de barreras deberá ser ERROR. El procedimiento deberá comprobar, para asegurar que el o los componentes faltantes se han registrado, que el tamaño de la estructura de datos (arreglo) Componentes actual sea mayor que el que poseía antes de finalizar la función, indicando esto que el sistema posee registrado al menos un componente más que antes de ejecutarse.

```
/*@ requires informacionDeEstado.componentesRegistrados == 'no' && subestado == 'ERROR';
   ensures sizeof(\old(Componentes)) < sizeof(Componentes);
```

```
*/

/*registrarComponente()*/
```

Esta tercera implementación generó la tercera versión del documento **R_ERS_01 Especificación de Requisitos Software - Monitor de Barreras**. El mismo se puede observar en el *Anexo R_ERS_01 Monitor de Barreras - Tercer implementación*. Cabe aclarar que los otros dos documentos generados por el procedimiento no sufrieron modificaciones, y que los requerimientos fueron correctamente validados con los stakeholders, como se puede ver en el documento *Anexo R_VRS_01 Monitor de Barreras - Segunda implementación*, obteniendo la aprobación de los mismos.

4.1.9 Integración del procedimiento con Eclipse Process Framework (EPF)

Con la herramienta Eclipse Process Framework (EPF) [53] se realizó la integración del procedimiento desarrollado con todos sus procesos, dejándolo disponible al público, mediante la página web [54]. Esta actividad estuvo a cargo del Ing. Roque Ortega, integrante del equipo de proyectos del GICSAFe.

El EPF es un framework de código abierto, diseñado especialmente para la ingeniería de procesos del software, que brinda la posibilidad de realizar la gestión de nuevos procesos y procedimientos, permitiendo el desarrollo evolutivo, iterativo e incremental, la gestión documental, la reusabilidad, entre otras funcionalidades.

Con esta herramienta se lograron unificar e integrar todos los procedimientos de gestión de calidad que se crearon en el proyecto, permitiendo su mejor identificación y navegación a través de los mismos, mediante una interfaz gráfica amigable para el usuario de estos, como se puede ver en la Fig. 30.

El procedimiento de Gestión de Requisitos del Software puede ser observado integrado al EPF en la Fig. 31, con cada uno de los formularios que genera.

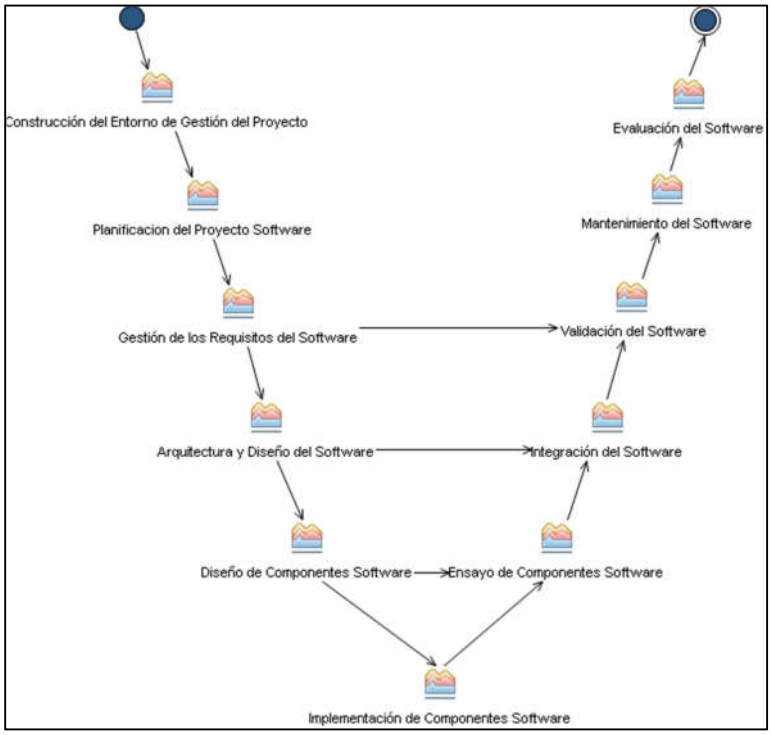


Fig. 30. Ciclo de vida en V del proyecto

Fig. 31. Gestión de Requisitos Software

Al ingresar, por ejemplo, a la actividad de Especificación de Requisitos del Software, la herramienta muestra los procesos que la componen, y sus relaciones, como se puede ver en la Fig. 32: Obtención, Especificación y Verificación. Al ingresar a cada uno de estos, el sistema muestra la descripción de los mismos, tal como se los diseñó, indicando el orden de ejecución de cada una de sus actividades.

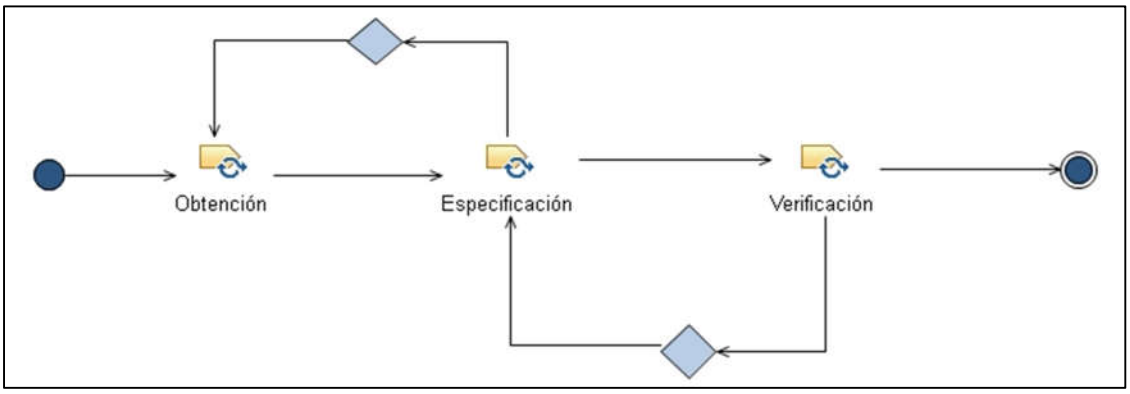


Fig. 32. Especificación de Requisitos del Software

La herramienta EPF fue integrada al resto de herramientas que componen al ecosistema del proyecto [55], como se puede observar en la Fig. 33. Se encuentra ubicada en el subsistema lógico de gestión y modelado, que instrumenta la gestión documental, la planificación de las tareas para los proyectos, el seguimiento y control, los reportes y la trazabilidad, e incluye también las herramientas de modelado y diseño. El rol del EPF dentro de este subsistema es el de permitir construir la metodología de trabajo según la norma EN-50128 para que sea accesible y adaptable. Permite entonces la construcción del Sistema de Gestión de Calidad, de los procedimientos generales y la comunicación de estos últimos, los formularios y las guías que se desarrollen. Sus relaciones con las otras herramientas son:

- **Redmine**: el inicio de un nuevo proyecto siguiendo el procedimiento descrito en el EPF crea un arquetipo de un proyecto Redmine con las tareas precargadas. Su mecanismo de contacto se da por medio de publicaciones Web.
- **Git**: el desarrollo colaborativo de los procedimientos es gestionados mediante el control de versiones Git. El mecanismo de contacto entre ambas herramientas se da mediante el cliente SCM de Git [56].

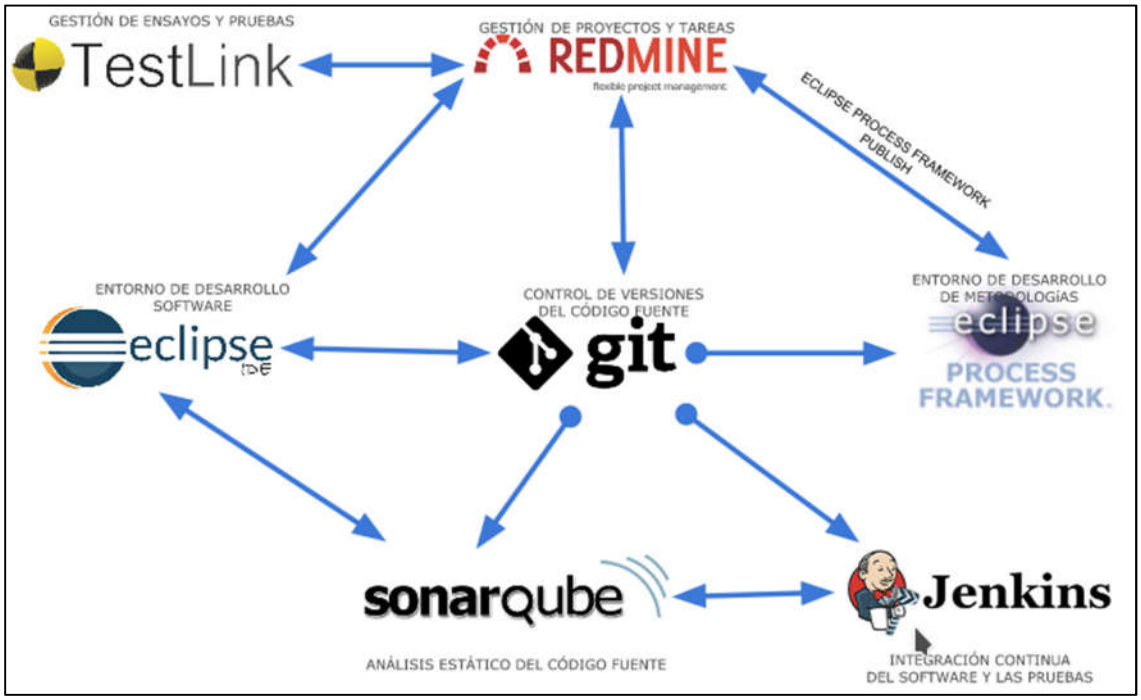


Fig. 33. Ecosistema de herramientas del proyecto

4.2. Verificación

Se ha realizado la comprobación del cumplimiento del capítulo **7.2 Requisitos del Software**, de la norma EN-50128, de los requerimientos especificados mediante el procedimiento, a modo de evidencia de uso del procedimiento, que, como se indicó anteriormente mediante la tabla Cumplimiento de EN-50128:2012, cumple con los puntos indicados en dicha norma. A modo de ejemplo, se presenta la verificación de cumplimiento del requerimiento con el que se venían ilustrando los capítulos anteriores, el RQ-000001, en la Tabla 49.

Tabla 49. RQ-000001 - Verificación de cumplimiento con EN-50128

Ítem de UNE-EN 50128	Elemento del procedimiento definido
7.2.1.1	El RQ ha pasado por los procesos de Obtención y Especificación. Para su tratamiento se han tenido en cuenta los aspectos definidos en los documentos: ● PG_REQ_10 Requisitos del sistema ● PG_PSG_10 Plan de seguridad, Se ha generado la documentación correspondiente.
7.2.1.2	Se ha descrito la Especificación de Ensayos del Software en Conjunto del RQ, generando el documento Anexo R_ESC_01 Monitor de Barreras - Segunda implementación.
7.2.2	Se ha analizado la documentación de entrada definida a la hora de realizar el proceso de Obtención.
7.2.3	Se han generado los documentos pertinentes, con la información del RQ en ellos: ● Anexo R_ERS_01 Monitor de Barreras - Tercer implementación ● Anexo R_ESC_01 Monitor de Barreras - Segunda implementación ● Anexo R_VRS_01 Monitor de Barreras - Segunda implementación
7.2.4.1	Se ha redactado el documento Anexo R_ERS_01 Monitor de Barreras - Tercer implementación bajo la responsabilidad del Gestor de Requisitos.
7.2.4.2	El RQ ha sido definido como del tipo Funcional, indicando la funcionalidad y la seguridad.
7.2.4.3	Todos los RQ del proyecto DIMBA son SIL 0, aunque fueron tratados como si fueran SIL 2, para dotarlos de mayor seguridad y por tratarse de un proyecto piloto.
7.2.4.4 a	La especificación del RQ ha sido definida de una forma en que sea completa, clara, precisa, inequívoca, verificable, que se pueda someter a ensayo, que se pueda mantener y sea realizable. Esto se puede comprobar posteriormente en
7.2.4.4 b	El identificador RQ-000001 permite trazar al mismo por todos los documentos generados por el uso del procedimiento.
7.2.4.5	Se generó el Glosario, como se puede ver en el documento Anexo R_ERS_01 Monitor de Barreras - Tercer implementación.
7.2.4.6	Se ha analizado la documentación de Especificaciones de la Interfaz Externa a la hora de especificar el RQ, y tenido en cuenta para realizar el análisis operacional y sistémico del mismo.
7.2.4.7	Se han indicado los modos de funcionamiento del RQ mediante sus respectivos análisis SSTA y SFTA.
7.2.4.8	Se han detallado los modos de comportamiento de fallo mediante el análisis SFMEA del RQ.
7.2.4.9	Se han indicado las restricciones a nivel HW y SW a tener en cuenta a la hora de resolver el RQ.
7.2.4.10	El RQ tratado es parte del conjunto de RQ's que realizan las comprobaciones al HW y al SW, tanto de la barrera automática como las del mismo monitor (autocomprobación de SW).

7.2.4.11	El RQ tratado no es de tipo no funcional, ni de subtipo ensayo, pero se han definido los ensayos de SW a realizar sobre el mismo y se han verificado dichas definiciones.
7.2.4.12	Se han definido los ensayos de SW a realizar sobre el RQ y se han verificado dichas definiciones.
7.2.4.13	Se han definido las funciones que se necesita que realice el SW, indicando también si las mismas cumplen con el nivel de seguridad deseado. Se recuerda que dicho nivel SIL es cero para todos los RQ's del proyecto DIMBA, pero se los trata como de nivel 2.
7.2.4.14	El RQ está relacionado con la seguridad, por lo que no es representativo para el caso. En caso de tratarse de un RQ no relacionado con la seguridad, es totalmente factible especificarlo mediante el procedimiento definido.
7.2.4.15	El RQ ha sido especificado de manera informal, semi formal y formal.
7.2.4.16	Se ha redactado el documento de Especificación de Ensayos del Software en Conjunto, en donde se encuentra incluido el mismo.
7.2.4.17	La Especificación de Ensayos del Software en Conjunto es una descripción de los ensayos a realizar en el software terminado.
7.2.4.18	Se ha definido el tipo de ensayo "funcional" para el RQ.
7.2.4.19	Se han especificado las señales de entrada, de salida y los criterios de éxito para los ensayos del RQ.
7.2.4.20	Se ha redactado el Informe de Verificación de los Requisitos del Software
7.2.4.21	Se identificaron correctamente la identidad y configuración del RQ verificado, así como los nombres de los verificadores, entre los demás datos necesarios.
7.2.4.22	Se establecieron correctamente los indicadores de adecuación de cumplimiento de la documentación necesaria, la legibilidad y trazabilidad de la misma, la definición de los ensayos a realizar en el SW, la coherencia interna de la especificación y la definición y cumplimiento de restricciones de HW/SW para el RQ verificado.

Se puede decir entonces que, en definitiva, el requerimiento propuesto a modo de ejemplo cumple con lo expuesto en la norma EN-50128. Los requerimientos restantes tuvieron un tratamiento similar a este, por lo que también se encuentran en condiciones de ser catalogados como verificados.

Capítulo 5

Conclusiones y trabajos futuros

5. Conclusiones y trabajos futuros

5.1. Conclusiones

En este capítulo se analiza el cumplimiento de los objetivos establecidos en el Capítulo 1 a lo largo de todo el trabajo. Además, se informan y analizan los artículos publicados a lo largo de su desarrollo, que brindan un mayor sustento académico a la misma.

5.1.1. Conclusiones generales

Se desarrolló un procedimiento de gestión de requerimientos software para sistemas críticos ferroviarios, analizando el estado del arte y el dominio del problema, por medio de la realización de una RSL y la investigación de la problemática a nivel regional. El procedimiento fue creado teniendo en cuenta las buenas prácticas definidas por la norma EN-50128, específicamente en el capítulo 7.2. Requisitos del Software, y se verificó su cumplimiento con la misma por medio del uso de una tabla de verificación. Para lograr el cumplimiento de estos puntos, se añadieron técnicas de seguridad y el uso de métodos semi formales y formales de especificación de requerimientos al mismo, características que se fueron mejorando de manera iterativa e incremental. El procedimiento fue integrado a un ecosistema de gestión de calidad en sistemas ferroviarios, conformado por el grupo de investigación GICSAFe, bajo las normas ISO 9001 y la EN 50126, y se lo adaptó al mismo y vinculó con las demás herramientas que lo componen por medio del software EPF. El procedimiento generado se puso a prueba mediante su uso en un proyecto piloto: el proyecto DIMBA; en este, se gestionaron requerimientos software reales solicitados por los miembros de SOFSE y se generó la documentación pertinente, mediante el uso de formularios, quedando sus registros en el sistema de gestión de calidad.

La concreción de este procedimiento permite visualizar la posibilidad de creación de herramientas de gestión de calidad de sistemas críticos ferroviarios que cumplan con normas específicas dentro del país, brindando apoyo al desarrollo de tecnologías de manera local, que de otra manera son adquiridas desde el exterior. A su vez, demuestra la importancia del trabajo en equipo para la realización de proyectos de tal envergadura y complejidad, mediante una forma de trabajo bien definida, estructurada y organizada.

5.1.2. Conclusiones específicas

En el Capítulo 1 de este trabajo se presentaron una serie de objetivos específicos (parciales), necesarios para cumplir el objetivo general de la misma: *"... desarrollar un procedimiento de gestión de requerimientos software para sistemas críticos ferroviarios, bajo la normativa EN-50128, que pueda ser aplicado a un ecosistema de gestión de calidad en sistemas ferroviarios, en el marco del Grupo de Investigación en Calidad y Seguridad de las Aplicaciones Ferroviarias (GICSAFe), del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) de la República Argentina. ..."*. A continuación, se analiza el cumplimiento de cada uno de estos objetivos específicos (OE), para los que, de haberse cumplido, se cumpliría el objetivo general.

- **OE1**: *Investigar y evaluar los métodos y las herramientas utilizadas actualmente a nivel mundial para la gestión de requerimientos software en sistemas críticos ferroviarios. En esta investigación se pretenden descubrir qué tipos o módulos de sistemas ferroviarios se implementan generalmente y por qué, qué metodologías y herramientas tecnológicas son usadas en este proceso, como se trata el concepto de seguridad y que normativas son las que se intentan cumplir.*

  Para la concreción de este objetivo, se realizaron: análisis de información de distintas fuentes altamente relevantes en el tema y una revisión sistemática de la literatura, presentes en el Capítulo 4 de este Trabajo Final de Maestría. Con esto, se lograron responder a las interrogantes planteadas en el OE1 mediante cinco preguntas de investigación, permitiendo obtener la información necesaria con respecto a la integración de los distintos aspectos tratados en distintos proyectos a nivel global, las formas de implementación de este tipo de procedimientos y los resultados alcanzados por los mismos. Dichas preguntas fueron:

  - **PI-1**: ¿Qué tipos o módulos de sistemas software se implementan siguiendo metodologías de gestión de requerimientos software en sistemas críticos ferroviarios?
  - **PI-2**¿Qué metodologías se utilizan para la gestión de requerimientos software en

sistemas críticos ferroviarios?

- **PI-3**¿Qué software o herramientas se utilizan y cómo se logra la integración de las mismas en la gestión de requerimientos software en sistemas críticos ferroviarios?
- **PI-4**¿Cómo se gestiona la seguridad de los sistemas de gestión de requerimientos software en sistemas críticos ferroviarios?
- **PI-5**¿Bajo qué normativas se implementan sistemas de gestión de requerimientos software en sistemas críticos ferroviarios?

Todo esto se utilizó para ser volcado dentro del procedimiento a generar, generando una base de conocimiento sólida acerca del estado de la cuestión actual, información que no existía hasta el momento. Además, se realizó un análisis del contexto del proyecto en el que se encuentra enmarcado este trabajo, con el fin de lograr seleccionar las técnicas, metodologías y herramientas más apropiadas, y acordes con dicho contexto del proyecto de gestión de la calidad, como también se indica en el Capítulo 4.1.1.

- **OE2**: *Desarrollar un procedimiento de gestión de requerimientos software en sistemas críticos ferroviarios que se adapte a la normativa EN-50128, y a niveles generales con las buenas prácticas de la EN-50126, e incluirlo en el proyecto Computadora Industrial Abierta Argentina (CIAA), del GICSAFe. El mismo deberá constar de actividades que garanticen la seguridad y calidad del software a desarrollar, aplicando las técnicas, metodologías y herramientas obtenidas con los resultados del cumplimiento del OE1.* Se desarrolló un procedimiento, tal como se propuso, teniendo en cuenta principalmente el cumplimiento de los puntos indicados en el capítulo 7.2 de la norma EN-50128:2012. El mismo se adaptó al entorno de gestión de calidad realizado por el grupo de investigación GICSAFe, teniendo en cuenta las pautas indicadas por la norma EN-50126 y la ISO 9001, logrando su inclusión dentro del proyecto mencionado. El procedimiento fue plasmado en el formato de tres documentos, que se indican en el Capítulo 4.1.2 de este Trabajo Final de Maestría, que son necesarios completar para cada proyecto que se quiera realizar bajo este marco de trabajo. Los mismos son:
  - **Especificación de Requisitos Software (ERS)**

- **Especificación de Ensayos del Software en Conjunto (ESC)**
- **Informe de Validación y verificación de los Requisitos del Software (VRS)**

Además, se controló el cumplimiento con la sección indicada de la norma mediante el cuadro que se informa en el Capítulo 4.1.3 y lo expuesto en el Capítulo 4.2. La integración de esta documentación con el resto del proyecto se llevó a cabo mediante la herramienta Eclipse Process Framework (EPF), como se indica en el capítulo 4.1.9 de este documento.

En cuanto a los resultados obtenidos en el OE1, se aplicaron:

- **Técnicas**: técnicas de seguridad SFTA, SSTA y SFMEA, como se indica en el Capítulo 4.1.5.
- **Metodologías**: métodos estructurados (el procedimiento en sí), semi formales (diagramas UML: de clases y de estados) y formales (lenguaje ACSL), como se indica en el Capítulo 4.1.7.
- **Herramientas**: Eclipse Process Framework, para la gestión documental de la calidad, como se indica en el Capítulo 4.1.9.

