Revisión sistemática de la literatura sobre implementación de arquitecturas software para sistemas críticos

Joaquín Acevedo[1], Andrea Lezcano, Emanuel Irrazábal[1]

[1] Grupo de Investigación en Innovación de Software y Sistemas Computacionales, FaCENA – UNNE, Corrientes, Argentina
{jacevedo, alezcano, eirrazabal}@exa.unne.edu.ar

Abstract. Contexto: los sistemas críticos presentan funcionalidades específicas y un conjunto de buenas prácticas normativas que buscan asegurar niveles de seguridad mínimos en cada etapa en su ciclo de vida. Esto define características en el software presente en estos sistemas que requieren una integración conjunta con el hardware, particularidad presente en los sistemas embebidos. Por ello, es posible encontrar técnicas constructivas que pueden cumplir los niveles de seguridad requeridos, pero utilizando diferentes estrategias y recursos. Objetivo: realizar un estudio secundario amplio y sistematizado sobre las arquitecturas software aplicadas en el dominio de los sistemas críticos, el nivel de seguridad alcanzado y las herramientas utilizadas para lograrlo. Método: se utilizó una revisión sistemática de la literatura para identificar estudios publicados desde enero de 1999 a diciembre de 2019 sobre arquitecturas software para sistemas críticos. Resultados: se lograron identificar los tipos de arquitectura más utilizados de acuerdo al nivel de seguridad pretendido. Asimismo, se encontró evidencia de estudio en diferentes dominios de aplicación, con especial hincapié en las normativas automotrices e industriales.

Keywords: ISO 61508, Arquitectura, Software, Estudio Secundario.

1 Introducción

Cada vez es más común trabajar con sistemas de propósitos dedicados, especialmente en aplicaciones como la de los procesos industriales, la automotriz, o la aviónica. En particular, ciertas aplicaciones son usadas en entornos críticos de tal manera que los fallos podrían provocar pérdidas financieras o incluso pérdida de vidas humanas [1]. Como respuesta a esto existen marcos regulatorios que estipulan la necesidad de demostrar la seguridad del sistema construído. Respecto del software embebido los estándares principales en sistemas críticos provienen de la normativa IEC 61508 – parte 3. Además, la norma IEC 61508 detalla el concepto de nivel de integridad de seguridad o SIL por sus siglas en inglés. El SIL ofrece una escala contra la cual medir y cuantificar el nivel de seguridad de un sistema, desde el SIL 1, el nivel más bajo posible hasta el valor SIL 4.

A partir de la estandarización de las medidas de seguridad funcional en el ámbito de los sistemas críticos con la publicación de la norma IEC 61508 [1] en 1998, el tratamiento apropiado de las características del software encargado de controlar procesos industriales es una problemática que fue creciendo con la modernización y diversificación de las aplicaciones industriales. Así, por ejemplo, la industria automotriz y su normativa ISO 26262 [2] establece los niveles de integridad de seguridad automotriz (llamado ASIL por sus siglas en inglés) o la industria aviónica y su normativa DO-178B [3] que establece los niveles de seguridad de diseño (o DAL por sus siglas en inglés).

En particular, en el apartado 7.4.3 de la norma IEC 61508-3 se especifican las buenas prácticas al construir la arquitectura del software en términos de actividades, documentación, especificación integral de cada módulo de la arquitectura y uso de buenas prácticas de programación. Desde el punto de vista de esta norma la arquitectura software consiste la definición de los subsistemas o módulos junto con sus interconexiones y, especialmente, la manera en la cual el nivel SIL es logrado. También se define el comportamiento general del software, sus interfaces y las decisiones que sostendrán las técnicas detalladas de diseño de componentes.

En este sentido, las elecciones de los tipos de arquitectura software que cumplen con las buenas prácticas antes mencionadas pueden ser complejas. Actualmente existen diferentes estudios que presentan arquitecturas software para sistemas críticos, pero no emergen arquitecturas estándar por nivel de seguridad integral y orientadas a cubrir la mayor cantidad de dominios de problemas.

Por todo ello, en este artículo se ha llevado adelante una Revisión Sistemática de la Literatura (RSL) para identificar las arquitecturas software habituales en el desarrollo de software para sistemas críticos que han demostrado ser validadas para determinados niveles de seguridad. Asimismo, se han analizado los dominios de aplicación y las tecnologías relacionadas.

El trabajo se encuentra organizado de la siguiente manera además de esta introducción. La sección 2 describe los estudios secundarios relacionados con la temática. En la sección 3 se detalla la metodología empleada para el estudio correspondiente a la fase de planificación de una RSL y se presentan las preguntas de investigación en la tabla 1. En la sección 4 se reportan las actividades correspondientes a la etapa de conducción de la RSL. En la sección 5 se proveen los resultados obtenidos; para su presentación se sintetizaron los resultados de 23 estudios y se responden las preguntas de investigación. Finalmente, en la sección 6 se incluye la conclusión de la RSL.

