



# Cadenas de procesamiento re-configurables en SoC/MPSoC

Autor:

Ing. Carlos Jorge Maffrand

Director:

Esp. Ing. Gonzalo Lavigna (FIUBA)

*Esta planificación fue realizada en el curso de gestión de proyectos  
entre el 24 de junio de 2021 y el 24 de agosto de 2021.*

## Índice

1. Descripción técnica-conceptual del proyecto a realizar . . . . .	5
2. Identificación y análisis de los interesados . . . . .	7
3. Propósito del proyecto . . . . .	7
4. Alcance del proyecto . . . . .	7
5. Supuestos del proyecto. . . . .	8
6. Requerimientos . . . . .	9
7. Historias de usuarios ( <i>Product backlog</i> ). . . . .	12
8. Entregables principales del proyecto . . . . .	13
9. Desglose del trabajo en tareas . . . . .	14
10. Diagrama de Activity On Node. . . . .	16
11. Diagrama de Gantt . . . . .	17
12. Presupuesto detallado del proyecto . . . . .	20
13. Gestión de riesgos . . . . .	21
14. Gestión de la calidad . . . . .	24
15. Procesos de cierre . . . . .	26

## Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	24 de junio de 2021
1	Se genera la primera versión de los incisos 1, 2, 3, 4 y 5.	04 de julio de 2021
2	Se genera la primera versión de los incisos 6, 7, 8 y 9.	08 de julio de 2021
3	Se genera la primera versión de los incisos 10, 11 y 12.	17 de julio de 2021
4	Se genera la primera versión de los incisos 13, 14 y 15.	31 de julio de 2021

## Acta de constitución del proyecto

Buenos Aires, 24 de junio de 2021

Por medio de la presente se acuerda con el Ing. Carlos Jorge Maffrand que su Trabajo Final de la Especialización en Sistemas Embebidos se titulará “Cadenas de procesamiento re-configurables en SoC/MPSoC ”, consistirá en implementar un sistema que permita seleccionar bloques de procesamiento almacenado en memoria, realizar re-configuración dinámica en lógica programable y ofrecer un entorno de pruebas para el filtro seleccionado. Este entorno de pruebas debe ser capaz de generar una onda de excitación para el filtro, adquirir los datos filtrados y generar reportes de pruebas. Este desarrollo tiene como finalidad evaluar el uso de bancos de filtros mediante re-configuración dinámica de lógica programable para aplicaciones satelitales y de radar, y tendrá un presupuesto preliminar estimado de 800 horas de trabajo y \$ 4.429.000, con fecha de inicio 24 de junio de 2021 y fecha de presentación pública 24 de junio de 2022.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg  
Director posgrado FIUBA

Mg. Ing. Franco Alcaraz  
INVAP

Esp. Ing. Gonzalo Lavigna  
Director del Trabajo Final

## 1. Descripción técnica-conceptual del proyecto a realizar

En los desarrollos tecnológicos, tanto aeroespaciales como de radares, es muy frecuente la utilización de procesamiento digital de señales en dispositivos de lógica programable. En consecuencia, dado los elevados costos de los lanzamientos, misiones espaciales y desarrollos de defensa, es necesaria la utilización eficiente de los recursos. Por lo que, mediante la re-configuración parcial, se podría reducir el número de dispositivos de lógica programable utilizados en futuros desarrollos.

El sistema propuesto en este plan de trabajo permitirá hacer una evaluación del uso de técnicas de re-configuración dinámica parcial en dispositivos de lógica programable para aplicaciones espaciales y de defensa.

Las Cadenas de procesamiento re-configurables en SoC/MPSoC permiten seleccionar filtros de procesamiento, previamente generados, de un banco de filtros almacenado en memoria, realizar la re-configuración dinámica y evaluar el funcionamiento del banco seleccionado. Como componente principal, el dispositivo a desarrollar cuenta con un System on Chip (SoC) o Multi Processor System on Chip (MPSoC).

En la Figura 1 se muestra un diagrama de bloques en el que se observan las funcionalidades principales de la aplicación a desarrollar. De la misma, se desprende que el objetivo de la aplicación es generar un entorno de software-HDL que permita la selección de bancos de filtros que re-configuran la lógica programable del SoC/MPSoC.