- **OE3**: *Aplicar el procedimiento desarrollado a un proyecto piloto, en el diseño de un monitor de barreras ferroviarias para la Autoridad Ferroviaria Nacional, enmarcado dentro de los límites del sistema de gestión de calidad, con la finalidad de probar su correctitud y madurez, dentro de un proceso de mejora continua. La implementación del procedimiento deberá ser validada por los stakeholders, con lo que se espera lograr un proceso de retroalimentación que permita la mejora de dicho procedimiento.*
  Se generaron los correspondientes registros de aplicación de los documentos informados en el punto anterior, como se indica en el Capítulo 4.1.4, luego de la primera educción de requerimientos a los stakeholders, correspondiente al proyecto piloto del monitor de barreras. Los correspondientes documentos fueron:
  - **R_ERS_01 Especificación de Requisitos Software - Monitor de Barreras**.
  - **R_ESC_01 Especificación de Ensayos del Software en Conjunto - Monitor de Barreras**.
  - **R_VRS_01 Informe de Verificación de los Requisitos del Software - Monitor**

de Barreras.

Luego de esto, se aplicaron las técnicas de seguridad definidas sobre los requerimientos existentes, y se obtuvieron nuevos, como se indica en el Capítulo 4.1.6. Por último, se aplicaron métodos semi formales y formales sobre uno de los requerimientos, como se indica en el Capítulo 4.1.8, y se validaron correctamente todos los requerimientos, como se puede ver en el documento *Anexo R_VRS_01 Monitor de Barreras - Segunda implementación*, obteniendo la conformidad de los stakeholders con los mismos.

5.1.3. Publicaciones realizadas

Como resultados parciales de la concreción del presente Trabajo Final de Maestría, se han publicado tres artículos en congresos nacionales, los cuales se informan a continuación:

Autores: Irrazábal E., Bernal R., **Pinto Luft C.**, Sambrana I.

Título: Ingeniería de software para sistemas críticos ferroviarios

Congreso: XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018)

Publicación: XX Workshop de Investigadores en Ciencias de la Computación - Libro de actas - pp. 563-567.

ISBN: 8-987-3619-27-4

Lugar de celebración: Corrientes (Argentina)

Fecha: 26 a 27 de abril de 2018

Autores: Irrazábal E., Sambrana I., **Pinto Luft C.**, Gómez López M. de los Á., Gallina S. H., Laiuppa A., Brizuela J., Gómez P., Lutenberg A.

Título: Metodología de desarrollo de aplicaciones ferroviarias según las normas ISO 9001 - EN-50126

Congreso: Congreso Argentino de Sistemas Embebidos (CASE 2018)

Publicación: Congreso Argentino de sistemas Embebidos CASE 2018 - libro de trabajos en modalidad foro tecnológico y resumen - pp. 84-89.

ISBN: 978-987-46297-5-3

Lugar de celebración: Buenos Aires (Argentina)

Fecha: 15 a 17 de agosto de 2018

Autores: **Pinto Luft C.**, Irrazábal E., Sambrana I.

Título: Revisión Sistemática de la Literatura: aplicación de seguridad a requerimientos software de sistemas críticos ferroviarios

Congreso: Jornadas Argentinas de Informática 2018 (47 JAIIO) - Simposio Argentino de Ingeniería de Software (ASSE 2018)

Publicación: Proceedings of ASSE 2018 Argentine Symposium on Software Engineering - pp. 37-48.

ISSN: 2451-7593

Lugar de celebración: Buenos Aires (Argentina)

Fecha: 3 a 7 de septiembre de 2018

5.2. Futuras líneas de investigación

A lo largo del desarrollo del proyecto se detectaron oportunidades de mejora en cada iteración realizada, en algunos de los aspectos que conciernen al procedimiento, que por cuestiones de tiempo y/o complejidad no pudieron ser incluidas en el mismo. Las mismas son catalogadas como mejoras a integrar, y se listan a continuación.

5.2.1. Integración con la comunidad

Se ha detectado, que por más que se haya realizado una RSL para la obtención de información acerca del tema, hay conceptos muy propios del diseño de sistemas críticos ferroviarios que

podrían comprenderse mejor integrándose aún más con la comunidad que lo compone; esto es, con los autores de los principales artículos seleccionados en dicha RSL, y con el equipo de trabajo mismo del proyecto. Esto enriquecería mucho más el conocimiento que se tiene sobre el dominio del problema, llevando a tomar mejores decisiones de diseño. Esta integración podría realizarse, de acuerdo a la persona o al equipo de trabajo, mediante distintos medios: estableciendo reuniones con cierta periodicidad, consultando mediante envío de emails a referentes que vivan en zonas geográficas distantes, uniéndose a grupos de discusión y debate, entre otros.

5.2.2. Mejora de SFTA y SFMEA

Los métodos de análisis SFTA (SSTA') y SFMEA se aplicaron de una manera "reducida" durante el desarrollo del proyecto, sin hacer uso de todas sus propiedades de una manera exhaustiva, lo cual disminuye sus utilidades y no permite sacar provecho por completo de los mismos.

Por ejemplo, para el análisis SFTA se pueden utilizar más compuertas lógicas que las AND y las OR, más tipos de eventos, definir los minimal cut sets (conjuntos de corte mínimo, que indican el mínimo conjunto de eventos iniciales que se deben producir para que se produzca el fallo en el árbol), los path sets (conjunto de caminos, complemento de los cut sets) y también se pueden aplicar métodos probabilísticos, con tal de conocer la probabilidad de que ocurra cada cadena de acciones que puedan conducir a una falla.

Para el análisis SFMEA se pueden agregar distintos campos más a la tabla que se genera, de acuerdo a la necesidad de analizar alguna dimensión más por parte del equipo, o extenderlo a SFMECA, agregando la dimensión de criticalidad al análisis. Para ambos métodos se puede generar, luego de su representación gráfica, sus correspondientes expresiones en notación matemática, lo que permite el análisis automatizado y la generación de información por medio de programas informáticos. Además, se pueden generar reportes de los resultados arrojados por ambos tipos de análisis, simplificando la comprensión para los analistas y demás miembros del equipo, de modo que los mismos puedan tomar las medidas necesarias para paliar los problemas detectados.

5.2.3. Frama-C/ACSL

El lenguaje ACSL que se decidió utilizar como lenguaje de especificación formal dentro del procedimiento, puede ser utilizado en conjunto con la plataforma Framework for Modular Analysis of C programs (Frama-C) [57]. Esta es una plataforma utilizada para realizar análisis estático de código escrito en C, que permite, entre otras cosas, verificar que el código fuente cumpla con lo especificado en el lenguaje formal ACSL de manera automatizada, más rápida y menos riesgosa que una revisión de código. Esta es una de las características importantes de ACSL, ya que permite entonces realizar la especificación formal de los requerimientos (como se indica en el Capítulo 4.1.8) y luego comparar el cumplimiento del código generado con dicha especificación. Por lo que, se espera a futuro poder incluir la plataforma Frama-C al proyecto. Además, se espera poder seguir profundizando en el lenguaje ACSL, sus operadores, semántica, estructura y uso, con el fin de obtener más conocimientos del mismo y poder realizar especificaciones formales más completas para cada requerimiento.

5.2.4. Validación

A lo largo del ciclo de vida del proyecto se fueron realizando verificaciones parciales del mismo, analizando su cumplimiento con la norma EN-50128, y se validaron los requerimientos educidos con los stakeholders luego de la utilización de dicho procedimiento en un caso real (el monitor de barreras). Como futura mejora se espera poder realizar la validación y evaluación completa del procedimiento en sí, ya sea por parte de un par perteneciente al equipo de trabajo, con amplio conocimiento sobre la materia, o por un auditor formal, capaz de evaluar el cumplimiento del procedimiento con la normativa referida.

Bibliografía

[1] Estadísticas de la Comisión Nacional Reguladora del Transporte. Website, última visita 19/02/2018. https://www.argentina.gob.ar/cnrt/estadisticas-ferroviarias

[2] J. L. Arques Patón, "Ingeniería y gestión del mantenimiento en el sector ferroviario", 1ra Edición, España: Ediciones Díaz de Santos, 2009, ISBN 8479789166.

[3] Ejemplos de accidentes ferroviarios argentinos. Website, última visita 19/02/2018. https://es.wikipedia.org/wiki/Categor%C3%ADa:Accidentes_ferroviarios_en_Argentina.

[4] Ejemplo de licitación. Website, última visita 19/02/2018. https://www.clarin.com/ciudades/trenes-china-compra-licitacion-reestatizacion_0_rJduN7cwXe.html

[5] Compra de trenes a China. Website, última visita 19/02/2018. https://www.clarin.com/ieco/china-trenes_de_carga-randazzo-inversiones_0_rJygK8mKP7l.html

[6] Compra de trenes a Japón. Website, última visita 19/02/2018. https://www.argentina.gob.ar/noticias/japon-comenzara-fabricar-la-tecnologia-para-el-frenado-automatico-de-trenes

[7] Proyecto CIAA. Website, última visita 19/02/2018. http://www.proyecto-ciaa.com.ar

[8] CONICET-GICSAFe. Website, última visita 15/11/2018. https://sites.google.com/view/conicet-gicsafe

[9] UNE EN-50128, Aplicaciones ferroviarias. Sistemas de comunicación, señalización y procesamiento. Software para sistemas de control y protección del ferrocarril, 2012.

[10] R. S. Pressman, "Ingeniería del Software, Un Enfoque Práctico", 7ma Edición, España: McGraw-Hill Companies, 2010, ISBN 978-607-15-0314-5

[11] Esperanza, M., Marcos, A., "An Aristotelian Approach to the Methodological Research: a Method for Data Models Construction", Information Systems - The Next

Generation, edición de L. Brooks and C. Kimble, McGraw-Hill, 1998.

[12] Bunge, M., "La Investigación Científica", 1ra Edición, Barcelona: Ariel, 1979, ISBN 8434480107.

[13] J. L. Boulanger, "CENELEC 50128 and IEC 62279 Standards", Control, Systems and Industrial Engineering Series, John Wiley & Sons, Inc., 2015, p. 13.

[14] EN 50126. Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS). 2005.

[15] EN 50129. Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signaling. 2005.

[16] J. Vilela, J. Castro, L. E. G. Martins, T. Gorschek, “Integration between Requirements Engineering and Safety Analysis: A Systematic Literature Review”, The Journal of Systems & Software, Vol. 125, Pp. 68-92, Marzo, 2017.

[17] NASA Software Safety Guidebook. NASA Technical Standard. NASA-GB-8719.13. Marzo, 2004.

[18] Ansaldo STS. Website, última visita 11/07/2017. http://www.ansaldo-sts.com/en/index

[19] Siemens Rail Transportation. Website, última visita 11/07/2017. http://w3.siemens.com/mcms/industrial-controls/en/railway/Pages/overview.aspx

[20] CONICET. Website, última visita 15/11/2018. https://www.conicet.gov.ar/

[21] CNEA. Website, última visita 15/11/2018. https://www.argentina.gob.ar/comision-nacional-de-energia-atomica

[22] UBA. Website, última visita 15/11/2018. http://www.uba.ar/

[23] UNCA. Website, última visita 15/11/2018. http://www.unca.edu.ar/

[24] UNNE. Website, última visita 15/11/2018. http://www.unne.edu.ar/

[25] UNT. Website, última visita 15/11/2018. https://www.unt.edu.ar/

[26] UTN-FRBB. Website, última visita 15/11/2018. https://www.frbb.utn.edu.ar/

[27] UTN-FRH. Website, última visita 15/11/2018. http://www.frh.utn.edu.ar/

[28] ISO-9001. Sistemas de gestión de calidad - requisitos. 2015.

[29] SOFSE. Website, última visita 15/11/2018. https://www.argentina.gob.ar/transporte/trenes-argentinos

[30] Trenes Argentinos. Website, última visita 15/11/2018. https://www.argentina.gob.ar/transporte/trenes

[31] Eureka, desafio de ideas. Website, última visita 15/11/2018. http://www.desafioeureka.com/

[32] INNOVAR, concurso nacional de innovaciones. Website, última visita 15/11/2018. https://mia.gob.ar/convocatorias/innovar

[33] 1er Encuentro Intersectorial Ferroviario - Nueva Gestión de lo Público. Website, última visita 15/11/2018. https://institutoi4.net/1o-encuentro-intersectorial-ferroviario-nueva-gestion-de-lo-publico/

[34] CIAA-NXP. Website, última visita 15/11/2018. http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:hardware:ciaa_nxp:ciaa_nxp_inicio

[35] Datos en tiempo real del monitor de barreras. Website, última visita 15/11/2018. https://sites.google.com/view/conicet-gicsafe/inicio/resultados-obtenidos/datos-en-tiempo-real

[36] B. Malone, A. Siraj, “Tracking requirements and threats for secure software development”, ACM-SE 46 Proceedings of the 46th Annual Southeast Regional Conference on XX, Pp. 278-281, Auburn, Alabama, Marzo 28 - 29, 2008, ISBN: 978-1-60558-105-7.

[37] H. Hwang and Y. B. Park, "Safety - Critical Software Quality Improvement Using Requirement Analysis," 2017 International Conference on Platform Technology and Service (PlatCon), Busan, 2017, pp. 1-4. doi: 10.1109/PlatCon.2017.7883725.

[38] S. Li, S. Duo, "Safety Analysis of Software Requirements: Model and Process", Procedia Engineering, Vol. 80, Pp. 153-164, 2014, ISSN 1877-7058.

[39] A. Saeed, R. de Lemos, T. Anderson, "On the safety analysis of requirements specifications for safety-critical software", ISA Transactions, Vol. 34, Issue 3, Pp. 283-295, 1995, ISSN 0019-0578.

[40] A.P. Ravn, H. Rischel, V. Stavridou, "Provably Correct Safety Critical Software", IFAC Proceedings Volumes, Vol. 23, Issue 6, Pp. 13-18, 1990, ISSN 1474-6670.

[41] R. Shaw, "Safety-critical software and current standards initiatives", Computer Methods and Programs in Biomedicine, Vol. 44, Issue 1, Pp. 5-22, 1994, ISSN 0169-2607.

[42] M. J. Squair, "Issues in the application of software safety standards", SCS '05 Proceedings of the 10th Australian workshop on Safety critical systems and software - Vol. 55, Pp. 13-26, Sydney, Australia, 2006, ISBN:1-920-68237-6.

[43] S. Tiwari, S. S. Rathore, S. Gupta, V. Gogate, A. Gupta, "Analysis of Use Case Requirements Using SFTA and SFMEA Techniques", ICECCS '12 Proceedings of the 2012 IEEE 17th International Conference on Engineering of Complex Computer Systems, Pp. 29-38, Julio 18 - 20, 2012, ISBN: 978-2-9541-8100-4.

[44] P. J. Bryan, "Software safety and dependability for railway control systems," IET 13th Professional Development Course on Electric Traction Systems, London, 2014, pp. 1-21. doi: 10.1049/cp.2014.1445. ASSE, Simposio Argentino de Ingeniería de Software

47JAIIO - ASSE - ISSN: 2451-7593 - Página 46

[45] J. Brummer, M. Kersken, J. Märtz, "Tools for software safety analysis", Reliability Engineering & System Safety, Vol. 46, Issue 2, Pp. 123-138, 1994, ISSN 0951-8320.

[46] Kitchenham, B., "Procedures for Performing Systematic Reviews", Technical Report

TR/SE-0401, Keele University (UK), 2004.

[47] INTI. Website, última visita 15/11/2018. https://www.inti.gob.ar/

[48] Pietrantuono R., Russo S., "Introduction to Safety Critical Systems", Cotroneo D. Innovative Technologies for Dependable OTS-Based Critical Systems, Milano, 2013

[49] Jayasri K., Seetharamaiah P., "The Quantitative Safety Assessment and Evaluation for Safety-Critical Computer Systems", ACM SIGSOFT Software Engineering Notes, vol. 41, pp. 1-8, 2016.

[50] Tiwari S., Gupta A., "An Approach to Generate Safety Validation Test Cases from UML Activity Diagram", 20th Asia-Pacific Software Engineering Conference (APSEC), Bangkok, 2013, pp. 189-198, doi: 10.1109/APSEC.2013.35

[51] A. Lewiński y K. Trzaska–Rycaj, "The Safety Related Software for Railway Control with Respect to Automatic Level Crossing Signaling System", en Transport Systems Telematics, vol. 104, J. Mikulski, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 202-209.

[52] K. Hartig, J. Gerlach, J. Soto, y J. Busse, "Formal Specification and Automated Verification of Safety-Critical Requirements of a Railway Vehicle with Frama-C/Jessie", FORMS/FORMAT 2010, E. Schnieder y G. Tarnai, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 145-153.

[53] Eclipse Process Framework. Website, última visita 21/11/2018. https://www.eclipse.org/epf/

[54] LINSSE UNE-EN 50128. Website, última visita 21/11/2018. http://linsse.com.ar/epf/

[55] LINSSE. Website, última visita 21/11/2018. http://linsse.com.ar/

[56] GIT-SCM. Website, última visita 21/11/2018. https://git-scm.com/

[57] Frama-C. Website, última visita 21/11/2018. https://frama-c.com/

Anexo Artículos seleccionados RSL

[A1] A. Abdulkhaleq y S. Wagner, "A controlled experiment for the empirical evaluation of safety analysis techniques for safety-critical software", 2015, pp. 1-10.

[A2]S. Chandrasekaran, T. J. Madhumathy, M. Aparna, y R. S. Jain, "A safety enhancement model of software system for railways", 2009, pp. P2-P2.

[A3] J. Du, J. Wang, y X. Feng, "A Safety Requirement Elicitation Technique of Safety-Critical System Based on Scenario", Intelligent Computing Theory, vol. 8588, D.-S. Huang, V. Bevilacqua, y P. Premaratne, Eds. Cham: Springer International Publishing, 2014, pp. 127-136.

[A4] K. A. H. Nakamura y Y. Hirao, "A strategic approach to railway signalling software", International Conference on Main Line Railway Electrification 1989, York, 1989, pp. 327-331.

[A5] J. Luo, S. Liu, Y. Wang, y T. Zhou, "Applying SOFL to a Railway Interlocking System in Industry", Structured Object-Oriented Formal Language and Method, vol. 10189, S. Liu, Z. Duan, C. Tian, y F. Nagoya, Eds. Cham: Springer International Publishing, 2017, pp. 160-177.

[A6] T. L. Johnson, H. A. Sutherland, B. Ingleston, y B. H. Krogh, "DEPENDABLE SOFTWARE IN RAILWAY SIGNALLING", IFAC Proceedings Volumes, vol. 38, n.º 1, pp. 42-49, 2005.

[A7] T. Cichocki y J. Górski, "Failure Mode and Effect Analysis for Safety-Critical Systems with Software Components", Computer Safety, Reliability and Security, vol. 1943, F. Koornneef y M. van der Meulen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 382-394.

[A8] B. Dehbonei y F. Mejia, "Formal methods in the railways signalling industry", FME '94: Industrial Benefit of Formal Methods, vol. 873, M. Naftalin, T. Denvir, y M. Bertran, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 26-34.

[A9] K. Hartig, J. Gerlach, J. Soto, y J. Busse, "Formal Specification and Automated Verification of Safety-Critical Requirements of a Railway Vehicle with Frama-C/Jessie", FORMS/FORMAT 2010, E. Schnieder y G. Tarnai, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 145-153.

[A10] M. Wilikens, M. Masera, y D. Vallero, "Integration of Safety Requirements in the Initial Phases of the Project Lifecycle of Hardware/Software Systems", Safe Comp 97, P. Daniel, Ed. London: Springer London, 1997, pp. 83-97.

[A11] J. Qian, J. Liu, X. Chen, y J. Sun, "Modeling and Verification of Zone Controller: The SCADE Experience in China's Railway Systems", 2015, pp. 48-54.

[A12] Y. Chen, "Non-safety-related software in the context of railway RAMS standards", 2017, pp. 1-5.

[A13] A. El-Ansary, "Requirements Definition of Safe Software Using the Behavioral Patterns Analysis (PBA) Approach: The Railroad Crossing System", 2006, pp. 80-80.

[A14] P. Lüley, M. Franeková, y M. Hudák, "Safety and Functionality Assessment of Railway Applications in Terms of Software", Telematics in the Transport Environment, vol. 329, J. Mikulski, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 396-405.

[A15] U. Foschi, M. Giuliani, A. Morzenti, M. Pradella, y P. San Pietro, "Software procurement and methods for specification and validation in the railway transportation industry", 2002, vol. vol.6, p. 6.

[A16] R. C. Short, "Software requirements for railway signalling systems", IEE Colloquium on Software Requirements for High Integrity Systems, London, 1988, pp. 4/1-4/3.

[A17] B. S. Medikonda y P. S. Ramaiah, "Software Safety Analysis to Identify Critical Software Faults in Software-Controlled Safety-Critical Systems", ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol II, vol. 249, S. C. Satapathy, P. S. Avadhani, S. K. Udgata, y S. Lakshminarayana, Eds. Cham: Springer International Publishing, 2014, pp. 455-465.

[A18] P. J. Bryan, "Software Safety and Dependability for Railway Control Systems", 2014, pp. 16 (21 .)-16 (21 .).

[A19] S. Patra, "Software safety assurance process for railway platform software", 2007, vol. 2007, pp. 72-77.

[A20] A. J. Harrison y I. D. R. Shannon, "The Application of Formal Methods to Railway Signalling Systems Specification and the Esprit III Project CASCADE", Safe Comp 95, G. Rabe, Ed. London: Springer London, 1995, pp. 101-112.

[A21] A. Lewiński y K. Trzaska–Rycaj, "The Safety Related Software for Railway Control with Respect to Automatic Level Crossing Signaling System", en Transport Systems Telematics, vol. 104, J. Mikulski, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 202-209.

[A22] K. W. W. Johnston, P. S. P. Robinson, y L. van den Berg, "Tool support for checking railway interlocking designs", SCS '05 Proceedings of the 10th Australian

workshop on Safety critical systems and software, Sydney, Australia, 2006, vol. 55, pp. 101-107.

[A23] A. Fantechi, "Twenty-Five Years of Formal Methods and Railways: What Next?", Software Engineering and Formal Methods, vol. 8368, S. Counsell y M. Núñez, Eds. Cham: Springer International Publishing, 2014, pp. 167-183.

[A24] J. L. Boulanger, "CENELEC 50128 and IEC 62279 Standards", Control, Systems and Industrial Engineering Series, John Wiley & Sons, Inc., 2015, p. 13.

[A25] NASA Software Safety Guidebook. NASA Technical Standard. NASA-GB-8719.13. Marzo, 2004.

[A26] J. Vilela, J. Castro, L. E. G. Martins, T. Gorschek, "Integration between Requirements Engineering and Safety Analysis: A Systematic Literature Review", The Journal of Systems & Software, Vol. 125, Pp. 68-92, Marzo, 2017.

Anexo F_ESC_01

Especificación de Ensayos del Software en Conjunto - Proyecto XX

Evidencia de la aplicación de los Procedimientos de Gestión de Requisitos del Software V1.1

Autor del presente formulario:

Lic. Cristian Pinto Luft (UNNE)

Revisores del presente formulario:

Dr. Ing. Emanuel Irrazábal (UNNE)

Aprobó el presente formulario:

Este formulario fue realizado en el marco del Proyecto CIAA en abril de 2017 y es de libre distribución.