2 Trabajos Relacionados

Existen trabajos relacionados con el objetivo de esta revisión y que proveen un estado del arte acerca de diferentes cuestiones relacionadas a las arquitecturas para sistemas críticos. En [4] los autores presentan un mapeo sistemático de la literatura enfocado en las pruebas basado en modelos para seguridad del software, en este estudio se incluye un análisis sobre diferentes publicaciones que presentan un desarrollo software sobre un dominio determinado de la seguridad en software. Sin embargo, no se incluyen preguntas relacionadas con los niveles de seguridad. En [5] se presenta el estado del arte de la combinación de técnicas de ingeniería conducida por modelos e ingeniería de línea de productos para el desarrollo de arquitecturas de software para sistemas críticos. En este caso las preguntas están enfocadas a identificar estudios que traten sistemas embebidos desde la dimensión de la ingeniería dirigida por modelos. Esta aproximación compromete la validez interna de las técnicas evidenciadas y no se tratan en profundidad los requerimientos que aseguran la criticidad de los sistemas resultantes de las técnicas registradas en el estudio secundario.

Asimismo, en. [6] se presenta una caracterización de diferentes técnicas empleadas para representar arquitecturas software para sistemas embebidos. Se discute sobre los riesgos asociados de cumplir para arquitecturas de sistemas críticos sin incluir un análisis directo del nivel de seguridad. Por todo ello, esta revisión difiere de las revisiones mencionadas de la siguiente manera:

- Se identifican las normas y estándares relacionados a los sistemas críticos.
- Se incluyen estudios publicados desde 1999 a 2019.
- Se caracteriza la relación entre las arquitecturas software y el nivel de seguridad que presentan al considerar su aplicación en un sistema crítico.
- Se describe la tecnología empleada para el desarrollo de arquitecturas.
- Se provee un análisis de cada estudio incluido en términos de rigor, validez y aplicabilidad.

3 Metodología

En esta sección se describe el método empleado para la realización del estudio secundario. En este caso se ha seguido el enfoque identificado por las pautas para realizar Revisiones Sistemáticas de Literatura en Ingeniería de Software [7]. De acuerdo con lo descripto en la introducción de este trabajo, el objetivo de la presente RSL es identificar las arquitecturas software empleadas para aplicaciones críticas de sistemas embebidos, la evidencia del cumplimiento de las normativas internacionales y las tecnologías utilizadas para ello. El análisis se basa en preguntas de investigación que se encuentran en la tabla 1.

Tabla 1. Preguntas de investigación.

Preguntas de investigación	Motivación
RQ 1. ¿Qué arquitecturas software existen para software crítico?	Identificar las diferentes arquitecturas software en el campo de aplicación de los sistemas críticos y la frecuencia de su uso y reporte en estudios académicos.
RQ 2. ¿Qué normativa cumplen?	Identificar las normativas que se presentaron como requerimientos a cumplir por las arquitecturas presentadas y el grado de cumplimiento de la misma.
RQ 3. ¿Qué nivel de seguridad verifican?	Determinar la evaluación realizada en cuanto al cumplimiento del nivel de seguridad que la arquitectura software presenta.
RQ 4.1 ¿Qué tecnologías utilizan?	Determinar las tecnologías empleadas para el desarrollo de las arquitecturas software y su capacidad para lograr el cumplimiento de los requisitos incluidos en las normativas.
RQ 4.2 ¿Cuáles son las plataformas utilizadas?	Identificar las plataformas sobre las cuales se desarrollan arquitecturas software, si se utilizan herramienta propietarias o de fuente abierta, propias o de terceros.
RQ 5 ¿Cuáles son los dominios de aplicación?	Determinar la evaluación realizada en cuanto al cumplimiento del nivel de seguridad que la arquitectura software presenta.
RQ 6 ¿Cuáles son las plataformas utilizadas?	Determinar las tecnologías empleadas para el desarrollo de las arquitecturas software y su capacidad para lograr el cumplimiento de los requisitos incluidos en las normativas.

3.2 Estrategia de búsqueda

En esta sección se describe la estrategia de búsqueda explicando el alcance en cuanto a fuentes, el método empleado y la cadena de búsqueda.

Se decidió realizar la búsqueda en bases de datos que contengan artículos relacionados a ciencias de la computación e ingeniería. Se emplearon las bases de datos generalmente utilizadas en otras revisiones de la misma temática: Scopus, IEEE Xplore, ACM Digital Library y Springer Link. El artículo se realizó en base a la revisión de publicaciones en revistas, conferencias y artículos. En la tabla 2 se presenta cada término junto con sus palabras claves[1].