El entorno de pruebas se debe implementar haciendo uso de un kit de desarrollo que contenga un SoC o MPSoC de serie 7 o superior de la marca Xilinx. Por lo que, se deben establecer los mecanismos de re-configuración mediante el entorno de desarrollo Vivado.

El dispositivo debe constar de los principales bloques constitutivos:

- Interfaz de usuario (UI): debe proveer al operador un modo de configurar el filtro de procesamiento empleado en el dispositivo, disparar la ejecución de pruebas y obtener los archivos de reporte. Adicionalmente, el operador debe poder actualizar los archivos que conforman el banco de filtros con los que se puede configurar la lógica programable (PL).
- Sistema de procesamiento (PS): debe contar con un sistema operativo que permita la ejecución de diferentes aplicaciones. Las mismas tienen como funcionalidades principales: controlar los periféricos del procesador, realizar la re-configuración de la PL, generar y administrar los reportes de pruebas, administrar el banco de filtros, configurar las pruebas a realizar y gestionar la comunicación con la PL mediante el bus AXI.
- Lógica programable (PL): en la misma se debe disponer de una parte estática donde se instancia una lógica de generación de estímulos y adquisición de datos para la realización de las pruebas sobre los filtros. Por otro lado, en la PL debe existir una sección configurable donde se deben instanciar los bloques de procesamiento.
- Memoria estática (SD): espacio de memoria donde se deben almacenar el banco de filtros, los reportes de las pruebas ejecutadas, archivos de configuración del sistema y los datos de las pruebas recientemente realizadas por el usuario.
- Memoria RAM: a ser utilizada para la generación de reportes.