1 Registro de cambios y lista de distribución

1.1 Registro de cambio

Versión	Fecha	Descripción	Autor
1.0	01/04/2017	Creación del Documento	Cristian Pinto Luft
1.1	10/08/2017	Correcciones	Cristian Pinto Luft

1.2 Lista de distribución

Versión	Fecha	Entregada a	Firma
1.0	01/04/2017	Intranet	Cristian Pinto Luft

2 Tabla de contenido

3 Especificación de Ensayos del Software en Conjunto

3.1 Ensayos del software

RQ-ID		Tipo de ensayo	
Señales de entrada			
Señales de salida			
Criterios de éxito			

Anexo F_VRS_01

Informe de Validación y verificación de los Requisitos del Software - Proyecto XX

Evidencia de la aplicación de los Procedimientos de Gestión de Requisitos del Software V0.4

Autor del presente formulario:

Lic. Cristian Pinto Luft (UNNE)

Revisores del presente formulario:

Dr. Ing. Emanuel Irrazábal (UNNE)

Aprobó el presente formulario:

Este formulario fue realizado en el marco del Proyecto CIAA en abril de 2017 y es de libre distribución.

1 Registro de cambios y lista de distribución

1.1 Registro de cambio

Versión	Fecha	Descripción	Autor
1.0	01/04/2017	Creación del Documento	Cristian Pinto Luft
1.2	09/08/2017	Correcciones	Cristian Pinto Luft

1.2 Lista de distribución

Versión	Fecha	Entregada a	Firma
1.0	01/04/2017	Intranet	Cristian Pinto Luft

2 Tabla de contenido

3 Informe de Verificación de los Requisitos del Software

3.1 Validación de requerimientos con stakeholders

RQ-ID	Modelo de validación	Stakeholders	Fecha	Resultado

3.2 Verificación de contradicciones y seguridad de los requisitos

RQ-ID	Contradicción con RQ-ID	Cumplimiento con nivel de seguridad

3.3 Verificación de los requerimientos del software

RQ-ID		Fecha		Responsable	
Adecuación		Legibilidad		Trazabilidad	
Ensayos		Coherencia interna		Restricciones HW y SW	
Observaciones					
Resultados					

Anexo Primeros Requerimientos

1. Descripción del banco de ensayos que se instalará en el INTI

El banco de ensayos que se instalará en el INTI se ilustra en la figura 1. Por la vía, que tendrá 18 metros, circulará una dresina, también conocida como «zorra de vía» o «zorra de rieles». Cuando la dresina ingrese en determinada zona se activará el motor que baja el brazo de la barrera, las luces indicadoras y la sirena. Al retirarse la dresina el motor levantará el brazo y se desactivarán las luces y la sirena.

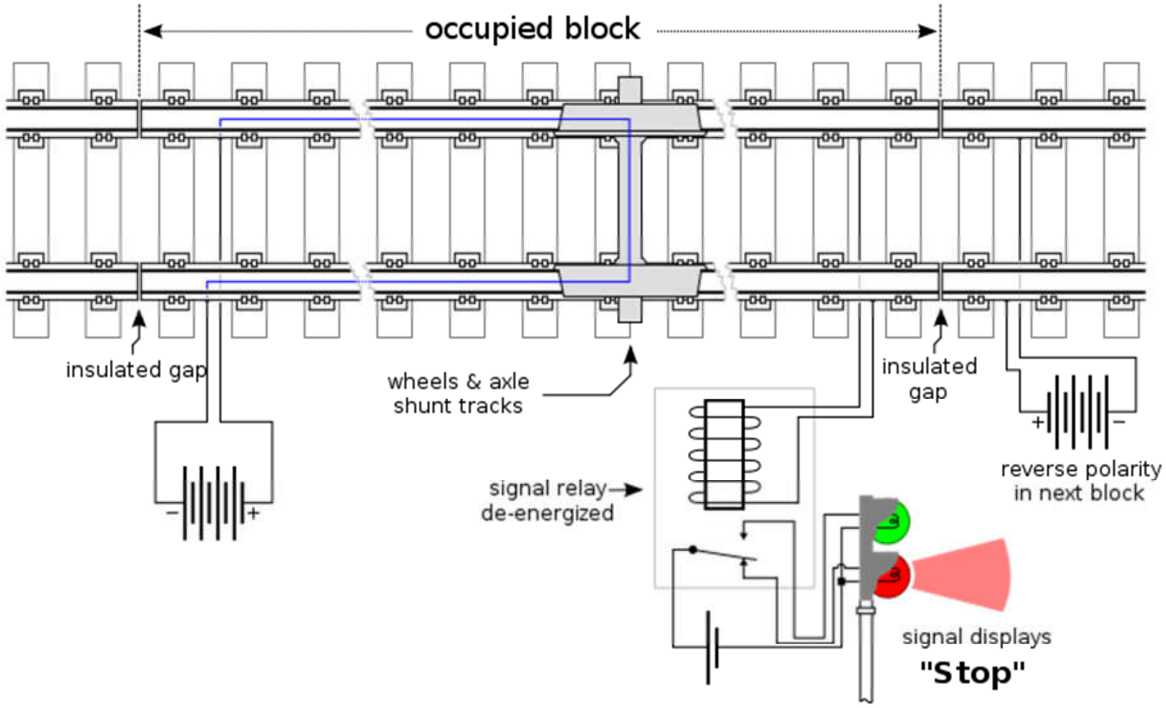


Figura 1. Esquema general de la instalación que se montará en el INTI, a lo que además se incorporará el motor que sube y baja el brazo de barrera y la sirena.

2. Requisitos de entradas, salidas e interfaces de comunicación

El Monitor de Barreras tendrá las siguientes entradas y salidas:

Entradas digitales:

- Señal de ocupación de vía (presencia de tren o ausencia del tren).
- Señal de barrera arriba (activada o desactivada).
- Señal de barrera abajo (activada o desactivada).
- Señal de sensor de brazo roto (activada o desactivada).
- Señal de semáforo del PAN (encendido o apagado).
- Señal de puerta de gabinete (abierta o cerrada).
- Señal de estado de los relé de accionamiento (activado o desactivado).
- Señal de energía eléctrica (presencia o ausencia de 220V en la barrera).

Salidas digitales:

- Indicador luminoso de estado normal (no hay falla detectada)
- Indicador luminoso de estado de riesgo (se ha detectado una situación anormal)
- Indicador luminoso de falla en batería
- Indicador luminoso de falla de relés
- Indicador luminoso de falla electrónica del monitor de barrera (autodiagnóstico)

Entradas analógicas

- Tensión de batería (es la batería que alimenta al motor que mueve el brazo)
- Sensor de ángulo de la barrera (es un a tensión vinculada a un potenciómetro)
- Señal de ocupación de vía (valor analogico, que permite analizar funcionamiento)
- Lámparas de semáforos para vehículos y peatones (sensado de corriente)
- Lámparas de semáforos para conductores del tren (sensado de corriente)
- Temperatura de habitáculo del monitor de barreras (es una tensión)

Interfaces de comunicación

- Interfaz 2G/3G

La premisa del desarrollo es la no interferencia con el sistema de accionamiento, esto significa que tanto las entradas como las salidas deberán ser independientes del sistema bajo monitoreo y técnicamente aisladas por dispositivos optoelectrónicos o similares.

3. Requisitos funcionales

El Monitor de Barreras podrá estar en uno de estados que se muestran en la Figura 2.

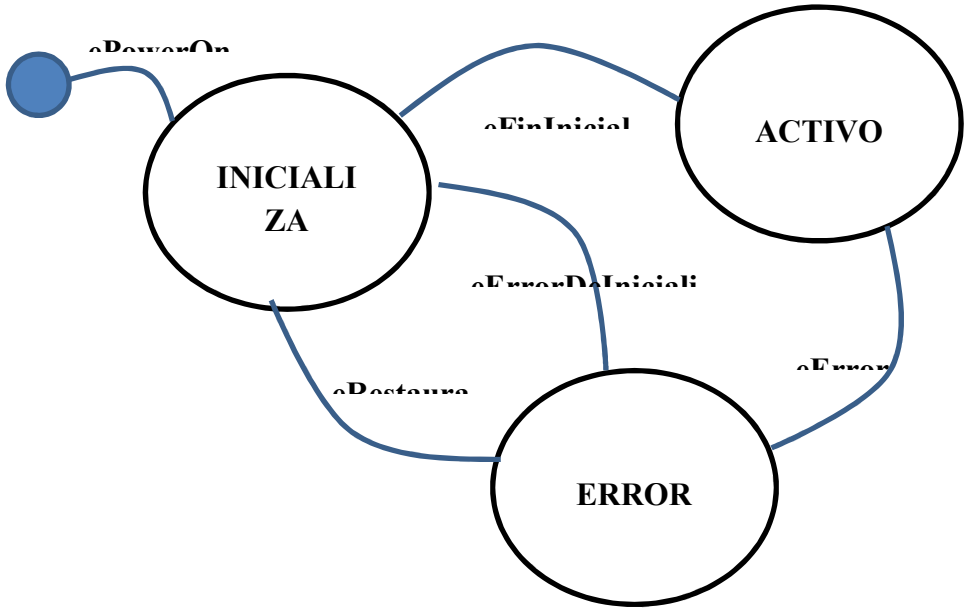


Figura 2 Diagrama en estados del Monitor de Barreras

Al energizar el sistema se inicia en el estado INICIALIZA. En este estado se realiza una comprobación general del sistema.

Finalizado con éxito el proceso de inicialización el sistema evoluciona al estado ACTIVO. Este estado se monitorea el estado de los sensores y se activa las señales luminosas de indicación de estado. De no existir fallas el sistema transmite cada 10 minutos el estado del sistema de barreras y ante la aparición de un evento también transmite.

En estado ACTIVO el sistema debe comprobar que al activarse la señal de ocupación de vía la barrera se baja en un tiempo acorde al esperado (lo que se determina a partir de las señales de barrera arriba y barrera abajo) y que se activan la sirena y las luces.

Al desactivarse la señal de ocupación de vía el sistema debe comprobar que la barrera sube y que se desactiva la sirena y las luces.

Al estado de ERROR se ingresa ante un evento de error relacionado con temperatura, tensión de batería, etc.

Anexo R_ESC_01 Monitor de Barreras - Segunda implementación

Especificación de Ensayos del Software en Conjunto - Monitor de Barreras

Evidencia de la aplicación de los Procedimientos de Gestión de Requisitos del Software V0.4

Autor del presente formulario:

Lic. Cristian Pinto Luft (UNNE)

Revisores del presente formulario:

Dr. Ing. Emanuel Irrazábal (UNNE)

Aprobó el presente formulario:

Este formulario fue realizado en el marco del Proyecto CIAA en abril de 2017 y es de libre distribución.

1 Registro de cambios y lista de distribución

1.1 Registro de cambio

Versión	Fecha	Descripción	Autor
1.0	10/08/2017	Creación del Documento	Cristian Pinto Luft
1.1	17/09/2018	Completado de requerimientos de la primer implementación	Cristian Pinto Luft

1.2 Lista de distribución

Versión	Fecha	Entregada a	Firma
1.0	10/08/2017	Intranet	Cristian Pinto Luft

2 Tabla de contenido

3 Especificación de Ensayos del Software en Conjunto

3.1 Ensayos del software

3.1.1 RQ-000001

RQ-ID	RQ-000001	Tipo de ensayo	Funcional
Señales de entrada		Entrada de corriente en el monitor de barreras	
Señales de salida		Informe de comprobación del estado de los componentes	
Criterios de éxito		El monitor de barreras envía un informe de comprobación de componentes a la central de control dentro del lapso de tiempo establecido.	

3.1.2 RQ-000002

RQ-ID	RQ-000002	Tipo de ensayo	Funcional
Señales de entrada		• Rango de tiempo (10 minutos) • Evento de error • Señal de control	
Señales de salida		• Señal luminosa • Informe de comprobación del estado de los componentes	
Criterios de éxito		El monitor de barreras enciende las señales luminosas en caso de ocurrencia de un error, cada 10 minutos o bajo demanda de la central. Además envía un informe de comprobación de componentes a la central de control dentro del lapso de tiempo establecido.	

3.1.3 RQ-000003

RQ-ID	RQ-000003	Tipo de ensayo	Funcional
Señales de entrada		Señal de ocupación de vía	
Señales de salida		• Señal luminosa • Señal auditiva • Señal de control	
Criterios de éxito		Al recibirse la señal de ocupación de vía, el monitor de barreras registra la duración del evento de bajada de la barrera en su memoria local. En caso de que la misma supere los umbrales establecidos, envía una señal de control indicando la situación al centro de control. Además, activa las sirenas y luces correspondientes.	

3.1.4 RQ-000004

RQ-ID	RQ-000004	Tipo de ensayo	Funcional
Señales de entrada		Señal de desocupación de vía	
Señales de salida		Señal de control	
Criterios de éxito		Al recibirse la señal de desocupación de vía, el monitor de barreras registra la duración del evento de subida de la barrera en su memoria local. En caso de que la misma supere los umbrales establecidos, envía una señal de control indicando la situación al centro de control.	

3.1.5 RQ-000005

RQ-ID	RQ-000005	Tipo de ensayo	Funcional
Señales de entrada		Señal de error de un componente registrado	
Señales de salida		Señal de control	
Criterios de éxito		El monitor de barreras envía una señal de control (alerta) con la información correspondiente al centro de control al producirse un error en un componente registrado.	

3.1.6 RQ-000006

RQ-ID	RQ-000006	Tipo de ensayo	Funcional
Señales de entrada		Señal de evento anómalo	
Señales de salida		Cambio de estado del monitor a ACTIVO con estado interno ALARMA	
Criterios de éxito		El monitor de barreras cambia a estado ACTIVO con estado interno ALARMA al producirse un evento anómalo.	

3.1.7 RQ-000007

RQ-ID	RQ-000007	Tipo de ensayo	Funcional
Señales de entrada		Intervalo de tiempo configurable	
Señales de salida		Información de control almacenada en memoria local del monitor	
Criterios de éxito		El centro de control obtiene información de los últimos N minutos (configurables) del estado de los componentes del monitor de barreras al solicitarla al mismo mediante una señal de control.	

3.1.8 RQ-000008

RQ-ID	RQ-000006	Tipo de ensayo	Funcional
Señales de entrada		Señal de mantenimiento enviada al monitor de barreras.	
Señales de salida		El monitor de barreras pasa a estado EN CONFIGURACIÓN y se desactivan las alarmas.	
Criterios de éxito		Al enviarse la señal de mantenimiento desde el centro de control al monitor de barreras, el mismo pasa a estado EN CONFIGURACIÓN y se desactivan las alarmas.	

Anexo R_VRS_01 Monitor de Barreras - Segunda implementación

Informe de Validación y verificación de los Requisitos del Software - Monitor de Barreras

Evidencia de la aplicación de los Procedimientos de Gestión de Requisitos del Software V0.4

Autor del presente formulario:

Lic. Cristian Pinto Luft (UNNE)

Revisores del presente formulario:

Dr. Ing. Emanuel Irrazábal (UNNE)

Aprobó el presente formulario:

Este formulario fue realizado en el marco del Proyecto CIAA en abril de 2017 y es de libre distribución.

1 Registro de cambios y lista de distribución

1.1 Registro de cambio

Versión	Fecha	Descripción	Autor
1.0	10/08/2017	Creación del Documento	Cristian Pinto Luft
1.1	17/10/2018	Completado de requerimientos de la primer implementación	Cristian Pinto Luft

1.2 Lista de distribución

Versión	Fecha	Entregada a	Firma
1.0	10/08/2017	Intranet	Cristian Pinto Luft

2 Tabla de contenido

3 Informe de Verificación de los Requisitos del Software

3.1 Validación de requerimientos con stakeholders

RQ-ID	Modelo de validación	Stakeholders	Fecha	Resultado
RQ-000001	Diagramas, documentación	GH-01	01/09/2017	Correcto
RQ-000002	Diagramas, documentación	GH-01	17/10/2018	Correcto
RQ-000003	Diagramas, documentación	GH-01	17/10/2018	Correcto
RQ-000004	Diagramas, documentación	GH-01	17/10/2018	Correcto
RQ-000005	Diagramas, documentación	GH-01	17/10/2018	Correcto
RQ-000006	Diagramas, documentación	GH-01	17/10/2018	Correcto
RQ-000007	Diagramas, documentación	GH-01	25/10/2018	Correcto
RQ-000008	Diagramas, documentación	GH-01	25/10/2018	Correcto

3.2 Verificación de contradicciones y seguridad de los requisitos

RQ-ID	Contradicción con RQ-ID	Cumplimiento con nivel de seguridad
RQ-000001	-	Cumple
RQ-000002	-	Cumple
RQ-000003	-	Cumple
RQ-000004	-	Cumple

RQ-000005	-	Cumple
RQ-000006	-	Cumple
RQ-000007	-	Cumple
RQ-000008	-	Cumple

3.3 Verificación de los requerimientos del software

3.3.1 RQ-000001

RQ-ID	RQ-000001	Fecha	10/08/2017	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	10	Trazabilidad	10
Ensayos	9	Coherencia interna	9	Restricciones HW y SW	9
Observaciones	Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	El requerimiento cumple con todos los puntos indicados de manera correcta				

3.3.2 RQ-000002

RQ-ID	RQ-000002	Fecha	17/10/2018	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	8	Trazabilidad	10
Ensayos	9	Coherencia interna	9	Restricciones HW y SW	9
Observaciones	Se deben indicar detalles de las señales luminosas: colores, frecuencia de intermitencia y demás. Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	Corregir los puntos indicados en las observaciones.				

3.3.3 RQ-000003

RQ-ID	RQ-000003	Fecha	17/10/2018	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	7	Trazabilidad	10
Ensayos	9	Coherencia interna	8	Restricciones HW y SW	9
Observaciones	Se deben indicar detalles de las señales luminosas: colores, frecuencia de intermitencia y demás. Se deben indicar detalles de las señales sonoras: volumen, tono, frecuencia, duración y demás. Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	Corregir los puntos indicados en las observaciones.				

3.3.4 RQ-000004

RQ-ID	RQ-000004	Fecha	17/10/2018	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	9	Trazabilidad	10
Ensayos	9	Coherencia interna	9	Restricciones HW y SW	9
Observaciones	Se debe indicar la forma de configurar el tiempo definido como “normal” para la subida (y bajada) de la barrera. Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	Corregir los puntos indicados en las observaciones.				

3.3.5 RQ-000005

RQ-ID	RQ-000005	Fecha	17/10/2018	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	8	Trazabilidad	9
Ensayos	8	Coherencia interna	9	Restricciones HW y SW	7
Observaciones	Se debe indicar la forma en que los distintos componentes indican al monitor de barreras que se ha producido un error (sus interfaces). Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	Corregir los puntos indicados en las observaciones.				

3.3.6 RQ-000006

RQ-ID	RQ-000006	Fecha	10/08/2017	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	9	Trazabilidad	10
Ensayos	9	Coherencia interna	9	Restricciones HW y SW	9
Observaciones	Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	Corregir los puntos indicados en las observaciones.				

3.3.7 RQ-000007

RQ-ID	RQ-000007	Fecha	10/08/2017	Responsable	Cristian Pinto Luft
Adecuación	10	Legibilidad	9	Trazabilidad	10
Ensayos	8	Coherencia interna	9	Restricciones HW y SW	9
Observaciones	Se debe indicar la forma de configurar el tiempo en que se mantendrán los datos en memoria local. Se debe indicar la forma en que se consultarán, insertarán y borrarán los datos del almacenamiento local. Se debe indicar el formato de los datos que se guardarán. Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	Corregir los puntos indicados en las observaciones.				

3.3.8 RQ-000008

RQ-ID	RQ-000008	Fecha	10/08/2017	Responsable	Cristian Pinto Luft
Adecuación	7	Legibilidad	7	Trazabilidad	10
Ensayos	9	Coherencia interna	5	Restricciones HW y SW	9
Observaciones	Aclarar el objetivo en la especificación, ya que indica que hay que generar una alarma, cuando en la necesidad se indica que se deben desactivar las alarmas. Esta tarea en realidad debería realizarlo el Verificador de Requerimientos, no el Gestor de Requerimientos				
Resultados	Corregir los puntos indicados en las observaciones.				

Anexo Tercer versión del PG

PG_RS_04 Requisitos del Software

Versión 0.5

Autores:

Lic. Cristian Pinto Luft (UNNE)

Revisores:

Dr. Ing. Emanuel Irrazábal (UNNE)

Aprobó:

1 Registro de cambios y lista de distribución

1.1 Registro de cambio

Versión	Fecha	Descripción	Autor
0.1	06/03/2017	Creación del Documento	Cristian Pinto Luft
0.2	27/03/2017	Agregado Capítulo 8 - Entregas	Cristian Pinto Luft
0.3	10/04/2017	Agregado Anexo 1	Cristian Pinto Luft
0.4	29/06/2017	Agregados aspectos de seguridad	Cristian Pinto Luft
0.5	21/08/2018	Agregadas especificaciones formales y semi formales	Cristian Pinto Luft

1.2 Lista de distribución

Versión	Fecha	Entregada a	Firma
0.1	06/03/2017	Intranet	Cristian Pinto Luft

2 Tabla de contenido

3 Objetivo

3.1 Este documento tiene como objeto establecer un procedimiento general para la gestión de requisitos del software, que sirva para gestionar de manera sistemática todos los procesos relativos al correcto tratamiento de los requerimientos de software dentro del proyecto, cumpliendo estos con las pautas de seguridad establecidas en el mismo.

4 Alcance

4.1 Este procedimiento general se usa para gestionar los requisitos del software de aplicaciones ferroviarias de acuerdo con la normativa UNE-EN 50126.

5 Referencias

5.1 Los procesos definidos por el presente documento se basan en la serie ISO 9000 de Normas Internacionales y en la serie UNE-EN 50128.

6 Responsabilidades

6.1 El responsable de actualizar este documento es la figura del Gestor de Requerimientos, rol definido en la tabla 6 del documento PG_ACS_09 Aseguramiento de la Calidad del Software.

7 Descripción del procedimiento

- Los procesos de obtención, especificación y verificación relacionados a la Gestión de los Requerimientos del Software y sus secuencias se ilustran en la Figura 1.
- Durante el procedimiento, cuando se utilicen escalas de valoración, las mismas irán del 1 al 10, siendo 1 el valor menos y 10 el más significativo.