Tabla 2. Términos clave y sinónimos empleados para crear la cadena de búsqueda.

Términos	Palabras clave
safety	"safety*" OR "functional safety" OR critical OR "mission-critical" OR "fault tolerance" OR "fault tolerant" OR "fault-tolerance" OR "fault-tolerant" OR sil OR ssil OR rams
architecture	architect* OR "architectural pattern" OR "architectural design" OR design
software	software OR application OR "source code" OR firmware
61508	61508 OR 61508-3 OR 50128 OR MISRA OR "MISRA C" OR 62279 OR do-178* OR 26262

3.3 Criterio de selección de los estudios

Se incluyen los artículos referidos al desarrollo de arquitecturas software para aplicaciones en sistemas críticos, que presenten un nivel de seguridad y publicados a partir de 1999 en revistas indexadas y en conferencias. Se excluyen los artículos de descripción o uso de herramientas, duplicados y presentaciones de conferencias. En cuanto al contenido, se excluyen los estudios que describan, implementen o evalúen lenguajes de descripción de requerimientos.

Para seleccionar los estudios primarios se aplican los criterios de inclusión/exclusión leyendo los resúmenes de los artículos encontrados. En caso de que se registren dudas acerca de su pertinencia se leerá el artículo completo. El proceso de revisión será acompañado por el criterio de los investigadores para consensuar sobre la correcta aplicación de los criterios de inclusión/exclusión[2].

3.4 Estrategia para la extracción y síntesis de datos

El procedimiento empleado para la extracción de datos consiste en un primer ordenamiento de todas las publicaciones resultantes de la consulta a la base de datos de investigación en una planilla respetando los identificadores de exportación de la fuente sumado a un campo de control para indicar el estado de inclusión o exclusión del artículo. Cada base de datos tiene sus campos propios para indexar publicaciones, por consiguiente, luego de llegar al conjunto de todos los estudios a incluir en el estudio, se procedió a una normalización de todos sus identificadores.

Para realizar la síntesis de los datos se organizó cada publicación aceptada en un directorio de acuerdo a la fuente de acceso. Luego se procedió a la creación de etiquetas para identificar cada dimensión de análisis correspondiente a los criterios sobre las preguntas de investigación. El resultado de este procedimiento fue presentar la información de cada publicación que signifiquen descripciones concisas relacionadas a las preguntas de investigación.

4 Realización de la Revisión

La RSL fue realizada siguiendo todos los pasos del protocolo definido en la sección previa y completada en un año, en este periodo se incluye el tiempo requerido para cada una de las tres fases, es decir, planeamiento, realización y reporte. Inicialmente fueron encontradas 7550 publicaciones. Cada fase de búsqueda se detalla en la tabla 3 y la tabla 4 con mayor precisión.

[1] Anexo: bit.ly/rslcacic2020ASC, en la tabla A1 se describen las cadenas de búsqueda para cada base de datos.
[2] Anexo: bit.ly/rslcacic2020ASC, en la tabla A2 se describen los criterios de inclusión y exclusión.

4.1 Selección de los estudios primarios

En esta etapa de selección de los estudios primarios se encuentran los criterios de aseguramiento, aplicación de los criterios de aseguramiento y extracción de los datos de los artículos. Se encontraron en total de 7550 artículos aplicando la estrategia de búsqueda definida el protocolo. La búsqueda se efectuó empleando título, resumen y palabras claves indexadas (ver tabla 3).

Tabla 3. Detalles de la búsqueda automatizada.

Base de datos	Resultados de búsqueda
IEEE Xplore	295
SCOPUS	696
ACM Digital Library	2000
Springer Link	4559
TOTAL	7550

De los primeros 7550 resultados se realizó el filtrado empleando la lista de comprobación establecida en el protocolo, se realizaron revisiones reiteradas hasta llegar a un número final de 23 artículos con fechas de publicación que varían desde del 2004 al 2017 (ver tabla 4). Además del criterio de inclusión/exclusión, se considera también el aseguramiento de la calidad de los estudios primarios para proveer un criterio más detallado y con el sentido de evaluar la importancia de los estudios individuales cuando los resultados están siendo sintetizados. En la presente sección se trata la calidad en términos de minimizar el sesgo y maximizar la validez interna y externa como se indica en el manual Cochrane para revisiones sistemáticas de intervenciones [8] desde las practicas adoptadas por las buenas prácticas para RSL [7]. Para esta etapa se emplearon una serie de preguntas orientadas a evaluar el modo en que los estudios primarios presentan la información relacionada al objetivo, así como también el rigor de los tópicos desarrollados[3].