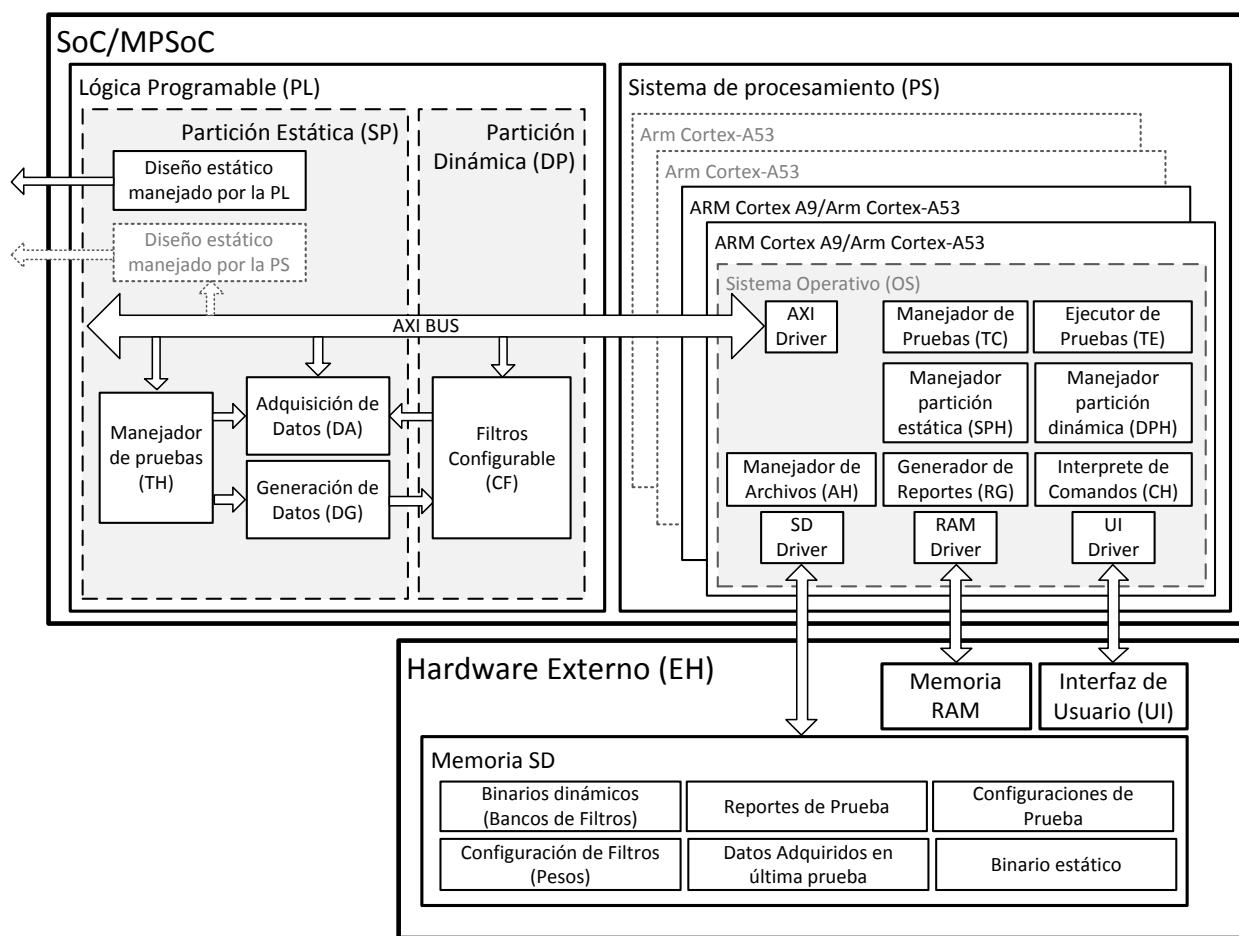


Figura 1. Diagrama en bloques.

## 2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Franco Alcaraz	INVAP	-
Responsable	Carlos Jorge Maffrand	FIUBA	Alumno
Orientador	Gonzalo Lavigna	FIUBA	Director Trabajo Final
Usuario final	Ingeniero de desarrollo	INVAP	-

Cuadro 1. Identificación de los interesados.

- Cliente: Franco Alcaraz, es un profesional altamente calificado en la industria aeroespacial y defensa.
- Orientador: Gonzalo Lavigna, es egresado de la Especialización en Sistemas Embebidos de la FIUBA, posee muchos años de experiencia desarrollando proyectos en la industria aeroespacial y de defensa. Probablemente esté viviendo en Europa, por lo que hay que tener en cuenta ese hecho cuando se decida contactarlo.

## 3. Propósito del proyecto

El propósito de este proyecto es realizar una evaluación tecnológica de la re-configuración dinámica de dispositivos de lógica programable, con el fin de establecer criterios de uso para desarrollos aeroespaciales y de radares. Adicionalmente, el cliente desea adquirir la capacidad de desarrollar una aplicación con esa tecnología de modo de subir el Technology Readiness Level (TRL) que se posee dentro de INVAP.

## 4. Alcance del proyecto

El proyecto descripto en este plan de trabajo deberá brindar las siguientes funcionalidades:

- Administrar un banco de filtros previamente generados en un sistema de archivos.
- Seleccionar el filtro que se planea utilizar.
- Realizar la re-configuración parcial de la lógica programable del SoC/MPSoC, implementando el filtro previamente seleccionado.
- Configurar el entorno de pruebas presente en la parte estática de la PL.
- Ejecutar pruebas configuradas en las cadenas de filtros elegidas.
- Adquirir los datos generados en las pruebas.
- Emitir reportes sobre las pruebas ejecutadas.

El presente proyecto no incluye las siguientes funcionalidades:

- Generar archivos binarios de filtros para la configuración dinámica de la PL.
- Rutinas de verificación, validación y evaluación de performance de las pruebas realizadas en los filtros configurados en la PL.
- Evaluación de los tiempos de re-configuración.
- Almacenamiento de grandes volúmenes de datos.

## 5. Supuestos del proyecto

Se supone que el software correrá en una plataforma de hardware tipo System on Chip de la línea Xilinx Zynq (ARM+FPGA).

El responsable del proyecto dispone de una placa ZedBoard para la realización del desarrollo. Con el fin de que el banco de pruebas sea implementado en una tecnología más moderna, se debe evaluar si INVAP dispone de un hardware de desarrollo que contenga un MPSoC.

Se asume que se dispondrá de la plataforma de desarrollo durante el tiempo que dure el proyecto para poder depurar el diseño sobre el hardware.

Se va a hacer uso de Vivado y las herramientas adicionales de dicho entornos de desarrollo para dispositivos de la marca Xilinx. Por lo que, se supone que se cuentan con las licencias para tal fin.

Se supone que se va a hacer uso de HDL Coder para obtener los filtros para el prototipo del sistema. Por lo que, se supone que se cuentan con las licencias para su uso.