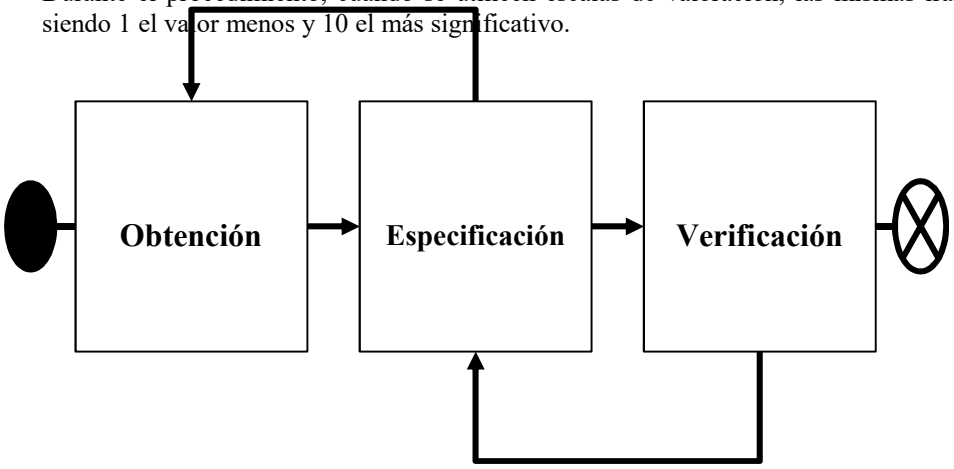


Figura 1

Gestión de Requerimientos del Software.

7.1 Proceso de Obtención

7.1.1 El proceso de obtención de requerimientos indica las pautas a seguir para obtener, documentar, identificar, clarificar y justificar los mismos.

7.1.2 La Figura 2 describe este proceso. La Tabla 1 muestra las actividades secuenciales que definen el proceso.

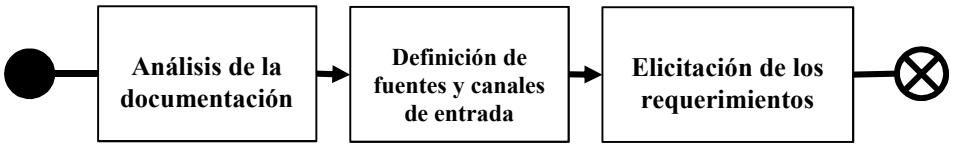


Figura 2

Diagrama del proceso de Obtención de requerimientos.

Tabla 1

Información específica del proceso

Responsable: Gestor de Requerimientos.

Objetivo: Realizar la obtención de requerimientos de software del sistema asegurando la calidad de los mismos.

Resultados esperados: Requerimientos software del sistema.

Alcance: El Sistema de Gestión de Requerimientos del Software.

Errores a evitar: Fuentes de entradas de requerimientos no identificadas, canales de comunicación no definidos y requerimientos mal obtenidos.

Marco normativo: Norma UNE-EN 50128.

Orden	Actividad	Responsable de ejecución	Registro
1	Realizar un análisis sistemático de la documentación de entrada.	Gestor de Requerimientos	Filas 2, 3, 4, 5, 6 y 7 de la Tabla 2
2	Identificar los stakeholders y canales de entrada de los requerimientos. Los stakeholders serán identificados mediante un valor con el formato SH-XXXXX. Los grupos de stakeholders serán identificados mediante un valor con el formato GSH-XXX.	Gestor de Requerimientos	Filas 2 de la Tabla 3 y 5
3	Definir el formato y las técnicas de elicitación a utilizar, al igual que la planificación.	Gestor de Requerimientos	Fila 2 de la Tabla 7
4	Realizar la elicitación de los requerimientos. Los mismos serán identificados mediante un valor con el formato RQ-XXXXX. Los requerimientos de seguridad serán identificados mediante un valor con el formato RQS-XXXXX.	Gestor de Requerimientos	Columna 2 de la Tabla 9

7.1.3 Una de las fuentes de entrada de los requerimientos del software serán los requerimientos de seguridad del software identificados en el 7.2 Proceso de Identificación de secuencias de acontecimiento de peligros del PG_ ARI_10 Análisis de Riesgos.

Tabla 2

Análisis de documentación de entrada.

Documento	Fecha de análisis	Completitud	Correctitud
Especificación de Requisitos del Sistema			
Especificación de Requisitos de Seguridad del Sistema			
Descripción de la Arquitectura del sistema			
Especificaciones de la Interfaz Externa			
Plan de Aseguramiento de Calidad del Software			
Plan de Validación del Software			
Observaciones			

Tabla 3

Identificación de stakeholders y canales de comunicación

Identificador	Nombre y apellido	Puesto o función	Canales de comunicación

Un ejemplo del uso de la Tabla 3 puede ser observado en la Tabla 4.

Tabla 4

Ejemplo de identificación de stakeholders y canales de comunicación

Identificador	Nombre y apellido	Puesto o función	Canales de comunicación
SH-00001	Mariano Moreno	Maquinista del tren	Email: marianomoreno@ejemplo.com.ar Dirección: Ciudad, Calle, Altura Teléfono: 555- 4444
SH-00002	Roberto Perez	Operador de sistemas	Email: robertoperez@ejemplo.com.ar Teléfono: 555- 4443

7.1.4 Los stakeholders identificados pueden ser incluidos dentro de grupos para su mejor tratamiento, como se puede ver en la Tabla 5.

Tabla 5

Grupos de stakeholders

Identificador de grupo	Nombre de grupo	Descripción de grupo	Miembros del grupo

7.1.5 Un ejemplo del uso de la Tabla 5 puede ser observado en la Tabla 6.

Tabla 6

Ejemplo de grupos de stakeholders

Identificador de grupo	Nombre de grupo	Descripción de grupo	Miembros del grupo
GSH-001	Operarios de trenes	Personas vinculadas a la operación física y lógica de los trenes	SH-00001 SH-00002

Tabla 7

Técnicas de elicitación

Técnica	Duración aproximada	Objetivos

7.1.6 Ejemplos de técnicas de elicitación que se pueden utilizar son entrevistas y encuestas, como se puede ver por ejemplo en la Tabla 8.

Tabla 8

Ejemplos de técnicas de elicitación

Técnica	Duración aproximada	Objetivos
Entrevista	3 horas aprox.	Obtener información sensible y puntual sobre un tema en específico, difícil de hallar mediante comunicación no presencial.
Cuestionario	1 hora aprox.	Obtener información general sobre temas específicos.

Tabla 9

Elicitación de requerimientos

RQ-ID		Fecha		Fuente	
Necesidad					
Motivo					
Objetivo					
Verificación					

7.1.7 Un ejemplo del uso de la tabla anterior puede ser visto en la Tabla 10.

Tabla 10

Ejemplo de elicitación de requerimientos

RQ-ID	RQ-00001	Fecha	25/03/2017	Fuente	GSH-00001
Necesidad	El monitor de barrera debe informar su estado cada 5 segundos				
Motivo	Necesidad de contar con información actualizada del dispositivo				

Objetivo	Reducir las probabilidades de un accidente
Verificación	El monitor de barreras informa su estado cada 5 segundos

7.2 Proceso de Especificación

7.2.1 El proceso de especificación de requerimientos indica las pautas a seguir para analizarlos, definirlos y especificar los cambios que estos provocarán en el sistema en caso de implementarse.

7.2.2 El nivel de integridad de seguridad del software debe registrarse en la Especificación de Requisitos del Software.

7.2.3 Los requerimientos de tipo no funcional pueden tener incluídos, entre otros, los subtipos: robustez, mantenibilidad, seguridad, eficiencia, usabilidad y portabilidad.

7.2.4 Los requerimientos funcionales hacen referencia a cualidades o funcionalidades del software.

7.2.5 Se deben explicitar los requisitos utilizados para los ensayos periódicos de funciones, indicando que es un requisito de tipo no funcional y de subtipo ensayo.

7.2.6 Cuando sea necesario, se deben expresar los requerimientos mediante métodos formales o semi formales.

7.2.7 Los distintos estados por los que puede pasar el software serán modelados mediante un árbol SSTA. El evento iniciador del árbol será el estado descrito por la verificación del requerimiento ingresado.

7.2.8 Se deben explicitar los requisitos relacionados a la autocomprobación del hardware y del software, especificando la detección y el envío de informes de fallos y errores por parte de los mismos. Los mismos serán definidos como requisitos de tipo funcional y subtipo autocomprobación.

7.2.9 Se deben realizar análisis del impacto de la seguridad de cada uno de los requerimientos software especificados, para detectar los peligros que podrían llegar a provocar. Esta tarea se realizará a nivel funcional mediante el uso complementario de las técnicas de análisis de seguridad SFMEA y SFTA.

7.2.10 La Figura 3 describe este proceso. La Tabla 11 muestra las actividades secuenciales que definen el proceso.

Figura 3

Diagrama del proceso de Especificación de requerimientos.

Tabla 11

Información específica del proceso

Responsable: Gestor de Requerimientos.

Objetivo: Analizar, definir y especificar los requerimientos obtenidos en el proceso de Obtención, asegurando la calidad de los mismos.

Resultados esperados: Especificación de requisitos del software.

Alcance: El Sistema de Gestión de Requerimientos.

Errores a evitar: Requerimientos contradictorios, mal interpretados o mal definidos.

Marco normativo: Norma UNE-EN 50128.

Orden	Actividad	Responsable de ejecución	Registro
1	Realizar un análisis operacional de los requerimientos.	Gestor de Requerimientos	Fila 2 y 3 de la Tabla 12
2	Realizar un análisis sistémico de los requerimientos.	Gestor de Requerimientos	Fila 4 y 5 de la Tabla 12
3	Definir atributos característicos de los requerimientos	Gestor de Requerimientos	Fila 2 de la Tabla 14
4	Definir los distintos modos de comportamiento del software.	Gestor de Requerimientos	Árbol SSTA generado con la Figura 4.
5	Definir los distintos modos de fallo del software.	Gestor de Requerimientos	Árbol SFTA generado con el complemento del árbol SSTA, como se puede observar en el ejemplo de la Figura 6.
6	Realizar el SFMEA funcional de los requerimientos especificados.	Gestor de Requerimientos	Tabla 16
7	Realizar las especificaciones semi formales y formales.	Gestor de Requerimientos	
8	Construir un glosario de términos	Gestor de Requerimientos	Tabla 18
9	En caso de generarse nuevos requerimientos a partir de la especificación de los actuales,	Gestor de Requerimientos	7.1 Proceso de Obtención

	registrarlos.		

Tabla 12

Análisis de requerimientos

RQ-ID				
Análisis operacional	Fecha		Responsable	
Resultados				
Análisis sistémico	Fecha		Responsable	
Resultados				

7.2.11 Un ejemplo del uso de la tabla anterior puede ser visto en la Tabla 13.

Tabla 13

Ejemplo de análisis de requerimientos

RQ-ID	RQ-00001			
Análisis operacional	Fecha	24/03/2017	Responsable	Juan Ramirez
Resultados	La implementación del requerimiento permitiría tener información actualizada del estado del monitor de barreras, a un intervalo de tiempo razonable.			
Análisis sistémico	Fecha	25/03/2017	Responsable	Pedro Pujol
Resultados	El requerimiento puede ser implementado mediante la creación de una tabla en la base de datos que registre los estados del monitor de barreras, manteniendo un cierto número de registros del mismo.			

Tabla 14

Atributos de los requerimientos

RQ-ID		Versión		Responsable	
Tipo		Subtipo		Estado	
Prioridad		Esfuerzo		Impacto en la seguridad	

Restricciones de HW	
Restricciones de SW	

7.2.12 Un ejemplo del uso de la tabla anterior puede ser visto en la Tabla 15.

Tabla 15

Ejemplo de atributos de los requerimientos

RQ-ID	RQ-00001	Versión	1.0	Responsable	Pedro Pujol
Tipo	Funcional	Subtipo	Autocompro bación	Estado	Aprobado
Prioridad	8	Esfuerzo	5	Impacto en la seguridad	9
Restricciones de HW		Congestión de canales de comunicación.			
Restricciones de SW		Dificultad para adaptar las tablas involucradas en el proceso.			

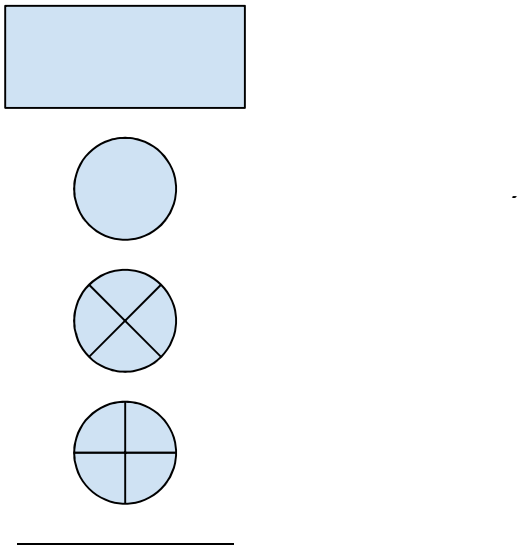


Figura 4

Símbolos básicos del árbol SSTA

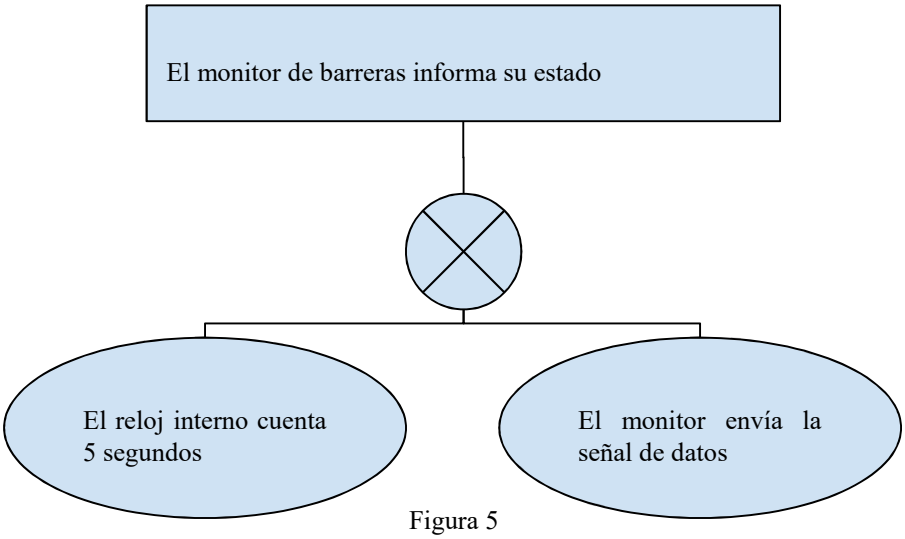


Figura 5

Ejemplo de árbol SSTA

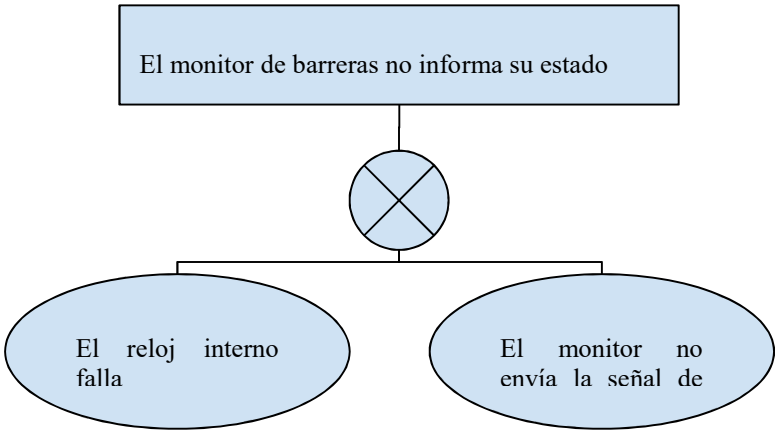


Figura 6

Ejemplo de árbol SFTA

Tabla 16

SFMEA funcional de los requerimientos

RQ-ID/SFMEA-ID						
Modo de fallo						
Efecto	Local					
	Subsistema					
	Sistema					
Causas						
Detección						
Mitigación						
Prevención						
Severidad			Frecuencia		Riesgo	

7.2.12 Cada caso puede generar un nuevo requerimiento de seguridad del software o del sistema, y se pueden descubrir nuevos peligros.

7.2.13 Por cada evento básico detectado en el árbol SFTA anteriormente generado se realiza un análisis SFMEA.

7.2.14 Los grados de severidad, en orden ascendente pueden ser: menor, mayor, crítico o catastrófico.

7.2.15 La frecuencia de ocurrencia, en orden ascendente puede ser: muy rara, remota, ocasional, probable o frecuente.

7.2.16 El grado de riesgo, en orden ascendente puede ser: aceptable, semi aceptable o no aceptable.

Para evaluar este último se debe utilizar la Tabla de Riesgos que se propone a continuación.

Tabla de Riesgos

Frecuencia/ Severidad	Muy rara	Remota	Ocasional	Probable	Frecuente
Catastrófico	S-Aceptable	N-Aceptable	N-Aceptable	N-Aceptable	N-Aceptable
Crítico	Aceptable	S-Aceptable	S-Aceptable	N-Aceptable	N-Aceptable
Mayor	Aceptable	Aceptable	S-Aceptable	S-Aceptable	N-Aceptable
Menor	Aceptable	Aceptable	Aceptable	S-Aceptable	S-Aceptable

7.2.17 Un ejemplo del uso de la Tabla 16 puede ser visto en la Tabla 17.

Tabla 17

Ejemplo de SFMEA funcional de los requerimientos

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000001
Modo de fallo		El reloj interno falla
Efecto	Local	Incertidumbre para el control del software Sobrecarga de buffers de memoria
	Subsistema	Desconocimiento del estado de la barrera Congestión de canales de comunicación
	Sistema	Peligro de mala señalización
Causas		Des sincronización del reloj Agotamiento de la batería del reloj Falla física del reloj
Detección		Pasan más de 5 segundos sin recibirse señales del monitor en el centro de control Se reciben 2 señales simultáneas del monitor con menos de 5 segundos de diferencia
Mitigación		El sistema de control solicita la re sincronización al reloj del monitor de barreras Se envía personal técnico a reemplazar las baterías del reloj Se envía personal técnico a reparar la avería del reloj

Prevención	Se posee redundancia de relojes para que se active uno si falla el otro Se posee redundancia de fuentes de alimentación para el reloj principal				
Severidad	Mayor	Frecuencia	Remota	Riesgo	Aceptable

7.2.18 En cuanto a las especificaciones formales y semi formales, se deberán llevar a cabo mediante lenguaje ACSL y diagramas de estado respectivamente.

7.2.19 Para realizar la especificación semi formal, por cada requerimiento se deberá graficar su diagrama de estados, indicando como mínimo los distintos estados por los que puede pasar el sistema (dentro del contexto del requerimiento), y los eventos que los desencadenan, como se puede ver en la Figura 7. Alternativamente se pueden realizar diagramas de clases que ayuden a comprender el contexto del requerimiento y a la creación de su diagrama de estados correspondiente.

Figura 7

Elementos del diagrama de estados

7.2.20 Para realizar la especificación formal, por cada especificación semi formal se deberán crear los contratos de funciones en los comentarios del código utilizando el lenguaje de especificación formal ACSL, indicando como mínimo las pre y post condiciones de los mismos. Solamente se deberán crear los contratos de funciones, no las funciones en sí, ya que las mismas se crearán en una etapa posterior del desarrollo. Esto permitirá a su vez la realización de la verificación del código a generar.

Un ejemplo del código ACSL a generar se puede ver en el siguiente contrato de función, en donde se especifica que las pre condiciones son que la variable A y la variable B sean dos enteros, y además se define la post condición, que indica que, al finalizar, la variable C deberá contener el número entero resultante de la suma de las dos anteriores:

```
/*@      requires variableA == (integer) && variableB == (integer);
         ensures variableC == (integer) variableA + variableB;
*/
```

Tabla 18

Glosario de términos

Término	Definición

7.2.21 Un ejemplo del uso de la tabla anterior puede ser visto en la Tabla 19.

Tabla 19

Ejemplo de glosario de términos

Término	Definición
Stakeholder	Personas u organizaciones que afectan o son afectadas por el proyecto.

7.3 Proceso de Verificación y validación

7.3.1 El proceso de validación de requerimientos del software indica las pautas a seguir para validar las especificaciones de requerimientos generadas en el proceso anterior con los stakeholders, modificarlos y redefinirlos en un proceso iterativo de ser necesario, definir los ensayos a realizarse sobre los mismos

7.3.2 Los requerimientos pueden ser validados con stakeholders individuales o grupos identificados.

7.3.3 El proceso de verificación de requerimientos del software indica las pautas para llevar a cabo los distintos tipos de controles y pruebas técnicas a llevarse a cabo sobre los requerimientos especificados, para mantener el nivel de seguridad requerido del software.

7.3.4 La Figura 8 describe este proceso. La Tabla 20 muestra las actividades secuenciales que definen el proceso.

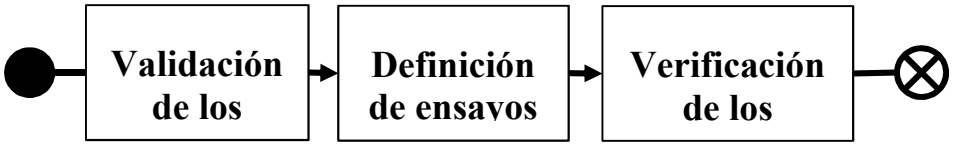


Figura 8

Diagrama del proceso de verificación y validación.

Tabla 20

Información específica del proceso

Responsable: Verificador de Requerimientos y Gestor de Requerimientos. Objetivo: Validar los requerimientos especificados con los usuarios, y definir sus modos de comportamiento y los ensayos a realizarse sobre los mismos con tal de verificarlos. Resultados esperados: Requerimientos del sistema validados. Especificación de ensayos del software en conjunto. Informe de verificación de los requisitos del software. Alcance: El Sistema de Gestión de Requerimientos. Errores a evitar: Requerimientos mal especificados o no verificados. Marco normativo: Norma UNE-EN 50128.			
Orden	Actividad	Responsable de ejecución	Registro
1	Validar con los stakeholders los requerimientos especificados mediante el uso de modelos.	Gestor de Requerimientos	Fila 2 de la Tabla 21
2	Definir los distintos ensayos del software a realizar.	Verificador de Requerimientos, Encargado de Ensayos del Software	Tabla 23
3	Verificar la no contradicción de los requerimientos.	Verificador de Requerimientos	Fila 2 de la Tabla 25
4	Verificar el cumplimiento de los requerimientos con los niveles de seguridad especificados para el software.	Verificador de Requerimientos	Fila 2 de la Tabla 25
5	Verificar los requisitos del software	Verificador de Requerimientos	Tabla 27
6	En caso de generarse modificaciones en los requerimientos durante el proceso, modificar las especificaciones.	Gestor de Requerimientos	7.2 Proceso de Especificación

Tabla 21

Verificación de requerimientos con stakeholders

RQ-ID	Modelo de validación	Stakeholders	Fecha	Resultado

7.3.4 Un ejemplo de la instanciación de la tabla anterior puede ser visto en la Tabla 22.

Tabla 22

Ejemplo de validación de requerimientos con stakeholders

RQ-ID	Modelo de validación	Stakeholders	Fecha	Resultado
RQ-00001	Prototipo de alto nivel Presentaciones Documentación	GSH-00001	27/03/2017	Aprobado
RQ-00010	Prototipo de bajo nivel	SH-00009	30/03/2017	Rechazado. Volver a especificar.