A continuación, se extrajeron los datos cuantitativos y cualitativos de cada uno de los 23 artículos para contar con información valida y objetiva. Se organizó cada estudio en una disposición para enfocar los contenidos de cada publicación, bajo los fines de la investigación. La importancia de esta actividad radica en la presentación apropiada del reporte de la revisión al momento de la realización de la discusión, presentada en la sección 5. Para la síntesis de los datos representativos de cada publicación se emplearon los métodos establecidos en la sección 3. La realización de esta actividad permitió caracterizar y evaluar cada publicación empleando en conjunto el estudio completo y la información extendida de la tabla de clasificación y el ordenamiento de las herramientas de apoyo. Al momento de recolectar la información relacionada a cada pregunta de investigación se registraron datos puntuales en cada campo del formulario de extracción.

Tabla 4. Detalles de cada fase de selección de estudios primarios.

Fase	Descripción	Incluidos	Excluidos
Fase 1	Resultados de búsqueda	7550	0
Fase 2	Selección por criterios de inclusión	2899	4651
Fase 3	Selección por criterios de exclusión	57	2842
Fase 4	Selección por título y abstract	26	31
Fase 5	Validación conjunta	23	3

5 Reporte de la Revisión

A continuación, se describen los resultados organizados por pregunta de investigación. En la discusión de los resultados se hace referencia a los artículos con la forma Px siendo “x” el número de artículo tal y como está descripto en la sección 2 de la documentación anexa a este trabajo[4].

[3] Anexo: bit.ly/rslcacic2020ASC, en la sección 3 se describe el análisis de calidad de los estudios.
[4] Anexo: bit.ly/rslcacic2020ASC, sección 2.

5.1 RQ1. ¿Qué arquitecturas software existen para sistemas críticos

En la tabla 5 se resumen las arquitecturas software encontradas en el análisis de los artículos y el número total encontrado. El 39% de los trabajos bajo estudio aplican una implementación de arquitecturas de capas. Esta decisión de diseño de los sistemas bajo estudio requiere pruebas exhaustivas y una clara separación de funciones que pueden introducir más posibilidades de fallo; por tanto, al seleccionar una arquitectura en capas se asegura que cada función tenga acotada su probabilidad de fallo. En P3 se evidencia un caso de seguridad modular para una red de trabajo que requiera cumplir con la normativa IEC 61508 y describe un protocolo para transferir datos de forma segura cumpliendo con el nivel 3 SIL.

El 30% de las publicaciones emplean una arquitectura con un esquema de votación del tipo *M-out-of-N* y un 70% de este subconjunto emplean específicamente el esquema 1oo2. Esta distribución entre los estudios se justifica ya que ciertos dominios requieren que el sistema se mantenga funcional bajo condiciones extremas y cumplimentando el nivel de SIL requerido. En el trabajo P20 los autores presentan una solución que implementa hardware redundante y dado el caso de implementación de la solución sobre un entorno de alta presión alta temperatura, no recomiendan el uso de su esquema de transmisores de doble presión porque podrían agregar una falla en el sensor completo.

Cerca de un 26% de los trabajos mencionan y aplican un esquema software en conjunto con redundancia de hardware como medida para mitigación de riesgos. En P13 se menciona la redundancia de hardware como un método de monitoreo seguro bajo pruebas de comparación empleando un votador.

Asimismo, el 13% de las publicaciones emplean una arquitectura maestro esclavo como solución propuesta. Así, en P9 se recomienda un microcontrolador maestro y diferentes microcontroladores esclavos, con su tratamiento respectivo de fallas posibles.

Solo un 8% de los trabajos proponen una arquitectura que contemple redundancia de software y las mismas no cubren todo el procedimiento de certificación necesario para la implantación de un sistema crítico dirigido por software.

Tabla 5. Tipos de arquitectura software utilizadas.

Identificador	Arquitectura	Artículo que implementa la solución	Total
MooN	m-out-of-n	P5 P6 P8 P9 P13 P14 P15 P20	8
MS	Maestro esclavo	P9 P18 P19	3
RHW	Redundancia hardware	P7 P13 P14 P16 P20 P21	6
AC	Arquitectura en capas	P3 P10 P11 P16 P17 P19 P20 P21 P22	9
RSW	Redundancia software	P1 P2	2
FSM	Máquina de estados finitos	P12	1
SCP	Safety channel pattern	P23	1

5.2 RQ2. ¿Qué normativa intentan cumplir?

Cerca del 78% de los estudios tratan expresamente la normativa IEC 61508 mientras que el 22% restante indican normativas específicas del dominio y del ámbito de aplicación de la arquitectura del software en un sistema critico refiriendo a las características y restricciones de diseño demandados: automotriz [P7] [P25] [P15] aviónica [P22] y normativas específicas de telecomunicación [P3].

Un resultado notable es el creciente desarrollo de las tecnologías de conducción automática tratada en las practicas recomendadas para vehículos de superficie [9] aumentando la rigurosidad requerida para los componentes software y los circuitos programables, con tiempos de respuesta acotado. Se menciona, entonces, a la normativa ISO/IEC/IEEE 42010 que regula esta nueva área sin excluir a la norma ISO 26262 existente.

El trabajo P19 no hace referencia directa la norma IEC 61158 porque el ámbito del estudio en cuestión es la implementación de un protocolo de comunicación segura. Técnicamente la norma IEC 61784 define protocolos basados en la norma mencionada anteriormente e incluye otras definiciones extendidas.

En P21 se presentan resultados experimentales sobre la auditoria del sistema bajo una prueba de inyección de fallas para verificar la implementación de los conceptos de seguridad, además se indica la posibilidad de realizar pruebas automatizadas para validación del sistema en conformidad con el estándar ISO7637[10] . En la tabla 6 se resumen las normativas que se intentan cumplir y los artículos que la mencionan.

Tabla 6. Tipos de arquitectura software utilizadas.

Normativa	Artículo
IEC 61508	P1 P2 P3 P4 P5 P6 P7 P9 P10 P11 P13 P14 P15 P16 P18 P19 P20 P21
DO 178	P12
EN 14908	P5
EN 50126	P13
EN 50128	P13
EN 50129	P13
ISO 26262	P8 P9 P11 P17 P21 P22 P23
IEC 61131	P7
EN 954-1	P6
IEC 61800	P6
IEC 60204	P6
IEC 61511	P16
API RP 14C	P16
IEC 61784	P19 P22
IEC 61158	P22
ISO/IEC/IEEE 42010	P23
ISO/IEC 25010	P23

5.3 RQ3. ¿Qué nivel de seguridad verifican?

Un 60% de los trabajos incluidos en el estudio tienen como objetivo alcanzar el nivel SIL 3 o similar de acuerdo con la normativa específica. La distribución de publicaciones referidas a arquitecturas software para sistemas críticos que abordan este nivel de seguridad integral permite visualizar la presencia de soluciones software en este ámbito que presenta conformidad con las normas vigentes en cada dominio tratado.

Alrededor de un 39% de los trabajos revisados tienen como objetivo certificar hasta un nivel SIL4 o equivalente en ASIL y DAL. Finalmente, el 30% de las publicaciones tienen como objetivo alcanzar un nivel SIL2 o inferior, es importante aclarar que el nivel de seguridad requerido varía de acuerdo al dominio del sistema crítico y que la presencia de publicaciones que mencionan estos niveles de SIL1 y SIL2.

Tabla 7. Nivel SIL identificado en cada artículo.

SSIL	Artículo
SIL1 / ASIL A	P2 P15
SIL2	P2 P6 P10 P11 P16 P20
SIL3 / ASIL B / DAL B	P1 P2 P3 P4 P5 P6 P7 P9 P12 P14 P16 P18 P19 P22
SIL4 / ASIL D /DAL A	P1 P2 P4 P8 P12 P13 P17 P21 P23

5.4 RQ4.1 ¿Qué tecnologías utilizan?

El apartado de tecnología elegida como plataforma de implementación es la dimensión de análisis que presenta más diversidad de resultados. El total de las publicaciones emplea un esquema software de alto nivel propio para disminuir la complejidad de los esquemas de hardware específicos. El 30% de los estudios incluidos emplea un *framework* como tecnología para el desarrollo de software. El 48% de los estudios incluidos presenta un desarrollo de software empleando librerías. En menor proporción, cerca del 22% de los estudios incluidos solo describe el desarrollo de software a nivel de código fuente.

En algunos casos del dominio automotriz se describe el uso de un bus CAN [P8] [P11] [P17] [P23] que a nivel de tecnología software emplean librerías o un *framework* para una mayor integración de funciones. La implementación del bus es específica para cada caso tratado. En la tabla 8 se indican los artículos por tecnología.

Tabla 8. Tecnologías empleadas para arquitecturas software.