7.3.5 Los tipos de Ensayos del software a utilizar pueden ser: ensayos de las prestaciones, funcionales/de caja negra o modelado.

Tabla 23

Ensayos del software

RQ-ID		Tipo de ensayo	
Señales de entrada			
Señales de salida			
Criterios de éxito			

7.3.6 Un ejemplo de la instanciación de la tabla anterior puede ser visto en la Tabla 24.

Tabla 24

Ejemplo de ensayos del software

RQ-ID	RQ-00001	Tipo de ensayo	Análisis estático de software
Señales de entrada		Contador de 5 segundos del reloj interno del monitor de barreras.	
Señales de salida		Datos del monitor de barreras: ID, fecha, hora y estado actual.	
Criterios de éxito		El monitor de barreras envía sus datos de manera correcta al centro de control. El tiempo de recepción de los mismos es como máximo de un segundo después del envío.	

Tabla 25

Verificación de contradicciones y seguridad de los requisitos

RQ-ID	Contradicción con RQ-ID	Cumplimiento con nivel de seguridad

7.3.7 Un ejemplo de la instanciación de la tabla anterior puede ser visto en la Tabla 26.

Tabla 26

Ejemplo de verificación de contradicciones y seguridad de los requisitos

RQ-ID	Contradicción con RQ-ID	Cumplimiento con nivel de seguridad
RQ-00001	-	Cumple
RQ-00010	RQ-00009	No cumple

Tabla 27

Verificación de los requerimientos software

RQ-ID		Fecha		Responsable	
Adecuación		Legibilidad		Trazabilidad	
Ensayos		Coherencia interna		Restricciones HW y SW	
Observaciones					

Resultados	

Tabla 28

Ejemplo de verificación de los requerimientos software

RQ-ID	RQ-00001	Fecha	25/01/2017	Responsable	Jorge Lopez
Adecuación	10	Legibilidad	8	Trazabilidad	Cumple
Ensayos	8	Coherencia interna	10	Restricciones HW y SW	Cumple
Observaciones	La redacción del requisito no es del todo clara. No se encuentran definidos todos los ensayos del software que involucra este requisito, de acuerdo a los modos de comportamiento definidos sobre el mismo.				
Resultados	Corregir los puntos especificados en las observaciones.				

8 Entregas

8.1 Las salidas generadas al aplicar el presente procedimiento general de Requisitos del Software son las siguientes:

1. El documento de Especificación de Requisitos Software (ERS), que contiene:
   a. Análisis de la documentación de entrada de la Tabla 2.
   b. Identificación de stakeholders, grupos de stakeholders y canales de comunicación definidos de la Tabla 3 y la Tabla 5.
   c. Definición de técnicas de elicitación de la Tabla 7.
   d. Requerimientos elicitados de la Tabla 9.
   e. Análisis operacional y sistémico de los requerimientos anteriores de la Tabla 12.
   f. Definición de atributos característicos de los requerimientos anteriores de la Tabla 14.
   g. Definición de modos de comportamiento del software mediante los diagramas SSTA y SFTA generados con la Figura 4.
   h. Análisis SFMEA funcional de los requerimientos de la Tabla 16.
   i. Especificaciones semi formales y formales generadas con la Figura 7 y formales generadas en el lenguaje ACSL.
   j. Glosario de términos de la Tabla 18.
2. Especificación de Ensayos del Software en Conjunto (ESC), que contiene los resultados de la definición de ensayos del software a realizar de la Tabla 23.
3. Informe de Validación y verificación de los Requisitos del Software (VRS), que contiene:
   a. Validación de los requerimientos con los stakeholders de la Tabla 21.
   b. Verificación de no contradicción y cumplimiento con los niveles de seguridad definidos para el software de la Tabla 25.
   c. Verificación de los requerimientos del software de la Tabla 27.

8.2 Para la gestión de cada uno de estos formularios debe seguirse lo indicado en el Procedimiento de Soporte para la Gestión de la Documentación.

8.3 Las identificaciones de estos formularios son las siguientes:

1. F_ERS_01 Especificación de Requisitos Software - Proyecto XX. Se realizará un registro correspondiente a la Especificación de Requisitos Software dentro de la carpeta del proyecto, que se incorporará a la carpeta /Proyectos/Nombre_del_Proyecto/Requerimientos.
2. F_ESC_01 Especificación de Ensayos del Software en Conjunto - Proyecto XX. Se realizará un registro correspondiente a la Especificación de Ensayos del Software en Conjunto dentro de la carpeta del proyecto en /Proyectos/Nombre_del_Proyecto/Requerimientos.
3. F_VRS_01 Informe de Verificación de los Requisitos del Software - Proyecto XX. Se realizará un registro correspondiente al Informe de Verificación de los Requisitos del Software del proyecto que se incorporará a la carpeta /Proyectos/Nombre_del_Proyecto/Requerimientos.

Anexo 1 - Cumplimiento de UNE-EN 50128:2012

En la siguiente tabla se describe el cumplimiento de la norma UNE-EN 50128:2012 mediante el presente procedimiento definido. En la primer columna se indica el punto de la norma a cumplimentar y en la segunda el elemento de este procedimiento que lo cumple.

Tabla 29

Cumplimiento de UNE-EN 50128:2012

Ítem de UNE-EN 50128	Elemento del procedimiento definido
7.2.1.1	Resultado del Proceso 7.1 y del Proceso 7.2
7.2.1.2	Resultado del Proceso 7.3
7.2.2	Tabla 2
7.2.3	Resultados de los Procesos 7.2 y 7.3
7.2.4.1	Resultado del Proceso 7.2
7.2.4.2	Punto 7.2.3 y Fila 2 de la Tabla 16
7.2.4.3	Punto 7.2.2
7.2.4.4 a	Tabla 9
7.2.4.4 b	Identificador de los requerimientos definido en el punto 4 de la Tabla 1

7.2.4.5	Tabla 18: glosario para unificar términos
7.2.4.6	Fila 5 de la Tabla 2 y Fila 4 y 5 de la Tabla 12
7.2.4.7	Tabla 14
7.2.4.8	Tabla 16
7.2.4.9	Filas 4 y 5 de la Tabla 16
7.2.4.10	Punto 7.2.7
7.2.4.11	Punto 7.2.5, Tabla 23 y fila 3, columna 2 de la Tabla 27
7.2.4.12	Punto 7.2.5, Tabla 23 y fila 3, columna 2 de la Tabla 27
7.2.4.13	Punto 7.2.2, Tabla 12 y Fila 2 de la Tabla 25
7.2.4.14	Cuando el impacto en la seguridad de la Tabla 16 es cero
7.2.4.15	Punto 7.2.6
7.2.4.16	Resultado del Proceso 7.3
7.2.4.17	Tabla 23
7.2.4.18	Punto 7.3.5
7.2.4.19	Tabla 23
7.2.4.20	Resultado del Proceso 7.3
7.2.4.21	Tabla 27
7.2.4.22	Tabla 27

Anexo F_ERS_01 - Tercer versión

Especificación de Requisitos Software - Proyecto XX

Evidencia de la aplicación de los Procedimientos de Gestión de Requisitos del Software V1.1

Autor del presente formulario:

Lic. Cristian Pinto Luft (UNNE)

Revisores del presente formulario:

Dr. Ing. Emanuel Irrazábal (UNNE)

Aprobó el presente formulario:

Este formulario fue realizado en el marco del Proyecto CIAA en abril de 2017 y es de libre distribución.

1 Registro de cambios y lista de distribución

1.1 Registro de cambio

Versión	Fecha	Descripción	Autor
1.0	01/04/2017	Creación del Documento	Cristian Pinto Luft
1.2	09/08/2017	Correcciones	Cristian Pinto Luft

1.2 Lista de distribución

Versión	Fecha	Entregada a	Firma
1.0	01/04/2017	Intranet	Cristian Pinto Luft

2 Tabla de contenido

3 Especificación de Requisitos Software

3.1 Análisis de documentación de entrada

Documento	Fecha de análisis	Completitud	Correctitud
Especificación de Requisitos del Sistema			
Especificación de Requisitos de Seguridad del Sistema			
Descripción de la Arquitectura del sistema			
Especificaciones de la Interfaz Externa			
Plan de Aseguramiento de Calidad del Software			
Plan de Validación del Software			
Observaciones			

3.2 Identificación de stakeholders y canales de comunicación

Identificador	Nombre y apellido	Puesto o función	Canales de comunicación

3.3 Grupos de stakeholders

Identificador de grupo	Nombre de grupo	Descripción de grupo	Miembros del grupo

3.4 Técnicas de elicitación

Técnica	Duración aproximada	Objetivos

3.5 Elicitación de requerimientos

RQ-ID		Fecha		Fuente	
Necesidad					
Motivo					
Objetivo					
Verificación					

3.6 Análisis de requerimientos

RQ-ID				
Análisis operacional	Fecha		Responsable	
Resultados				
Análisis sistémico	Fecha		Responsable	
Resultados				

3.7 Modos de comportamiento del software

RQ-ID	
Estado inicial	
Evento	
Estado final	
Resultado	

3.8 Atributos de los requerimientos

RQ-ID		Versión		Responsable	
Tipo		Subtipo		Estado	
Prioridad		Esfuerzo		Impacto en la seguridad	
Restricciones de HW					
Restricciones de SW					

3.9 Árbol SSTA

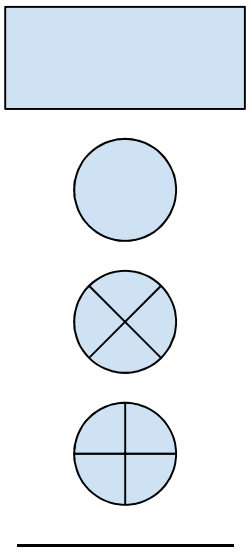


Figura 1

Símbolos básicos del árbol SSTA

3.10 Árbol SFTA

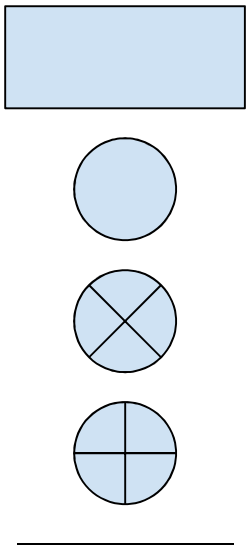


Figura 2

Símbolos básicos del árbol SFTA

3.11 Análisis SFMEA

RQ-ID/ SFMEA-ID	Modo de fallo	Efecto			Causa	Detección	Mitigación
		Local	Subsistema	Sistema			

3.12 Especificación semi formal/formal

Para la especificación semi formal, utilizar diagramas de estados, con los elementos de la Figura 3. También se pueden realizar los diagramas de clases necesarios, de manera opcional.

Figura 3

Elementos del diagrama de estados

Para la especificación semi formal, utilizar el lenguaje ACSL indicando como mínimo las pre y post condiciones. Solamente se deberán crear los contratos de funciones, no las funciones en sí, ya que las mismas se crearán en una etapa posterior del desarrollo.

3.13 Glosario de términos

Término	Definición

Anexo R_ERS_01 Monitor de Barreras - Tercer implementación

Especificación de Requisitos Software - Monitor de Barreras

Evidencia de la aplicación de los Procedimientos de Gestión de Requisitos del Software V0.4

Autor del presente formulario:

Lic. Cristian Pinto Luft (UNNE)

Revisores del presente formulario:

Dr. Ing. Emanuel Irrazábal (UNNE)

Aprobó el presente formulario:

Este formulario fue realizado en el marco del Proyecto CIAA en abril de 2017 y es de libre distribución.

1 Registro de cambios y lista de distribución

1.1 Registro de cambio

Versión	Fecha	Descripción	Autor
1.0	09/08/2017	Creación del Documento en base a los requerimientos del documento Prototipo del Monitor de Barreras a ensayar en el INTI	Cristian Pinto Luft
1.1	05/12/2017	Correcciones y adaptación a los requerimientos del documento Versión Preliminar de los Requerimientos del Monitor de Barreras Automáticas	Cristian Pinto Luft

1.2 Lista de distribución

Versión	Fecha	Entregada a	Firma
1.0	09/08/2017	Intranet	Cristian Pinto Luft

2 Tabla de contenido

3 Especificación de Requisitos Software

3.1 Análisis de documentación de entrada

Documento	Fecha de análisis	Completitud	Correctitud
Especificación de Requisitos del Sistema	09/08/2017	Versión preliminar	Versión preliminar
Especificación de Requisitos de Seguridad del Sistema	09/08/2017	De acuerdo con UNE-EN 50578	De acuerdo con UNE-EN 50578
Descripción de la Arquitectura del sistema	09/08/2017	Versión preliminar	Versión preliminar
Especificaciones de la Interfaz Externa	09/08/2017	Versión preliminar	Versión preliminar
Plan de Aseguramiento de Calidad del Software	09/08/2017	TBD	TBD
Plan de Validación del Software	09/08/2017	TBD	TBD
Observaciones	Sin cambios mayores desde la redacción del documento R_ERS_01 Especificación de Requisitos Software - Probador de relé		

3.2 Identificación de stakeholders y canales de comunicación

Identificador	Nombre y apellido	Puesto o función	Canales de comunicación
SH-0001	Ariel Lutenberg	Director GICSAFE	email: alutenberg@gmail.com tél: +54 9 11 5844-3749
SH-0002	Martín Harris	Coordinador de Desarrollos	Indirecto a partir de Ariel Lutenberg email: martin.harris@sofse.gob.ar
SH-0003	Mariano Soler	Subgerente Desarrollo y Normas Técnicas	Indirecto a partir de Ariel Lutenberg / Martín Harris email: mariano.fernandez@sofse.gob.ar
SH-0004	Adrián Laiuppa	Director del Proyecto Probador de Relé y desarrollador del sistema	email: alaiuppa@gmail.com tél: +54 9 291 643-4357

3.3 Grupos de stakeholders

Identificador de grupo	Nombre de grupo	Descripción de grupo	Miembros del grupo
GH-01	SOFSE	Representantes del desarrollo del monitor de barreras para SOFSE	SH-0001 y SH-0002

3.4 Técnicas de elicitación

Técnica	Duración aproximada	Objetivos
Prototipado mediante diagrama de estados	10 horas.	Realizar el análisis y la especificación de los requerimientos funcionales del monitor de barreras.

3.5 Elicitación de requerimientos

3.5.1 RQ-000001

RQ-ID	000001	Fecha	09/08/2017	Fuente	GH-01
Necesidad	El monitor de barreras se inicia en un estado ACTIVO y se realiza la comprobación de componentes de manera periódica. Mientras no se detecten fallas el sistema se mantendrá en estado ACTIVO con estado interno OK.				
Motivo	Controlar que los distintos componentes del sistema se encuentren funcionando correctamente durante su operación.				
Objetivo	Comprobar que el sistema se encuentre en un estado seguro, sin fallas internas de los componentes que puedan conducir a peligros.				
Verificación	• Se enciende el indicador luminoso correspondiente. • El monitor de barreras envía información relativa al estado general del sistema al centro de operaciones y la misma se recibe de manera correcta. • El sistema se encuentra en estado ACTIVO con estado interno OK.				

3.5.2 RQ-000002

RQ-ID	000002	Fecha	18/08/2017	Fuente	GH-01
Necesidad	En estado ACTIVO el monitor de barreras debe monitorizar el estado de los sensores y activar las señales luminosas de indicación de estado. De no existir fallas el sistema debe transmitir cada 10 minutos el estado del sistema de barreras, ante la aparición de un evento anómalo o ante un requerimiento del centro de control.				
Motivo	Controlar que los distintos componentes del sistema se encuentren funcionando correctamente durante su operación.				
Objetivo	Mantener el sistema en un estado seguro, sin fallas internas de los componentes que puedan conducir a peligros.				
Verificación	• Se enciende el indicador luminoso correspondiente. • El monitor de barreras envía información relativa al estado general del sistema al centro de operaciones y la misma se recibe de manera correcta cada 10 minutos en estado ACTIVO o ante un requerimiento del centro de control.				

3.5.3 RQ-000003

RQ-ID	000003	Fecha	12/09/2017	Fuente	GH-01
Necesidad	En estado ACTIVO el sistema debe comprobar que al activarse la señal de ocupación de vía la barrera se baja en un tiempo acorde al esperado (lo que se determina a partir de las señales de barrera arriba y barrera abajo, entre 5 y 20 segundos) y que se activan la sirena y las luces.				
Motivo	Controlar que los componentes de señalización del sistema responden adecuadamente al ocuparse la vía.				
Objetivo	Verificar que el sistema opera en un estado seguro, señalizando correctamente en uno de los momentos más críticos, al ocuparse la vía, evitando situaciones de peligro.				
Verificación	● El tiempo que tarda en bajarse la barrera es el predefinido (entre 5 y 20 segundos). ● Se activan las sirenas correspondientes. ● Se activan las luces correspondientes.				

3.5.4 RQ-000004

RQ-ID	000004	Fecha	14/09/2017	Fuente	GH-01
Necesidad	Al desactivarse la señal de ocupación de vía el sistema debe comprobar que la barrera sube y que se desactivan la sirena y las luces.				
Motivo	Controlar que los componentes de señalización del sistema responden adecuadamente al desocuparse la vía.				
Objetivo	Verificar que el sistema opera en un estado seguro, señalizando correctamente al desocuparse la vía, permitiendo el tránsito normal.				
Verificación	● El tiempo que tarda en subirse la barrera es el predefinido. ● Se desactivan las sirenas correspondientes. ● Se desactivan las luces correspondientes.				

3.5.5 RQ-000005

RQ-ID	000005	Fecha	09/12/2017	Fuente	GH-01
Necesidad	Al estado de ACTIVO con estado interno ERROR se ingresa ante un evento de error interno relacionado con los componentes de barrera: componentes apagados, componentes desconectados o falta de energización de la barrera.				
Motivo	Indicar con un estado de error la ocurrencia de alguna falla estando el sistema en estado ACTIVO.				
Objetivo	Poder conocer cuándo y por qué el sistema entra en un estado de error para poder tomar medidas y evitar situaciones de peligro.				
Verificación	• El monitor de barreras ingresa a un estado de error al ocurrir una falla en alguno de los componentes de la barrera y lo informa mediante una señal al centro de control.				

3.5.6 RQ-000006

RQ-ID	000006	Fecha	14/09/2017	Fuente	GH-01
Necesidad	Al estado de ACTIVO con estado interno ALARMA se ingresa ante un evento anómalo relacionado con: temperatura, tensión de batería, suministro de energía, salida del rectificador, tiempo de ocupación de vía excesivo, vía ocupada con brazo elevado, tiempo de brazo bajo excesivo, brazo roto, apertura de abrigo, corriente de motor, corriente de lámpara, frecuencia de corriente de lámpara, corriente de campana, frecuencia de corriente de campana, tiempo de ocupación de vía y bajada de brazo.				
Motivo	Indicar con un estado interno de ALARMA la ocurrencia de alguna falla estando el sistema en estado ACTIVO.				
Objetivo	Poder conocer cuándo y por qué el sistema entra en un estado de error para poder tomar medidas y evitar situaciones de peligro.				
Verificación	• El monitor de barreras ingresa a un estado ACTIVO con estado interno de ALARMA al detectar una anomalía en las señales monitorizados.				

3.5.7 RQ-000007

RQ-ID	000007	Fecha	06/12/2017	Fuente	GH-01
Necesidad	El Monitor de Barreras deberá llevar por tiempo limitado un registro local en memoria SD de las señales. El tiempo predeterminado debe ser de 30 minutos, configurable.				
Motivo	Disponer de información histórica (un log) de los últimos estados de las señales monitorizadas.				
Objetivo	Poder realizar un análisis estadístico de los últimos estados de los componentes para ayudar a prevenir fallas y evitar situaciones de peligro.				
Verificación	• El monitor de barreras mantiene un registro de manera local de las últimas señales monitorizadas en un intervalo de tiempo definido, tras el cual las mismas son borradas de este.				

3.5.8 RQ-000008

RQ-ID	000008	Fecha	06/12/2017	Fuente	GH-01
Necesidad	El monitor pasa a un estado EN CONFIGURACIÓN durante el mantenimiento. Mientras el sistema se encuentre en este estado se desactivan las alarmas.				
Motivo	Desactivar el envío de alarmas durante la configuración del sistema				
Objetivo	Diferenciar mantenimiento de vandalismo, generando una alarma.				
Verificación	• El sistema no emite alarmas durante el mantenimiento. • El sistema vuelve a estar activo con una configuración correcta.				

3.6 Análisis de requerimientos

3.6.1 RQ-000001

RQ-ID	000001			
Análisis operacional	Fecha	09/08/2017	Responsable	Cristian Pinto Luft
Resultados	Se comprobarán la barrera, los brazos, el semáforo PaN, la puerta del gabinete (abrigo), los relés de accionamiento, la energía y la temperatura del habitáculo del monitor de barreras, el suministro de energía primario, los rectificadores, los motores, la lámpara y la campana. Las comprobaciones hechas serán enviadas mediante una interfaz de comunicaciones Bluetooth o WiFi hacia el centro de operaciones.			
Análisis sistémico	Fecha	09/08/2017	Responsable	Cristian Pinto Luft
Resultados	● La comprobación se llevará a cabo cuando el sistema se encuentre en estado ACTIVO. De finalizarse con éxito, el estado cambiará a ACTIVO con estado interno OK. ● Los resultados se representarán con el formato JSON y serán enviados mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa correspondiente.			

3.6.2 RQ-000002

RQ-ID	000002			
Análisis operacional	Fecha	18/08/2017	Responsable	Cristian Pinto Luft
Resultados	Se comprobarán la barrera, los brazos, el semáforo PaN, la puerta del gabinete (abrigo), los relés de accionamiento, la energía y la temperatura del habitáculo del monitor de barreras, el suministro de energía primario, los rectificadores, los motores, la lámpara y la campana cada 10 segundos. Las comprobaciones hechas serán enviadas mediante una interfaz de comunicaciones Bluetooth o WiFi hacia el centro de control cada 10 segundos, al producirse una falla o bajo requerimiento del centro de control.			
Análisis sistémico	Fecha	18/08/2017	Responsable	Cristian Pinto Luft
Resultados	● La comprobación se llevará a cabo cuando el sistema se encuentre en estado ACTIVO. ● Los resultados se representarán con el formato JSON y serán enviados mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa correspondiente.			