Tecnologías	Artículo	Total
Framework	P3 P10 P13 P14 P19 P21 P22	7
Librerías	P2 P4 P5 P6 P7 P8 P11 P15 P17 P20 P23	11
Código fuente	P1 P9 P12 P16 P18	5

5.5 RQ4.2 ¿Cuáles son las plataformas utilizadas?

La distribución de plataformas evidenciadas para implementar una arquitectura software se visualiza en la tabla 9. Cerca del 65% de los estudios emplea una solución propia, el 21% de los estudios describe el uso de un microcontrolador disponible en el mercado. Un resultado notable es el descripto en el artículo [P18] en el que se emplea una plataforma que cuenta con certificación IEC61508 SIL 3 e ISO26262 ASIL D. En menor proporción se encuentra el uso de arquitecturas software sobre controladores lógicos programables (PLC) significando un 13% del total de estudios incluidos. En los casos que contemplan el uso de controladores lógicos programables, su implementación fue acompañada por un entorno operativo empleando un sistema operativo de tiempo real para validar el comportamiento esperado del sistema bajo prueba. Y por último se encuentra la elección de un sistema operativo de tiempo real (RTOS) como plataforma, significando cerca del 8% de los estudios incluidos. En P6 se menciona el uso de un sistema operativo en tiempo real para su implementación en los sistemas de control.

Tabla 9. Plataformas empleadas en cada tipo de arquitectura software desarrollada.

Plataforma	Artículo	Total
Solución específica	P1 P2 P3 P4 P5 P8 P10 P11 P14 P15 P16 P17 P19 P22 P23	15
MCU	P9 P12 P13 P18 P21	5
PLC	P6 P7 P20	3
RTOS	P6 P7	2

5.6 RQ5. ¿Cuáles son los dominios de aplicación?

Para responder este apartado se toma en consideración el dominio al que pertenece el sistema crítico trabajado en la publicación. Un 43% de las publicaciones revisadas emplean una arquitectura software para sistema crítico bajo el dominio de la seguridad en equipamiento industrial.

Alrededor del 26% de los estudios describen sistemas que se emplean bajo el dominio de la industria automotriz, incluyendo las actividades de validación, análisis y desarrollo de la arquitectura software. Cerca del 8% de los estudios revisados presentan un sistema que se emplea bajo el dominio de trenes [P23] [P9]. En P20 se trata la implementación de sistemas críticos en el dominio de seguridad en instalaciones *offshore*. Los dominios de aplicación para sistemas críticos que incluyen una arquitectura software (ver tabla 10).

Tabla 10. Dominio del sistema critico tratado en cada artículo.

Dominio	Artículo	Total
Seguridad en equipos industriales	P1 P2 P3 P4 P5 P6 P7 P14 P15 P16	10
Automotriz	P8 P9 P10 P11 P17 P21 P23	7
Aviónica	P12	1
Redes de comunicación segura	P19 P22	2
Ferroviario	P13 P18	2
Seguridad en instalaciones *offshore*	P20	1

5.7 RQ6 ¿Cuáles son las actividades reportadas?

Se requieren actividades que describan diseño de la arquitectura incluyendo casos de seguridad, y actividades que aseguren la aplicación del sistema empleando componentes electrónicos certificados y los roles de prueba respectivos siguiendo los casos de seguridad desarrollados de forma particular. Las actividades cubiertas en cada publicación difieren porque cada nivel de integridad de seguridad tiene mayor rigurosidad en los requerimientos. El 50% de este conjunto de estudios mencionados describe la etapa de pruebas de sistema. Una de las características notables en este dominio es la evaluación integral de todas las restricciones del entorno en el cual se va a implementar el sistema crítico, este hecho requiere una correcta documentación del software y del hardware que se va a emplear.

Finalmente, P23 y P9 incluyen actividades de validación a nivel de pruebas de sistema, esto se relaciona con el nivel de SIL que se pretende lograr en estos dos artículos. La identificación por actividad y artículo se encuentra en la tabla 11.

Tabla 11. **Actividades reportadas en el desarrollo de arquitecturas software para sistemas críticos.**

Actividad	Artículo	Total
Análisis	Todos los artículos	23
Investigación	P4 P5 P7 P9 P10 P11 P17	7
Desarrollo	P1 P2 P3 P5 P6 P8 P9 P10 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P23	19
Validación	P1 P2 P3 P4 P6 P7 P8 P9 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P22 P23	22
Implantación	P13 P16	2

5.8 Resultados adicionales

Como se muestra en la Fig. 1, la distribución de publicaciones que cumplen con los criterios de investigación propuesto presenta una fluctuación que alcanza sus máximos entre los años 2009 a 2012.

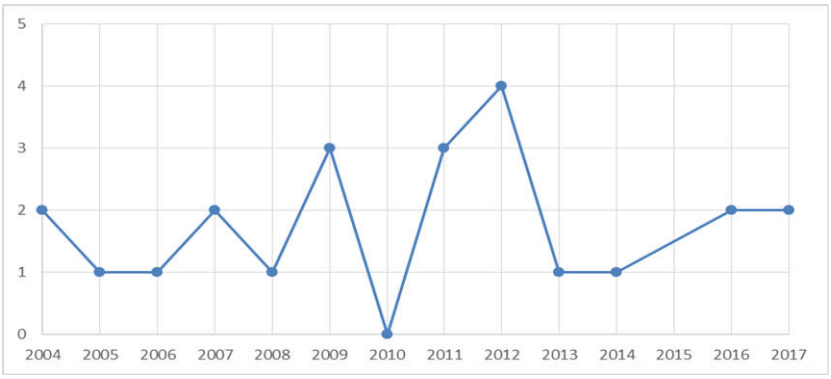

Fig. 1. Distribución de los estudios primarios por dominio.

Una vez extraídos los datos relevantes del conjunto de estudios seleccionados para conducir la revisión se llegó a un esquema que lo describe en función de los parámetros que se relacionan de forma directa con las preguntas de investigación, significando información con granularidad basada en la unicidad de cada publicación y en las características excluyentes y no excluyentes indicadas en la metodología. Este cambio en los esquemas presentados respetando el criterio de unicidad de las publicaciones y presentando vinculaciones ha sido estudiado formalmente en trabajos como Hashemi et al. [11]. Implementados sobre el conjunto de estudios se observa un agrupamiento importante de publicaciones que tratan una arquitectura en capas con SIL 2 y SIL 3. Y otra agrupación presentando una arquitectura tomando un esquema base de M-out-of-N agrupada en los niveles SIL 3 y SIL 4, esto es presentado en la Fig. 2. En cuanto a la proporción en la forma de presentar desarrollos de software relacionado a las arquitecturas incluidas teniendo presente la dimensión temporal, no se presentaron agrupaciones mayores a dos publicaciones y su distribución se visualiza en la Fig. 3.

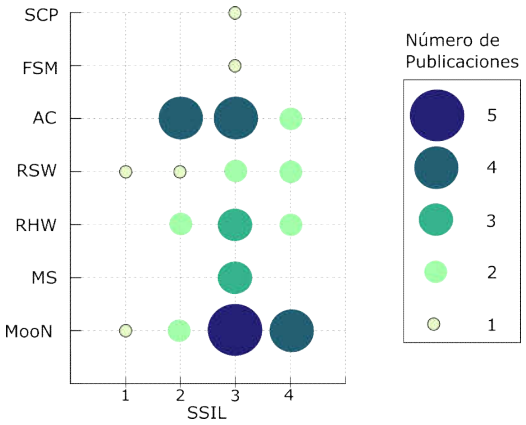

Fig. 2. Agrupación de estudios primarios por arquitectura y por SIL tratado.

5.9 Amenazas a la validez

A continuación, se indican las amenazas a la validez identificadas durante el desarrollo del trabajo. En cuanto a la búsqueda realizada: una revisión sistemática de la literatura empleando diversos motores de búsqueda presenta la necesidad de controlar los estudios que se encuentren indexados en varias fuentes y la selección de la versión de última revisión si esto ocurre. Este aspecto fue tratado al emplear un control por índice luego del resultado de la cadena de búsqueda al finalizar la fase 1 y hasta la fase 4 inclusive cuando se realizó la selección por título y resumen.

En cuanto a los estudios incluidos: se ha buscado asegurar la validez de los datos presentados en la presente revisión a partir de la calidad del conjunto de artículos incluidos. Según el enfoque presentado en las buenas prácticas para realizar revisiones los estudios de observación suelen ser más susceptibles al sesgo que los estudios experimentales; las conclusiones que se pueden extraer de ellos son necesariamente más tentativas y a menudo generan hipótesis, destacando áreas para futuras investigaciones [12]. Este enfoque fue adoptado para el criterio de aseguramiento de calidad por Kitchenham y Charters para minimizar el sesgo y maximizar la validez interna y externa. Empleando estas herramientas de aseguramiento de calidad para cada estudio se logró obtener una medida de calidad del conjunto de estudios para tratar este aspecto de validez.

Fig. 3. Agrupación de estudios primarios por arquitectura y por año de publicación.

5 Conclusiones

En este trabajo se presentó una RSL enfocada en las arquitecturas software y su aplicación en sistemas críticos. Por lo tanto, se definieron los métodos de búsqueda y selección de los estudios primarios, con el propósito de obtener el estado del arte de los enfoques. Se definieron los criterios de investigación, se presentó el método utilizado para realizar la búsqueda, los criterios de selección de los estudios primarios y se realizó el reporte.