3.6.3 RQ-000003

RQ-ID	000003			
Análisis operacional	Fecha	12/09/2017	Responsable	Cristian Pinto Luft

Resultados	Al activarse la señal de ocupación de vía se comprobará el tiempo que tarda en bajar la barrera y la activación de las sirenas y las luces correspondientes.			
Análisis sistémico	Fecha	12/09/2017	Responsable	Cristian Pinto Luft
Resultados	• La comprobación se llevará a cabo cuando el sistema se encuentre en estado ACTIVO con estado interno OK y se active la señal de ocupación de vía. • Los resultados se representarán con el formato JSON y serán enviados mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa y sonora correspondiente.			

3.6.4 RQ-000004

RQ-ID	000004			
Análisis operacional	Fecha	14/09/2017	Responsable	Cristian Pinto Luft
Resultados	Al desactivarse la señal de ocupación de vía se comprobará el tiempo que tarda en subir la barrera y la desactivación de las sirenas y las luces correspondientes.			
Análisis sistémico	Fecha	14/09/2017	Responsable	Cristian Pinto Luft
Resultados	• La comprobación se llevará a cabo cuando el sistema se encuentre en estado ACTIVO con estado interno OK y se desactive la señal de ocupación de vía. • Los resultados se representarán con el formato JSON y serán enviados mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa y sonora correspondiente (ausencia de señalización).			

3.6.5 RQ-000005

RQ-ID	000005			
Análisis operacional	Fecha	14/09/2017	Responsable	Cristian Pinto Luft
Resultados	Al ocurrir una falla en alguno de los componentes monitoreados por el monitor de barreras, el mismo ingresará a un estado de ACTIVO con estado interno ERROR y enviará información relativa al fallo. La información será enviada mediante una interfaz de comunicaciones Bluetooth o WiFi hacia el centro de operaciones.			
Análisis sistémico	Fecha	14/09/2017	Responsable	Cristian Pinto Luft
Resultados	• El cambio de estado y notificación del error se llevará a cabo cuando, estando el sistema en estado ACTIVADO con estado interno OK u ALARMA paso al estado ACTIVADO con estado interno ERROR, se detecte algún error en alguno de los componentes de la barrera. • Los resultados se representarán con el formato JSON y serán enviados mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa y sonora correspondiente (ausencia de señalización).			

3.6.6 RQ-000006

RQ-ID	000006			
Análisis operacional	Fecha	14/09/2017	Responsable	Cristian Pinto Luft
Resultados	Al detectarse una valor fuera de rango de las señales monitoreados por el monitor de barreras, el mismo ingresará a un estado de ACTIVO con estado interno ALARMA y enviará información relativa a dicha situación. La información será enviada mediante una interfaz de comunicaciones Bluetooth o WiFi hacia el centro de operaciones.			
Análisis sistémico	Fecha	14/09/2017	Responsable	Cristian Pinto Luft
Resultados	• El cambio de estado y notificación del peligro se llevará a cabo cuando, estando el sistema en estado ACTIVADO con estado interno OK paso al estado ACTIVADO con estado interno ALARMA, se detecte algún valor anómalo en los rangos de las señales monitorizadas. • Los resultados se representarán con el formato JSON y serán enviados mediante una interfaz Bluetooth o WiFi, además de informarse mediante la señal luminosa y sonora correspondiente (ausencia de señalización).			

3.6.7 RQ-000007

RQ-ID	000007			
Análisis operacional	Fecha	06/12/2017	Responsable	Cristian Pinto Luft
Resultados	El monitor de barreras mantendrá un registro histórico de las últimas señales monitoreadas de los distintos componentes durante un intervalo de tiempo predefinido y configurable. Tras haber superado dicho intervalo de tiempo, las señales que caigan fuera del mismo serán borradas. Este registro deberá poder ser consultado desde el centro de control en cualquier momento para obtener dicha información.			
Análisis sistémico	Fecha	06/12/2017	Responsable	Cristian Pinto Luft
Resultados	• Luego de cada lectura de monitoreo de las señales de uno o varios componentes, los resultados serán almacenados en el almacenamiento local del monitor de barreras (tarjeta SD), dentro de un archivo de bases de datos SQLite. • La información almacenada podrá ser consultada en cualquier momento desde el centro de control de manera remota. • La información será almacenada durante un intervalo de tiempo predefinido y configurable (30 minutos por defecto), luego de lo cual será eliminada.			

3.6.8 RQ-000008

RQ-ID	000008			
Análisis operacional	Fecha	04/05/2018	Responsable	Cristian Pinto Luft
Resultados	El monitor de barreras desactivará las alarmas al ingresar al estado EN CONFIGURACIÓN, permitiendo el mantenimiento y configuración del sistema. Se dejará registrado el momento en que se realice esta transición de estados así como las operaciones llevadas a cabo en el mismo.			
Análisis sistémico	Fecha	04/05/2018	Responsable	Cristian Pinto Luft
Resultados	• El sistema ingresará al estado EN CONFIGURACIÓN estando en estado ACTIVO, al recibirse la señal de “aviso de mantenimiento” desde el centro de control. • Al ingresar al estado EN CONFIGURACIÓN, el sistema desactivará todas sus alarmas. • Al finalizar el proceso de configuración y mantenimiento, se indicará con una señal de “configuración OK” desde el centro de control, y el sistema volverá a pasar al estado ACTIVO con sub estado interno OK, volviendo a activar todas sus alarmas. • Todos los cambios realizados mientras el sistema se encuentre en estado EN CONFIGURACIÓN serán registrados y almacenados en el almacenamiento local (tarjeta SD) del mismo.			

3.7 Atributos de los requerimientos

3.7.1 RQ-000001

RQ-ID	000001	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	40 horas	Impacto en la seguridad	9
Restricciones de HW	● El sistema debe ser alimentado con energía de manera contínua durante el proceso de comprobación. ● Los distintos componentes a controlarse deben estar correctamente conectados y configurados en el sistema para su comprobación. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.				
Restricciones de SW	● El sistema debe encontrarse en estado ACTIVO con estado interno OK.				

3.7.2 RQ-000002

RQ-ID	000002	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	40 horas	Impacto en la seguridad	9
Restricciones de HW	● El sistema debe ser alimentado con energía de manera contínua durante todo el proceso. ● Los distintos componentes a controlarse deben estar correctamente conectados y configurados en el sistema para su comprobación. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.				
Restricciones de SW	● El sistema debe encontrarse en estado ACTIVO con estado interno OK.				

3.7.3 RQ-000003

RQ-ID	000003	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	40 horas	Impacto en la seguridad	9
Restricciones de HW	● La señal de ocupación de vía debe estar activada. ● El sistema debe ser alimentado con energía de manera contínua durante todo el proceso. ● La barrera, las sirenas y las luces deben estar correctamente conectadas y configurados en el sistema para su comprobación. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.				
Restricciones de SW	● El sistema debe encontrarse en estado ACTIVO con estado interno OK.				

3.7.4 RQ-000004

RQ-ID	000004	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	6	Esfuerzo	20 horas	Impacto en la seguridad	6
Restricciones de HW	● La señal de ocupación de vía debe estar desactivada. ● El sistema debe ser alimentado con energía de manera contínua durante todo el proceso. ● La barrera, las sirenas y las luces deben estar correctamente conectadas y configurados en el sistema para su comprobación. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.				
Restricciones de SW	● El sistema debe encontrarse en estado ACTIVO con estado interno OK.				

3.7.5 RQ-000005

RQ-ID	000005	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	20 horas	Impacto en la seguridad	10
Restricciones de HW	● Debe existir un error, ya sea físico o lógico. ● El sistema debe ser alimentado con energía de manera contínua durante todo el proceso. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.				
Restricciones de SW	● El sistema debe encontrarse en estado ACTIVO con estado interno ERROR.				

3.7.6 RQ-000006

RQ-ID	000006	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	20 horas	Impacto en la seguridad	10
Restricciones de HW	● Debe existir una lectura de un valor anómalo, fuera de sus rangos predefinidos. ● El sistema debe ser alimentado con energía de manera contínua durante todo el proceso. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.				
Restricciones de SW	● El sistema debe encontrarse en estado ACTIVO con estado interno ALARMA.				

3.7.7 RQ-000007

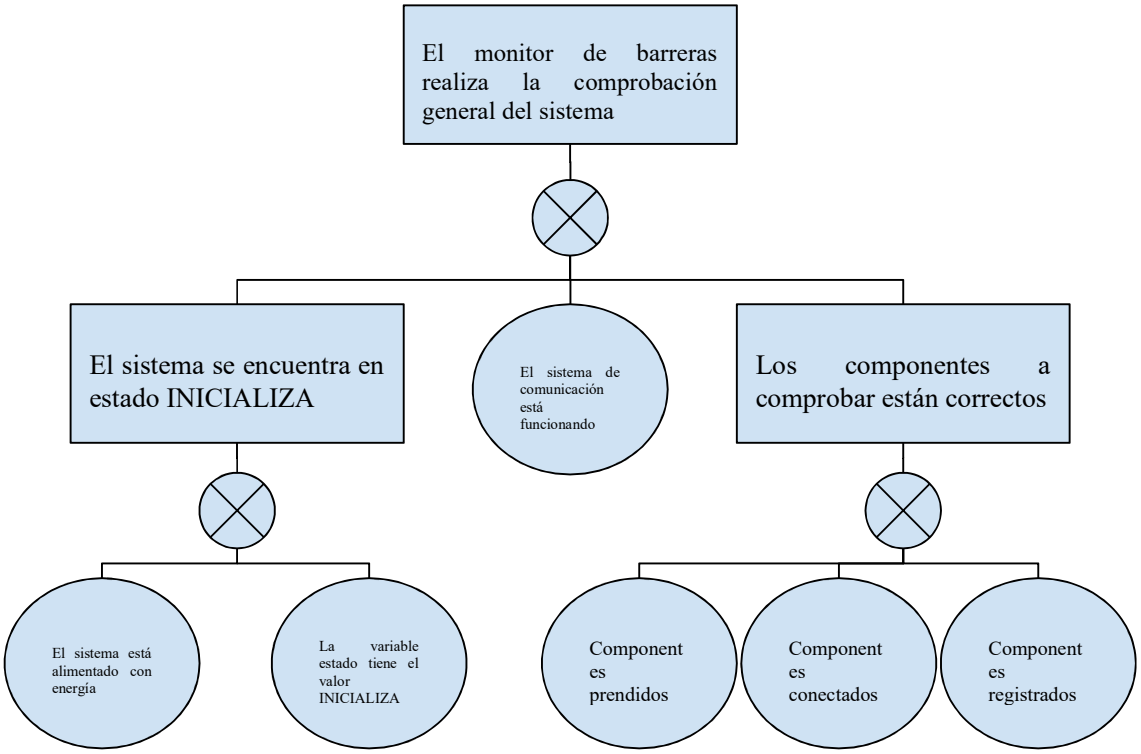
RQ-ID	000007	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	8	Esfuerzo	20 horas	Impacto en la seguridad	8
Restricciones de HW	● El monitor de barreras debe tener instalada una tarjeta SD con la capacidad de almacenamiento y velocidad suficientes para almacenar la información y poder consultarla, en los tiempos requeridos. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de comunicar la información solicitada.				
Restricciones de SW	● El sistema debe tener configurada una BD SQLite con sus tablas bien definidas para poder almacenar la información. ● El sistema debe eliminar la información histórica antigua basándose en su reloj interno, cada vez que se supere el intervalo de tiempo definido (30 minutos por defecto).				

3.7.8 RQ-000008

RQ-ID	000008	Versión	1.0	Responsable	Cristian Pinto Luft
Tipo	Funcional	Subtipo	-	Estado	En análisis
Prioridad	10	Esfuerzo	20 horas	Impacto en la seguridad	10
Restricciones de HW	● Se deben desactivar las alarmas del sistema. ● El sistema debe ser alimentado con energía de manera contínua durante todo el proceso. ● Los sistemas de comunicación de resultados deben encontrarse funcionando al momento de realizar dicha tarea.				
Restricciones de SW	● El sistema debe encontrarse en estado ACTIVO con estado interno EN CONFIGURACIÓN. ● Se deben poder configurar los parámetros de los componentes.				

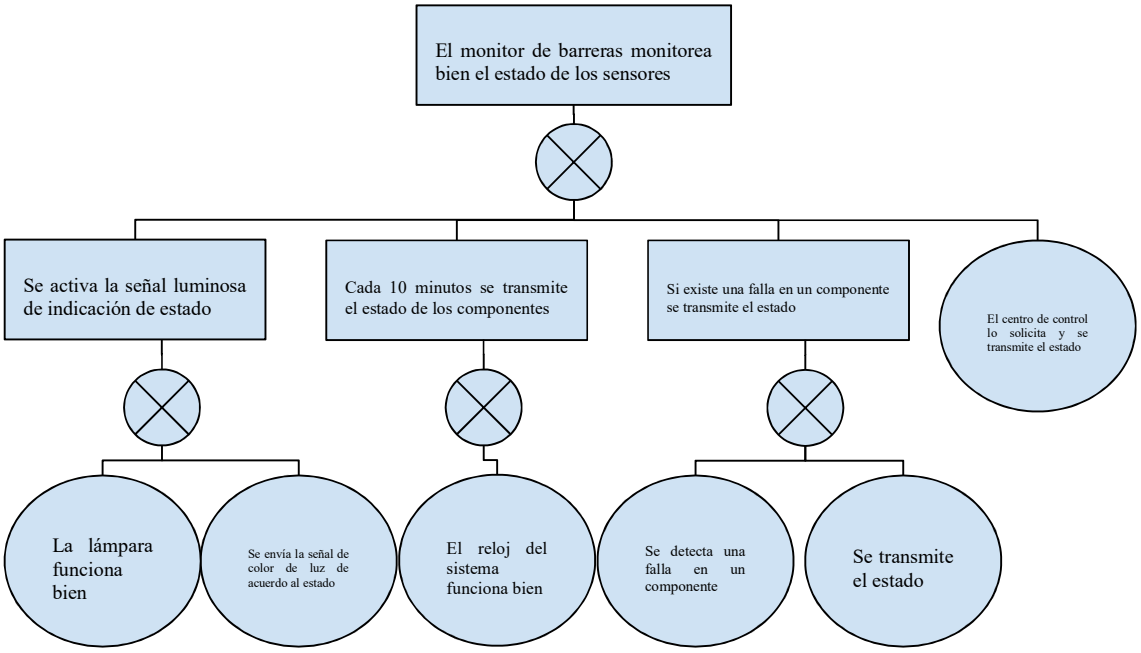
3.8 Árbol SSTA

3.8.1 RQ-000001

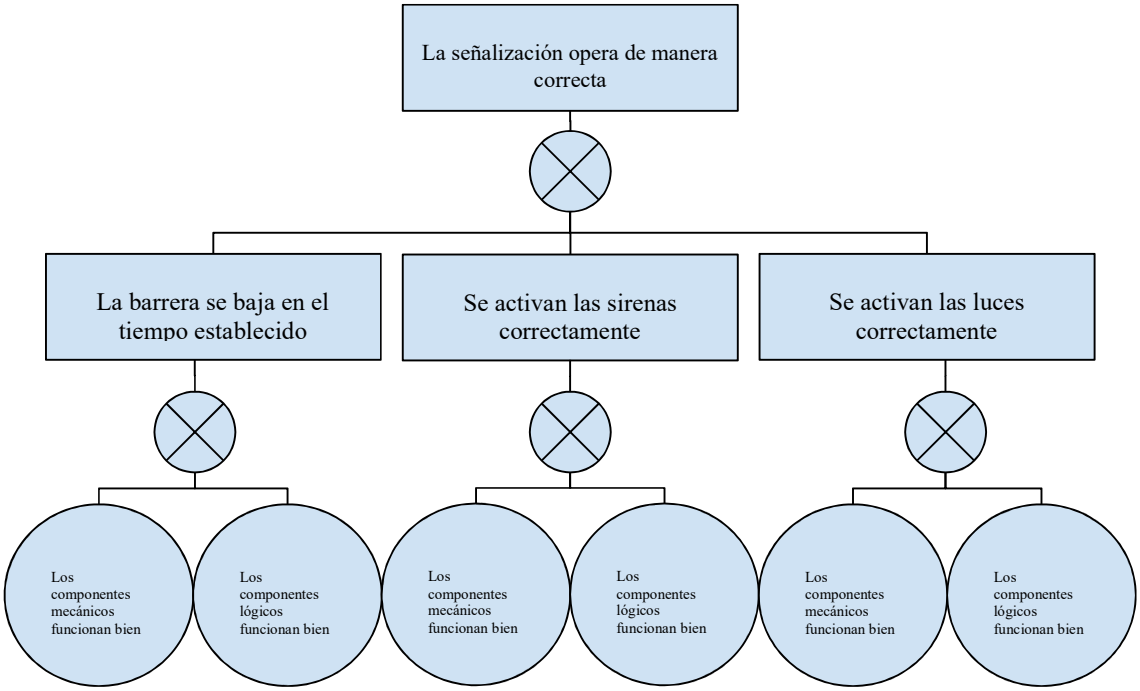


3.8.2 RQ-000002

Para el árbol SSTA del RQ-000002 se aplica el árbol SSTA del RQ-000001, solo que no se lo vuelve a escribir por una cuestión de optimización del espacio y no redundancia. El único cambio es el estado del sistema y de la variable *estado*, que en este caso tienen el valor ACTIVO y su estado interno es OK.



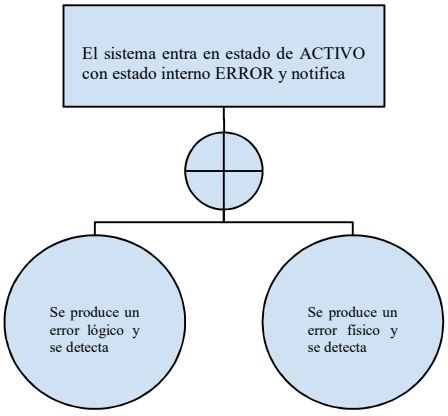
3.8.3 RQ-000003



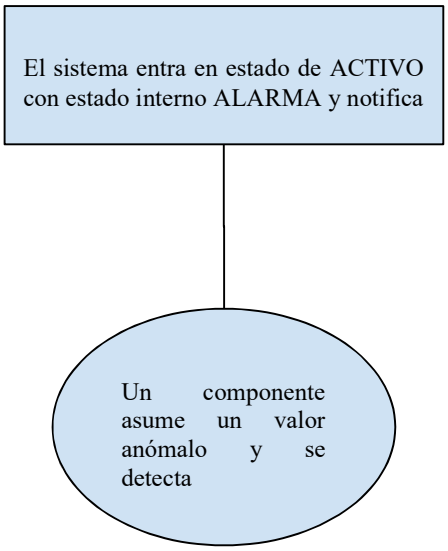
3.8.4 RQ-000004

En este caso, el árbol SSTA es idéntico al del punto 3.8.3, cambiándose solamente el evento de bajada de la barrera por el de subida, y la activación de sirenas y alarmas por la desactivación de las mismas.

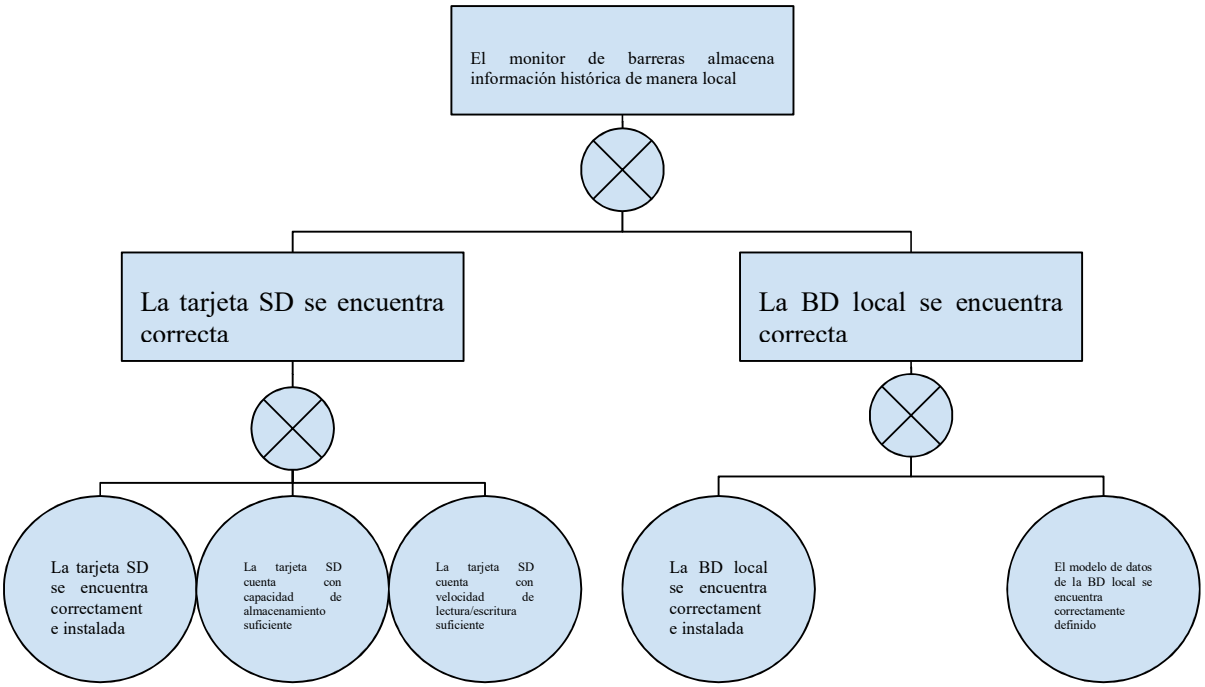
3.8.5 RQ-000005



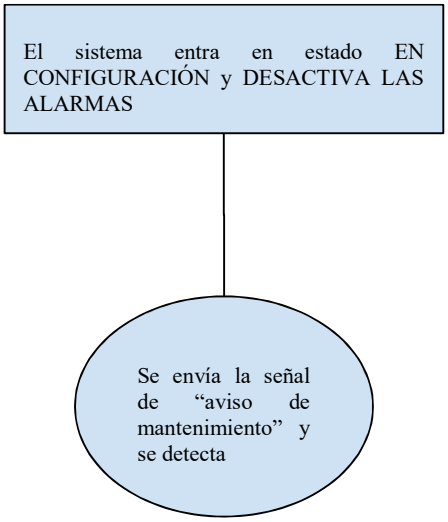
3.8.6 RQ-000006



3.8.7 RQ-000007

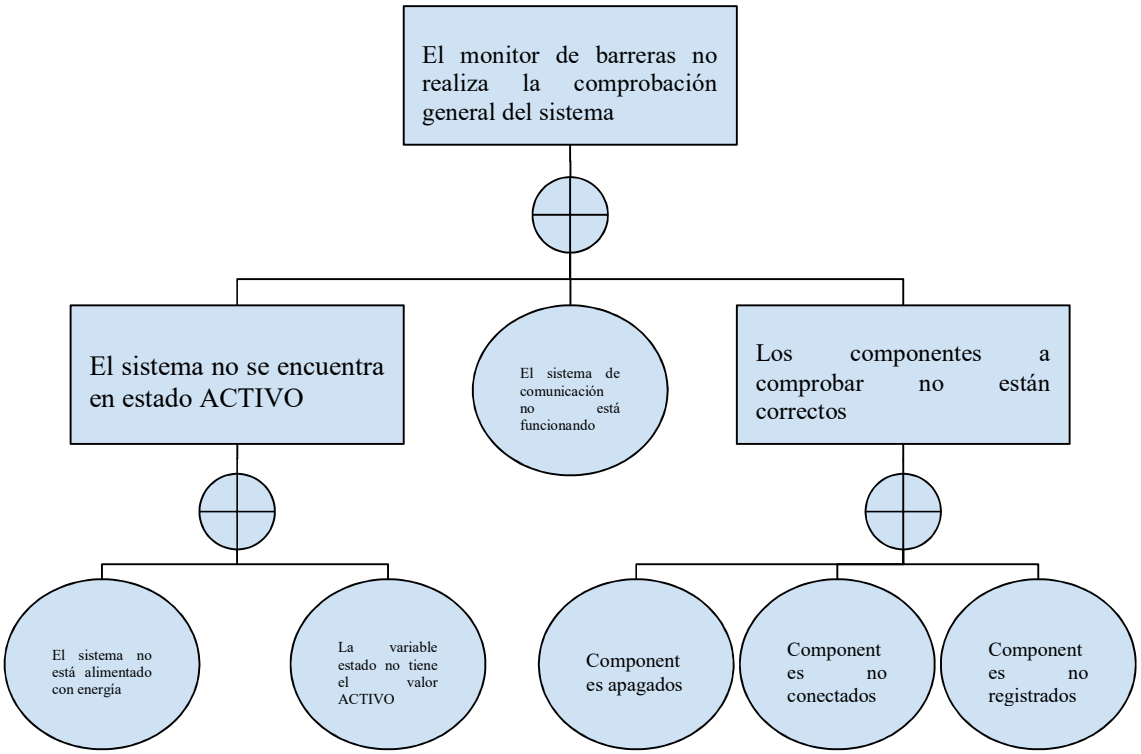


3.8.8 RQ-000008

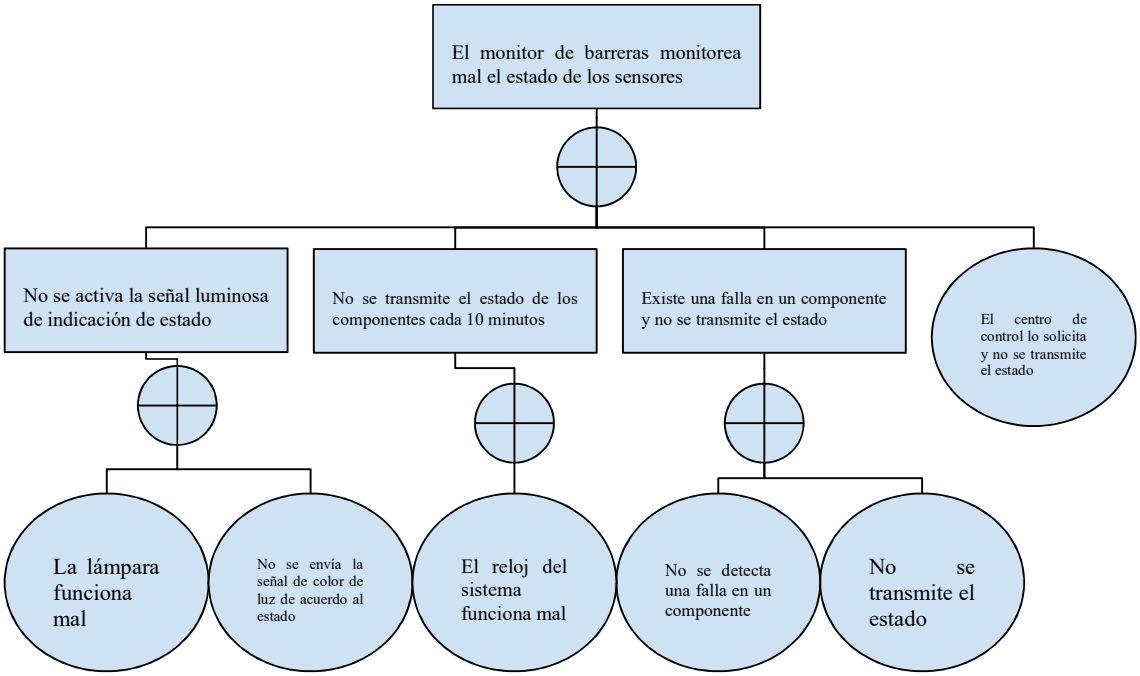


3.9 Árbol SFTA

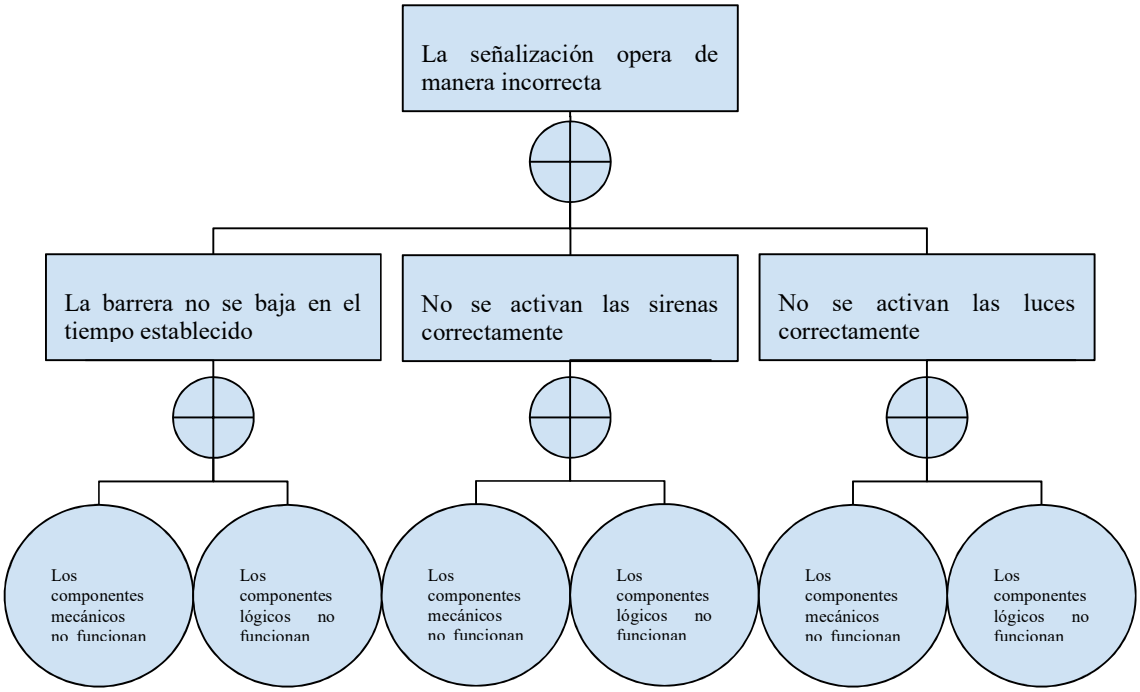
3.9.1 RQ-000001



3.9.2 RQ-000002



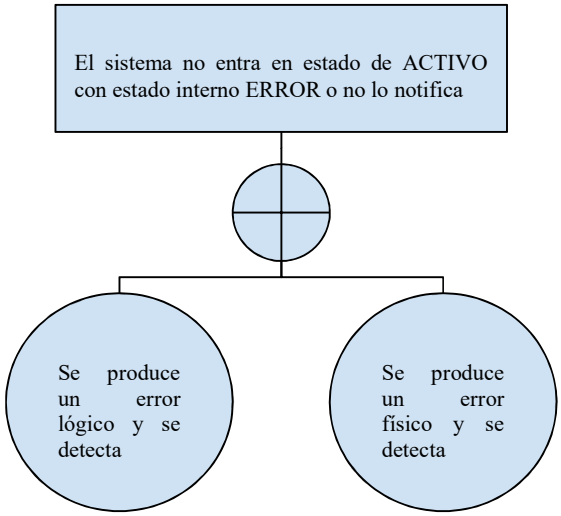
3.9.3 RQ-000003



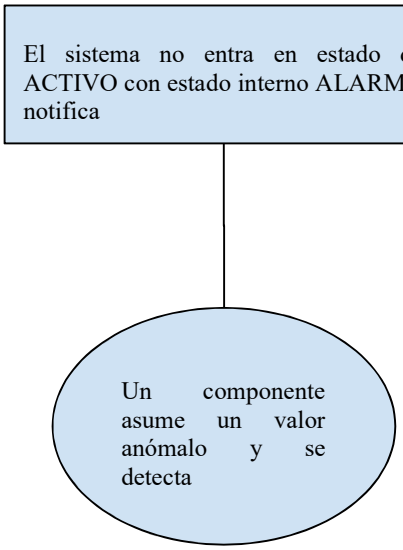
3.9.4 RQ-000004

En este caso, el árbol SFTA es idéntico al del punto 3.9.3, cambiándose solamente el evento de bajada de la barrera por el de subida, y la activación de sirenas y alarmas por la desactivación de las mismas.

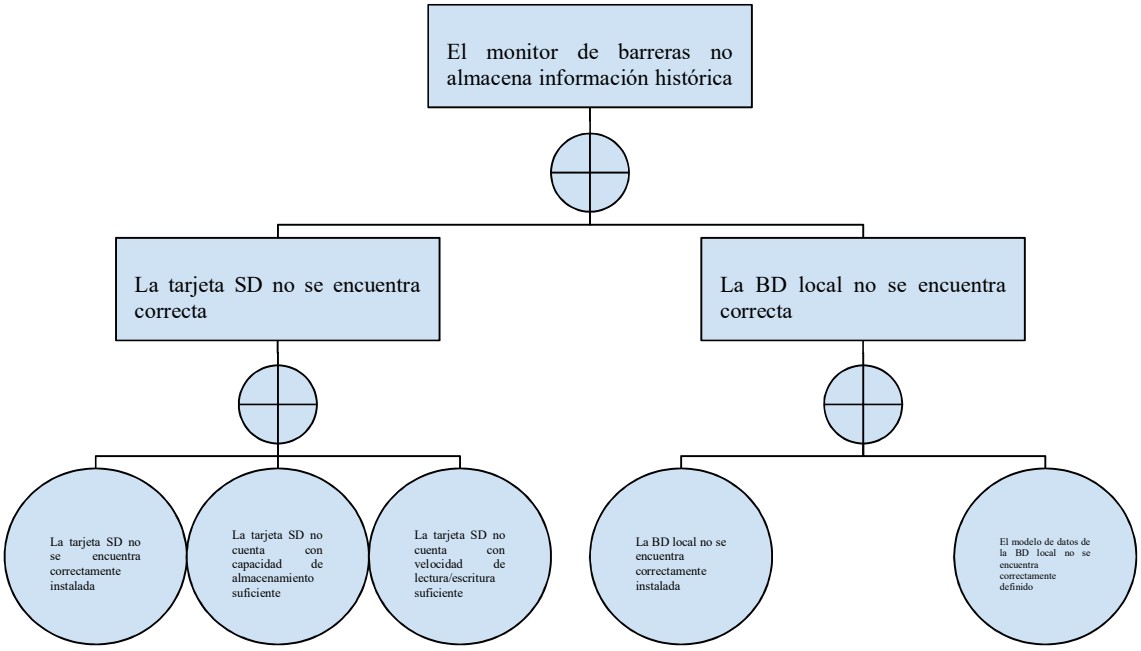
3.9.5 RQ-000005



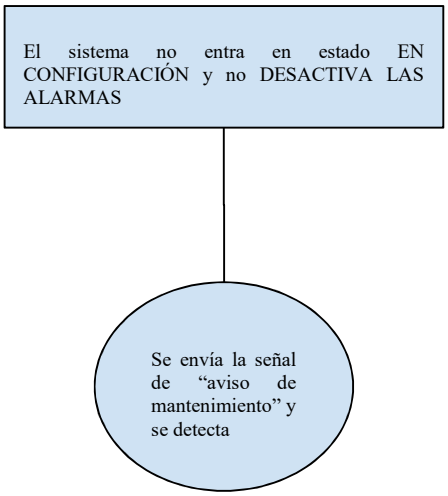
3.9.6 RQ-000006



3.9.7 RQ-000007



3.9.8 RQ-000008



3.10 Análisis SFMEA

3.10.1 RQ-000001

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000001				
Modo de fallo		El sistema no está alimentado con energía				
Efecto	**Local**	No se pueden realizar comprobaciones				
	Subsistema	El sistema de monitor de barreras no funciona				
	Sistema	Peligro de falta de señalización en el PaN				
Causas		• El sistema fué apagado intencionalmente • El sistema fué apagado no intencionalmente				
Detección		No se reciben señales del monitor de barreras desde el centro de control				
Mitigación		Si fué apagado no intencionalmente, se intenta prender el sistema de manera remota, y de no funcionar se envía a un operario a que lo prenda manualmente				
Prevención		• Se mantiene al monitor de barreras con redundancia de fuentes de energía. • Se utilizan baterías con reserva de energía para suministrar al monitor de barreras hasta un día.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000002
Modo de fallo		La variable estado no tiene el valor ACTIVO ni su estado interno OK
Efecto	**Local**	No se realiza la comprobación inicial
	Subsistema	El sistema de monitor de barreras se encuentra en estado EN CONFIGURACIÓN o apagado.
	Sistema	-
Causas		• El sistema está apagado • El sistema se encuentra en estado EN CONFIGURACIÓN
Detección		Chequeo del estado del monitor de barreras desde el centro de control

Mitigación	-				
Prevención	-				
Severidad	Menor	**Frecuencia**	Ocasional	**Riesgo**	Aceptable

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000003				
Modo de fallo		El sistema de comunicación no está funcionando				
Efecto	**Local**	No se pueden comunicar los resultados de las comprobaciones ni se pueden recibir órdenes del centro de operaciones				
	Subsistema	El sistema de monitor de barreras no puede comunicar sus datos ni recibir órdenes				
	Sistema	Desconocimiento del estado de la barrera y sus componentes				
Causas		Falla del sistema de comunicación o de uno de sus componentes (HW/SW)				
Detección		No se reciben señales del monitor de barreras desde el centro de control				
Mitigación		Se intenta restablecer el sistema de comunicación a distancia, y de no ser posible se envía personal especializado hasta el lugar.				
Prevención		Se mantiene al monitor de barreras con redundancia de canales de comunicación				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000004				
Modo de fallo		Componentes apagados				
Efecto	**Local**	No se pueden comprobar los componentes apagados				
	Subsistema	No se pueden utilizar de los componentes apagados				
	Sistema	Peligro de falta de señalización				
Causas		● Falla de HW/SW del componente que no le permite arrancar ● Apagado intencional de un componente por mantenimiento				

Detección	● Al realizar la comprobación, luego de N segundos el monitor de barreras no recibe respuesta del estado del componente. ● El monitor de barreras informa al centro de control sobre los componentes no disponibles.				
Mitigación	Se intenta prender el componente a distancia, y de no ser posible se envía personal especializado hasta el lugar.				
Prevención	Se mantiene al monitor de barreras con redundancia de componentes críticos que no se deberían apagar				
Severidad	Crítico	**Frecuencia**	Ocasional	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000001/SFMEA-000005				
Modo de fallo		Componentes no conectados				
Efecto	**Local**	No se pueden comprobar los componentes desconectados				
	Subsistema	No se pueden utilizar de los componentes desconectados				
	Sistema	Peligro de falta de señalización				
Causas		● Desconexión intencional de un componente por mantenimiento ● Desconexión intencional de un componente por vandalismo				
Detección		● Al realizar la comprobación, luego de 2 segundos el monitor de barreras no recibe respuesta del estado del componente. ● El monitor de barreras informa al centro de control sobre los componentes no disponibles.				
Mitigación		Se envía personal especializado hasta el lugar para conectar los componentes desconectados.				
Prevención		● Cada vez que el personal técnico realiza un mantenimiento, llena una planilla de comprobación de conexión correcta de los componentes principales. ● Se mantienen los componentes del monitor de barreras bien protegidos, sin acceso por parte del público a los mismos.				
Severidad		Crítico	**Frecuencia**	Remoto	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID	RQ-000001/SFMEA-000006
Modo de fallo	Componentes no registrados

Efecto	**Local**	No se pueden comprobar los componentes no registrados				
	Subsistema	No se conoce el estado de los componentes no registrados				
	Sistema	Peligro de falta de señalización				
Causas		Falta de registración en el software del monitor de barreras de un nuevo componente luego de haberlo conectado y prendido.				
Detección		El monitor de barreras no envía ningún tipo de información del componente en la señal de comprobación enviada al centro de control.				
Mitigación		Se registra correctamente el componente en el software del monitor de barreras mediante el software definido.				
Prevención		Cada vez que se añade, se reemplaza o se conecta un nuevo componente, se debe verificar que el mismo haya sido registrado correctamente.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

3.10.2 RQ-000002

RQ-ID/SFMEA-ID		RQ-000002/SFMEA-000001				
Modo de fallo		La lámpara funciona mal				
Efecto	**Local**	No se tiene certeza de estar prendiendo o apagando la luz correctamente				
	Subsistema	Incertidumbre sobre el estado de las luces				
	Sistema	Peligro de falta de señalización				
Causas		Falla física de la lámpara				
Detección		La lámpara no emite luz o lo hace de manera intermitente o errónea				
Mitigación		Se envía personal técnico a reparar o reemplazar la lámpara				
Prevención		En la comprobación de los componentes de la fase INICIALIZA se debe comprobar también, entre ellos, las luces.				
Severidad		Crítico	**Frecuencia**	Ocasional	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000002/SFMEA-000002
Modo de fallo		No se envía la señal de color de luz de acuerdo al estado
Efecto	**Local**	Indicación de estado errónea
	Subsistema	Incertidumbre sobre el estado de los componentes
	Sistema	Peligro de falta de señalización
Causas		● Lectura errónea del estado del componente conduce a indicar con las luces que está correcto cuando se encuentra erróneo o viceversa. ● Lectura correcta del estado del componente y falla lógica de la emisión de luz del color correspondiente.
Detección		Se recibe información de estado del componente fallado mientras que la luz indica que se encuentra correcto, o viceversa. Hace falta detección visual de las luces y análisis digital de la información.
Mitigación		● Se envía personal técnico a reparar o reemplazar la lámpara ● Se cambia el color de la luz de forma manual desde el SW.