Los resultados indican que hay un amplio desarrollo de arquitecturas software en varios dominios industriales, se evidencia una mayoría dominio automotriz y el dominio de seguridad en equipamiento industrial. Se llegó a una distribución de estudio por año, plataforma, tecnología y arquitectura en el diseño. Se buscó además determinar el nivel de seguridad integral tratado en cada arquitectura, en este sentido, se presenta una distribución importante sobre niveles SIL 3 y SIL 4 en el conjunto de estudios analizados. También se encontraron referencias a normas que regulan cada proceso implementado como función de seguridad delegado al software.

Del presente estudio, podemos concluir que la implementación de arquitecturas software en sistemas críticos requiere una evaluación integral que incluye las tecnologías a emplear, las normas existentes que brindan un marco de trabajo sobre el cual debe adecuarse el ciclo de desarrollo, el nivel de seguridad que deben cumplimentar las funciones implementadas por el software, la tecnología y la plataforma de desarrollo elegidas que permitan acotar el índice de fallos, dentro de lo permitido por el nivel de seguridad requerido.

En términos generales, se evidenciaron arquitecturas software que pueden cumplir altos niveles de SIL, en este aspecto se encontraron evidencias de que predominan las arquitecturas software con un diseño en capas. En cuanto a la normativa que intentan cumplir los desarrollos presentes en los estudios, Se evidencio que la utilización del estándar IEC 61508 se presenta en la mayoría de los estudios, pero su conformidad debe presentarse en conjunto con normas específicas del dominio.

Sobre los apartados de tecnología empleadas y plataformas de desarrollo, se identificó una tendencia por emplear soluciones específicas, si bien existen soluciones de hardware certificado que permite un mayor control

durante todo el ciclo de vida del proyecto, gran parte de los estudios emplean definiciones de hardware propias y definiciones de software a nivel de código fuente o librerías que hacen uso del soporte físico establecido, con *frameworks* destinados a la validación de las funciones implementadas por software.

En cuanto al apartado de actividades reportadas, no hay una definición única a seguir para el desarrollo de las arquitecturas, pero existen procesos claros y normalizados para establecer módulos, probar cada elemento software y controlar por simulación su interacción con el hardware, que, en muchos casos evidenciados, significa presentar el nivel de seguridad requerido.

Para resumir, los resultados de la RSL presentados en las secciones anteriores nos permiten afirmar que existe una fuerte tendencia de implementar funciones seguras mediante software, esta tendencia es creciente, y abarca varios dominios de la industria.

Agradecimientos. Este desarrollo se realizó a partir del proyecto de investigación PI 17F017 y 17F018 de la Secretaría General de Ciencia y Técnica de la Universidad Nacional del Nordeste.

Referencias

1. International Electro-technical Commission IEC, “IEC 61508: Functional safety of electrical/electronic/ programmable electronic safety-related systems.” 1998.
2. International Organization for Standardization (ISO), “IS026262 Road vehicles – Functional safety.” 2011.
3. Radio Technical Commission for Aeronautics (RTCA), “DO-178B, Software Considerations in Airborne Systems and Equipment Certification.” 1992.
4. H. G. Gurbuz and B. Tekinerdogan, “Model-based testing for software safety: a systematic mapping study,” Softw. Qual. J., vol. 26, no. 4, pp. 1327–1372, 2018, doi: 10.1007/s11219-017-9386-2.
5. P. G. G. Queiroz and R. T. V Braga, “Development of Critical Embedded Systems Using Model-Driven and Product Lines Techniques: A Systematic Review,” in 2014 Eighth Brazilian Symposium on Software Components, Architectures and Reuse, 2014, pp. 74–83.
6. E. A. Antonio, F. C. Ferrari, and S. C. P. F. Fabbri, “A Systematic Mapping of Architectures for Embedded Software,” in 2012 Second Brazilian Conference on Critical Embedded Systems, 2012, pp. 18–23, doi: 10.1109/CBSEC.2012.22.
7. B. A. Kitchenham and S. M. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering | Request PDF[1] B. A. Kitchenham and S. M. Charters, ‘Guidelines for performing Systematic Literature Reviews in Software Engineering | Request PDF.’ https://www.researchgate.” https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering (accessed May 23, 2020).
8. The Cochrane Collaboration, Cochrane Handbook for Systematic Reviews of Interventions. 2019.
9. SAE, “Surface vehicle recommended practice—Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles,” 2018.
10. ISO/DIS 7637-2, “SO7637-2 Road vehicles - Electrical disturbances from conduction and coupling.” International Organization for Standardization, 2011.
11. R. R. Hashemi, S. De Agostino, B. Westgeest, and J. R. Talburt, “Data granulation and formal concept analysis,” in Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS, 2004, vol. 1, pp. 79–83, doi: 10.1109/nafips.2004.1336253.
12. Centre for Reviews and Dissemination, Systematic Reviews: CRD"s guidance for undertaking reviews in health care. York Publishing Services Ltd, 2009.