Prevención	● En la comprobación de los componentes de la fase INICIALIZA se debe comprobar también, entre ellos, las luces. ● Se debe realizar una comprobación cruzada visual y digital cada vez que se activa el sistema para evitar errores de inicialización de estados y de luces.				
Severidad	Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000002/SFMEA-000003				
Modo de fallo		El reloj del sistema funciona mal				
Efecto	**Local**	Tiempos de envío/recepción de mensajes incorrectos/inconsistentes				
	Subsistema	Información sobre el estado de los componentes temporalmente errónea				
	Sistema	Peligro de falta de sincronía				
Causas		● Agotamiento/falla de las fuentes de energía del reloj. ● Falla o avería física del reloj. ● Retraso/adelantamiento del tiempo del reloj con respecto al del sistema debido a una falla del software.				
Detección		Se recibe información de estado del monitor de barreras con una marca de tiempo (timestamp) diferente a la del sistema, o en intervalos irregulares (distintos de 10 minutos).				
Mitigación		● Se envía personal técnico a reemplazar/reparar las fuentes de energía del reloj. ● Se envía personal técnico a reemplazar/reparar el reloj. ● Se envía un comando de re-sincronización del reloj con el tiempo del sistema desde el centro de operaciones.				
Prevención		● Se posee redundancia de fuentes de energía para el reloj. ● Se posee redundancia de relojes, en una arquitectura pasiva maestro-esclavo, pudiendo el segundo asumir el rol del primero en caso de fallar éste. ● Se envían señales de control de sincronización en intervalos regulares de tiempo desde el centro de control, en la confirmación de recepción de la señal de estado del monitor de barreras (cada 10 minutos) para mantener la sincronía entre el sistema y el sub-sistema.				
Severidad		Crítico	**Frecuencia**	Probable	**Riesgo**	N-Aceptable

RQ-ID/SFMEA-ID	RQ-000002/SFMEA-000004

Modo de fallo		No se detecta una falla en un componente				
Efecto	**Local**	Error al informar el estado de un componente fallido				
	Subsistema	Información errónea del estado de un componente				
	Sistema	Peligro de falla de componente				
Causas		● El componente fallido no informa su estado/no responde. ● El componente fallido informa incorrectamente su estado. ● No se monitorea el estado del componente.				
Detección		● En la señal de estado enviada desde el monitor de barreras el estado del componente se encuentra vacío. ● Cuando el componente fallido informa incorrectamente su estado se produce la situación más peligrosa, ya que se presenta el caso de un falso positivo, pudiendo ocurrir que se detecte la falla recién al ocurrir un incidente. ● En la señal de estado enviada desde el monitor de barreras no se encuentra el componente.				
Mitigación		Cuando en la señal de estado enviada desde el monitor de barreras el estado de un componente se encuentra vacío o no se encuentra uno de los componentes, se debe proceder a realizar una verificación del mismo desde el centro de operaciones, primero de manera lógica y de ser necesario de manera física, enviando un técnico al lugar. En el caso de que un componente fallido informe incorrectamente su estado (falso positivo) y se produzca un incidente, se deben tomar al instante todas las medidas necesarias para solventar la situación y reparar/reemplazar el componente fallido.				
Prevención		● El monitor de barreras no puede enviar una señal de estado con el estado de un componente vacío o sin un componente. En caso de que el componente no informe de su estado, no responda o no se lo monitoree, el monitor debe volver a escanear al mismo N veces más, y de persistir la situación, la señal de estado que se envía indicará un error en el estado del componente. ● Para evitar falsos positivos, el monitor de barreras debe verificar el estado de cada componente M veces. De encontrarse correcto en las M, el estado pasará a ser **correcto**, sino, se indicará una **falla**.				
Severidad		Catastrófico	**Frecuencia**	Remota	**Riesgo**	N-Aceptable

RQ-ID/SFMEA-ID		RQ-000002/SFMEA-000005
Modo de fallo		No se transmite el estado
Efecto	**Local**	Error al informar el estado de un componente

	Subsistema	Sin información temporal del estado de un componente				
	Sistema	Peligro de falla de componente				
Causas		• Error de indicación del estado de un componente: se indica **correcto** pero se encuentra en **falla**, o viceversa. • Falla de comunicación durante la transmisión: se pierden/alteran los paquetes.				
Detección		• En el caso en que se indica que el estado del componente es **correcto** pero se encuentra en **falla**, o viceversa, se presenta una situación de falso positivo, pudiendo llegar a detectarse recién al ocurrir un incidente, en el peor de los casos. • En el caso de que se pierdan paquetes durante la transmisión, esto se detecta desde el centro de operaciones por la comprobación de integridad que se realiza de los mismos al momento de recibirse. En el caso de que se alteren los paquetes, en el peor de los casos esto podría llevar a un falso positivo y detectarse recién al ocurrir un incidente.				
Mitigación		En el caso de que se pierdan/alteren paquetes durante la transmisión, y esta situación sea detectada por el receptor, se debe solicitar al instante la retransmisión de los mismos. En el caso de que un componente informe incorrectamente su estado (falso positivo) y se produzca un incidente, se deben tomar al instante todas las medidas necesarias para solventar la situación y reparar/reemplazar el componente fallido.				
Prevención		• Se deben realizar comprobaciones y mantenimientos periódicos a las interfaces de comunicación de datos de los emisores y receptores, así como también a los dispositivos de comunicación de datos intermedios. • Se debe optar por el uso de tecnologías de comunicación poco propensas a errores/ruido, como ser la fibra óptica, por sobre otras más vulnerables a estos problemas, como ser las inalámbricas.				
Severidad		Crítico	**Frecuencia**	Probable	**Riesgo**	N-Aceptable

RQ-ID/SFMEA-ID		RQ-000002/SFMEA-000006
Modo de fallo		El centro de control lo solicita y no se transmite el estado
Efecto	**Local**	Error al informar el estado de un componente
	Subsistema	Sin información temporal del estado de un componente
	Sistema	Peligro de falla de componente
Causas		• Error de indicación del estado de un componente: un componente no responde y bloquea la solicitud. • Falla de comunicación durante la transmisión: se pierden/alteran los

	paquetes.				
Detección	• En el caso en que un componente no responde y bloquea la solicitud, la respuesta llega después de mucho tiempo o nunca llega. • En el caso de que se pierdan paquetes durante la transmisión, esto se detecta desde el centro de operaciones por la comprobación de integridad que se realiza de los mismos al momento de recibirse. En el caso de que se alteren los paquetes, en el peor de los casos esto podría llevar a un falso positivo y detectarse recién al ocurrir un incidente.				
Mitigación	En el caso de que se pierdan/alteren paquetes durante la transmisión, y esta situación sea detectada por el receptor, se debe solicitar al instante la retransmisión de los mismos. En el caso de que un componente no informe su estado por no responder, se debe configurar el monitor de barreras para que trate de recuperar el estado del componente en 2 intentos de 2 segundos cada uno, y de no haber respuesta, devolver la señal con un estado de **error** para dicho componente.				
Prevención	• Se deben realizar comprobaciones y mantenimientos periódicos a las interfaces de comunicación de datos de los emisores y receptores, así como también a los dispositivos de comunicación de datos intermedios y a los componentes del sistema en general. • Se debe optar por el uso de tecnologías de comunicación poco propensas a errores/ruido, como ser la fibra óptica, por sobre otras más vulnerables a estos problemas, como ser las inalámbricas.				
Severidad	Mayor	**Frecuencia**	Remota	**Riesgo**	Aceptable

3.10.3 RQ-000003

RQ-ID/SFMEA-ID		RQ-000003/SFMEA-000001
Modo de fallo		Los componentes mecánicos funcionan mal
Efecto	**Local**	No se tiene certeza de estar señalizando correctamente
	Subsistema	Incertidumbre sobre el estado de las señales
	Sistema	Peligro de falta de señalización
Causas		Falla física de una o varias partes de los componentes mecánicos
Detección		La señalización no funciona o lo hace de manera errónea
Mitigación		Se envía personal técnico a reparar o reemplazar el componente

Prevención	Los 3 componentes deben ser verificados en estado ACTIVO, indicando de antemano en caso de existir una falla. Se debe contar con repuestos y reemplazos para un cambio rápido.				
Severidad	Catastrófico	**Frecuencia**	Remota	**Riesgo**	N-Aceptable

RQ-ID/SFMEA-ID		RQ-000003/SFMEA-000002				
Modo de fallo		Los componentes lógicos funcionan mal				
Efecto	**Local**	No se tiene certeza de estar señalizando correctamente				
	Subsistema	Incertidumbre sobre el estado de las señales				
	Sistema	Peligro de falta de señalización				
Causas		Falla lógica de uno o varios componentes				
Detección		La señalización no funciona o lo hace de manera errónea				
Mitigación		Se intenta reparar la falla del componente desde el centro de control o devolver su configuración/código a un estado anterior seguro				
Prevención		Los 3 componentes deben ser verificados en estado ACTIVO, indicando de antemano en caso de existir una falla. Se deben mantener copias de las antiguas configuraciones y código del software para poder realizar un rollback en caso de ser necesario.				
Severidad		Catastrófico	**Frecuencia**	Remota	**Riesgo**	N-Aceptable

3.10.4 RQ-000004

En este caso, el análisis SFMEA es idéntico al del punto 3.10.3.

3.10.5 RQ-000005

RQ-ID/SFMEA-ID	RQ-000005/SFMEA-000001
Modo de fallo	Se produce un error lógico y se detecta, pero el sistema no entra en estado de ACTIVO con estado interno ERROR o no lo notifica.

Efecto	**Local**	Se desconoce el estado real de la barrera				
	Subsistema	El sistema de monitor de barreras se encuentra en un estado inconsistente				
	Sistema	Peligro de señalización errónea en el PaN				
Causas		• Por un problema algorítmico el sistema no cambia correctamente de estado • Por un problema del subsistema de comunicación no se lleva a cabo la notificación del estado de error				
Detección		La detección del fallo puede ser muy dificultosa y darse principalmente por medio de la observación, debido a que se presenta una situación de falso positivo.				
Mitigación		Al haber detectado el falso positivo, informar desde el centro de control con la señalización adecuada del posible peligro y cambiar manualmente el estado del sistema, aplicando las correcciones necesarias				
Prevención		Se realizan auto comprobaciones periódicas del monitor de barreras, en intervalos de tiempo predeterminados, enviando los resultados al centro de control para su análisis automatizado.				
Severidad		Catastrófico	**Frecuencia**	Remota	**Riesgo**	N-Aceptable

RQ-ID/SFMEA-ID		RQ-000005/SFMEA-000002
Modo de fallo		Se produce un error físico y se detecta, pero el sistema no entra en estado de ERROR o no lo notifica.
Efecto	**Local**	Se desconoce el estado real de la barrera
	Subsistema	El sistema de monitor de barreras se encuentra en un estado inconsistente
	Sistema	Peligro de señalización errónea en el PaN
Causas		• Por un problema algorítmico el sistema no cambia correctamente de estado • Por un problema del subsistema de comunicación no se lleva a cabo la notificación del estado de error
Detección		La detección del fallo puede ser muy dificultosa y darse principalmente por medio de la observación, debido a que se presenta una situación de falso positivo.
Mitigación		Al haber detectado el falso positivo, informar desde el centro de control con la señalización adecuada del posible peligro y enviar con carácter urgente personal técnico especializado a reparar el error físico y de ser necesario el monitor de

	barrera				
Prevención	Se realizan auto comprobaciones periódicas del monitor de barreras, en intervalos de tiempo predeterminados, enviando los resultados al centro de control para su análisis automatizado. Se cuenta con cámaras de seguridad grabando la barrera y transmitiendo en directo hacia el centro de control, para aumentar la probabilidad de detectar visualmente un error físico si llegara a ocurrir				
Severidad	Catastrófico	**Frecuencia**	Remota	**Riesgo**	N-Aceptable

3.10.6 RQ-000006

RQ-ID/SFMEA-ID		RQ-000006/SFMEA-000001				
Modo de fallo		Se produce un valor anómalo en un componente y se detecta, pero el sistema no entra en estado de ACTIVO con estado interno ALARMA o no lo notifica.				
Efecto	**Local**	Se desconoce el estado real de los componentes de la barrera				
	Subsistema	El sistema de monitor de barreras se encuentra en un estado inconsistente				
	Sistema	Peligro de señalización errónea en el PaN				
Causas		• Por un problema algorítmico el sistema no cambia correctamente de estado • Por un problema del subsistema de comunicación no se lleva a cabo la notificación del estado de alarma				
Detección		La detección del fallo puede ser muy dificultosa y darse principalmente por medio de la observación, debido a que se presenta una situación de falso positivo.				
Mitigación		Al haber detectado el falso positivo, informar desde el centro de control con la señalización adecuada del posible peligro y cambiar manualmente el estado del sistema, aplicando las correcciones necesarias				
Prevención		Se realizan auto comprobaciones periódicas del monitor de barreras, en intervalos de tiempo predeterminados, enviando los resultados al centro de control para su análisis automatizado.				
Severidad		Catastrófico	**Frecuencia**	Remota	**Riesgo**	N-Aceptable

3.10.7 RQ-000007

RQ-ID/SFMEA-ID		RQ-000007/SFMEA-000001				
Modo de fallo		La tarjeta SD no se encuentra correctamente instalada.				
Efecto	**Local**	No se puede tener acceso a la tarjeta de memoria.				
	Subsistema	No se puede utilizar información histórica para control del subsistema.				
	Sistema	Peligro de imposibilidad de detección de errores en el PaN				
Causas		● La tarjeta no fué instalada físicamente de manera correcta. ● La tarjeta no se instaló con el formato lógico correcto. ● La tarjeta quedó bloqueada por algún inconveniente.				
Detección		● No se puede acceder a la tarjeta de memoria desde el centro de control. ● No se puede acceder a la tarjeta de memoria desde el monitor de barreras.				
Mitigación		● Al intentar acceder a datos históricos desde el centro de control, en caso de informarse un error, enviar a un técnico para su corrección. ● Al intentar escribir o leer datos desde el monitor de barreras a su tarjeta, en caso de encontrar algún error, informar al centro de control incorporando la información a la señal de estado que se envía.				
Prevención		Al realizar la autocomprobación periódica de los componentes del monitor de barreras, incluir entre estos a la tarjeta SD e informar de su estado.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000007/SFMEA-000002
Modo de fallo		La tarjeta SD no cuenta con capacidad de almacenamiento suficiente.
Efecto	**Local**	No se pueden almacenar más datos en la tarjeta de memoria.
	Subsistema	No se tiene información histórica actualizada para el control del subsistema.
	Sistema	Peligro de imposibilidad de detección de errores en el PaN
Causas		● Se ocupó todo el espacio de almacenamiento disponible de la tarjeta de memoria.
Detección		● No se pueden grabar más datos en la tarjeta de memoria, indicando esto la misma con un mensaje de error al intentarlo.

Mitigación	● Se obtienen y resguardan los datos de la tarjeta desde el centro de control, y luego de esto se la vacía. ● Se manda a un operario a reemplazar la tarjeta por otra vacía.				
Prevención	● Se utilizan tarjetas de memoria de gran capacidad de almacenamiento. ● Al realizar la autocomprobación periódica de los componentes del monitor de barreras, incluir entre estos a la tarjeta SD e informar de su capacidad de almacenamiento utilizada y libre.				
Severidad	Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000007/SFMEA-000003
Modo de fallo		La tarjeta SD no cuenta con velocidad de lectura/escritura suficiente.
Efecto	**Local**	Se pierde la sincronía de los datos en la tarjeta de memoria.
	Subsistema	No se tiene información histórica actualizada para el control del subsistema.
	Sistema	Peligro de imposibilidad de detección de errores en el PaN
Causas		● La tarjeta de memoria se averió físicamente, ralentizándose. ● El monitor de barreras intenta grabar información a intervalos de tiempo inferiores a los especificados.
Detección		● Llega una señal al centro de control indicando la avería de la tarjeta. ● No se registran eventos y operaciones en la tarjeta. Esta última forma de detección es complicada y dificultosa.
Mitigación		● Se manda a un operario a reemplazar la tarjeta de memoria. ● Se reconfigura el monitor de barreras desde el centro de control, corrigiendo los tiempos de las operaciones.
Prevención		● Se utilizan tarjetas de gran velocidad de acceso de lectura/escritura. ● Se realizan análisis periódicos de las configuraciones y código del monitor de barreras en cuanto a la temporalidad y secuencialidad de sus operaciones.

Severidad	Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000007/SFMEA-000004
Modo de fallo		La BD local no se encuentra correctamente instalada.
Efecto	**Local**	No se puede tener acceso a los datos de manera local.

	Subsistema	No se tiene acceso a información para el control del subsistema.				
	Sistema	Peligro de errores en el PaN				
Causas		• Se produjo un error en la BD local por un problema de configuración.				
Detección		• La BD emite un código de error al producirse algún error de configuración, enviándolo al centro de control.				
Mitigación		• Se reconfigura la BD desde el centro de control, mediante un gestor de bases de datos a través una conexión remota.				
Prevención		• Realizar correctamente la configuración inicial de la BD, siguiendo un plan de implementación, y otro de configuración. • En la autocomprobación de los componentes del monitor de barreras, incluir información del estado de la BD.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

RQ-ID/SFMEA-ID		RQ-000007/SFMEA-000005				
Modo de fallo		El modelo de datos de la BD local no se encuentra correctamente definido.				
Efecto	**Local**	No se pueden grabar los datos necesarios de manera correcta o no se cumplen las restricciones.				
	Subsistema	No se tiene acceso a información necesaria para el control del subsistema.				
	Sistema	Peligro de errores en el PaN y falta de actualización				
Causas		• Se produjo un error en el análisis y diseño de la BD local.				
Detección		• Los datos no se graban con el formato correcto o no se cumplen las restricciones.				
Mitigación		• Se realiza un análisis del modelo de datos, diseño e implementación de la solución, documentando todos los procesos.				
Prevención		• Se realizan los análisis correspondientes a los nuevos requerimientos software de tipo adaptativo y correctivo, documentando todo y comunicando con los responsables.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

3.10.8 RQ-000008

RQ-ID/SFMEA-ID		RQ-000008/SFMEA-000001				
Modo de fallo		Se envía la señal de "aviso de mantenimiento", pero El sistema no entra en estado EN CONFIGURACIÓN y no DESACTIVA LAS ALARMAS.				
Efecto	**Local**	No se puede entrar al modo de configuración de los componentes de la barrera				
	Subsistema	El sistema de monitor de barreras no puede ser configurado				
	Sistema	Peligro de falta de configuración y mantenimiento en el PaN				
Causas		● Por un problema algorítmico el sistema no cambia correctamente de estado ● Por un problema del subsistema de comunicación no se detecta la señal de "aviso de mantenimiento" ● Por un problema del subsistema de comunicación no se desactivan las alarmas				
Detección		● Se puede ingresar al modo de configuración, pero el sistema sigue estando en estado ACTIVO. ● Luego de mandar la señal de "aviso de mantenimiento", el sistema no cambia al estado EN CONFIGURACIÓN ni permite acceder a dicha funcionalidad. ● Estando en estado EN CONFIGURACIÓN, el sistema sigue emitiendo alarmas.				
Mitigación		● Si se ingresa al modo de configuración, pero el sistema sigue estando en estado ACTIVO, forzar la transición de estados desde el centro de control. ● Si no se puede acceder al modo de configuración al enviar la señal de "aviso de mantenimiento", reenviar la señal. ● En caso de seguir emitiendo alarmar el sistema, estando en estado EN CONFIGURACIÓN, desactivarlas manualmente o por software, forzando dicha desactivación.				
Prevención		Se realizan auto comprobaciones periódicas del monitor de barreras, en intervalos de tiempo predeterminados, enviando los resultados al centro de control para su análisis automatizado.				
Severidad		Crítico	**Frecuencia**	Remota	**Riesgo**	S-Aceptable

3.11 Especificación semi formal/formal

3.11.1 RQ-000001

Diagrama de clases:

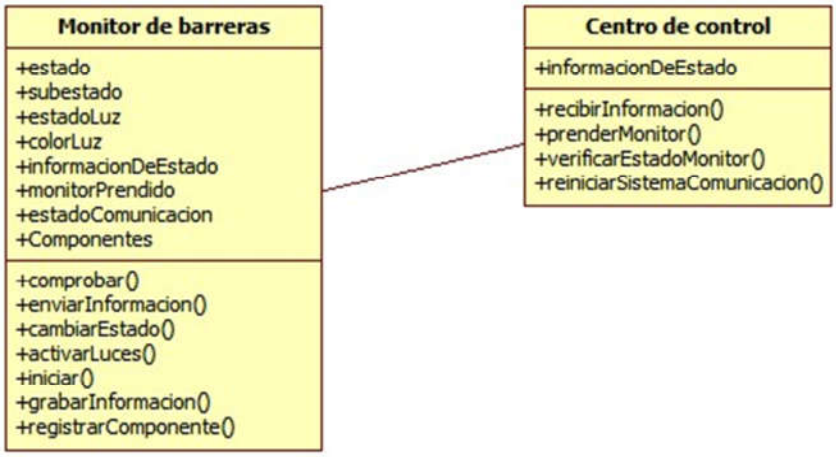
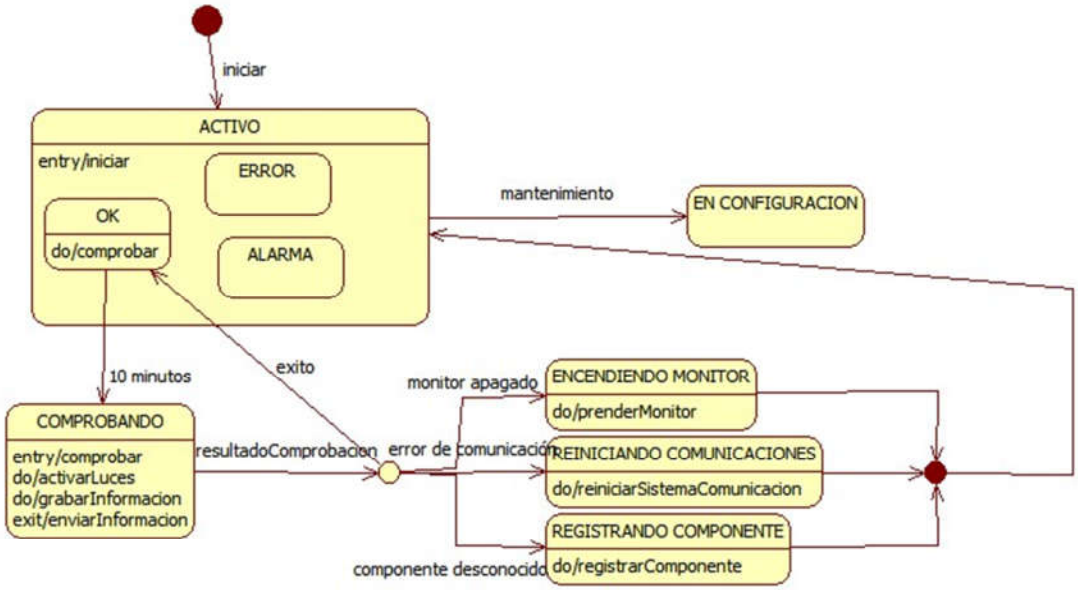


Diagrama de transición de estados:



Especificación formal en lenguaje ACSL:

```
/*@    requires estado == '';
 *  ensures estado == 'ACTIVO' && subestado == 'OK' && estadoLuz == 'on' && colorLuz
= 'verde';
*/
/*iniciar()*/

/*@    requires estado == 'ACTIVO' && subestado == 'OK';
   ensures estado == 'ACTIVO';
   behavior exito:
         assumes resultadoComprobacion = 'exito';
         ensures subestado == 'OK';
   behavior falla:
         assumes     resultadoComprobacion = 'monitor apagado' ||
                              resultadoComprobacion = 'error de comunicacion' ||
                              resultadoComprobacion = 'componente desconocido';
         ensures subestado == 'ERROR';
   complete behaviors exito, falla;
   disjoint behaviors exito, falla;
*/
/*comprobar()*/

/*@    requires estado == 'ACTIVO';
   ensures estadoLuz = 'on';
   behavior OK:
         assumes subestado == 'OK';
         ensures colorLuz == 'verde';
   behavior ERROR:
         assumes subestado == 'ERROR';
         ensures colorLuz == 'rojo';
   complete behavior OK, ERROR;
   disjoint behavior OK, ERROR;
```

```
*/
/*activarLuces()*/

/*@ requires estado == 'ACTIVO';
   ensures informacionDeEstado != '';
*/
/*grabarInformacion()*/

/*@ requires informacionDeEstado != '';
   ensures informacionDeEstado == '';
*/
/*enviarInformacion()*/

/*@ requires informacionDeEstado.monitorPrendido == 'no' && subestado != 'ERROR';
   ensures monitorPrendido == 'si';
*/
/*prenderMonitor()*/

/*@ requires informacionDeEstado.estadoComunicacion == 'erroneo' && subestado ==
'ERROR';
   ensures estadoComunicacion == 'correcto';
*/
/*reiniciarSistemaComunicacion()*/

/*@ requires informacionDeEstado.componentesRegistrados == 'no' && subestado ==
'ERROR';
   ensures sizeof(\old(Componentes)) < sizeof(Componentes);
*/
/*registrarComponente()*/
```

3.12 Glosario de términos

Término	Definición
TBD	To Be Defined, a definir
SSTA	Software Safety Tree Analysis, Análisis del Árbol de Seguridad del Software
SFTA	Software Failure Tree Analysis, Análisis del Árbol de Fallas del Software
SFMEA	Software Failure Mode and Effect Analysis, Análisis de los Modos y Efectos de Fallo del Software
HW/SW	Hardware/Software
Timestamp	Marca de tiempo
Rollback	Volver un objeto a un estado anterior al actual
JSON	JavaScript Object Notation, Notación de Objetos de JavaScript
BD	Base de Datos
SQLite	Sistema de gestión de BD relacional
ACSL	ANSI/ISO C Specification Language