## 6. Requerimientos

Para comprender los requerimientos del proyecto se enumeran los siguientes acrónimos:

- API: Application Programming Interface.
- AXI: Advanced eXtensible Interface.
- HDL: Hardware Description Language.
- MPSoC: Multi-Processor System on a Chip.
- PL: Programmable Logic en un SoC u MPSoC.
- PS: Processing System en un SoC o MPSoC.
- SoC: System on a Chip.
- TBD: To Be Defined.
- UART: Universal Asynchronous Receiver-Transmitter.
- VHDL: Very High Speed Integrated Circuit Hardware Description Language.

### 1. Requerimientos de interfaces externas

- 1.1. El sistema debe implementar una interfaz de usuario por consola.
- 1.2. La consola de interfaz de usuario debe ser implementada mediante UART y/o Ethernet.
- 1.3. En la partición estática de la PL se debe implementar una interfaz serie que tenga funcionamiento independiente de la PS.
- 1.4. En la partición estática de la PL se debe implementar una interfaz serie que tenga funcionamiento controlado por la PS.

### 2. Requerimientos de interfaces internas

- 2.1. El software alocado en la PS debe comunicarse con la PL mediante una interfaz AXI.
- 2.2. El software alocado en la PS debe implementar un sistema de archivos en una memoria SD.
- 2.3. El software alocado en la PS debe manejar una interfaz con una memoria RAM.
- 2.4. El software alocado en la debe tener una interfaz para re-configurar dinámicamente la porción de la PL dedicada para tal fin.
- 2.5. El software alocado en la PS debe tener una interfaz para configurar el sector de PL estático.

### 3. Requerimientos de almacenamiento de información (sistema de archivos)

- 3.1. El sistema debe almacenar reportes de las pruebas ejecutadas en el sistema de archivos.
- 3.2. El sistema debe almacenar pre-configuraciones para el entorno de pruebas en el sistema de archivos.

- 3.3. El sistema debe almacenar el archivo de configuración del sector estático de la PL en el sistema de archivos.
- 3.4. El sistema debe almacenar al menos dos archivos binarios (filtros previamente generados) de configuración del sector dinámico de la PL en el sistema de archivos.
- 3.5. El sistema debe almacenar archivos de configuración (pesos de los filtros) del sector dinámico de la PL en el sistema de archivos.
- 3.6. El sistema debe almacenar los datos obtenidos de al menos la última ejecución de pruebas en el sistema de archivos.
4. Requerimientos generales de HDL
  - 4.1. La lógica programable debe ser codificada en lenguaje VHDL.
  - 4.2. En la PL se debe generar una partición para uso dinámico.
  - 4.3. En la PL se debe generar una partición para uso estático.
  - 4.4. Para la comunicación entre bloques de lógica se debe utilizar el bus AXI.
5. Requerimientos de HDL estático
  - 5.1. En la partición estática se debe implementar una funcionalidad de lógica programable que sea independiente de la PS.
  - 5.2. En la partición estática se debe implementar una funcionalidad de lógica programable que sea manejada por la PS.
  - 5.3. En la partición estática se debe implementar una funcionalidad de lógica programable para la generación de señales de excitación de los filtros.
  - 5.4. En la partición estática se debe implementar una funcionalidad de lógica programable para la adquisición de datos generados por las salidas de los filtros.
  - 5.5. En la partición estática se debe implementar una funcionalidad de lógica programable que permita la configuración de las pruebas a realizar en el banco de filtros.
6. Requerimientos de configuración del entorno de pruebas
  - 6.1. El sistema debe instanciar el archivo de configuración de sector estático de la PS.
  - 6.2. El sistema debe permitir al usuario la selección del archivo de configuración de sector dinámico de la PL.
  - 6.3. El sistema debe permitir al usuario la solicitud de re-configuración dinámica de la porción de la PL dedicada a tal fin.
  - 6.4. El sistema debe ejecutar la re-configuración dinámica.
  - 6.5. El sistema debe permitir al usuario la configuración de la prueba en la parte estática de la PS haciendo uso de los siguientes parámetros:
    - a. Duración de la prueba.
    - b. Ancho de palabra del filtro configurado.
    - c. Tipo de onda generada para la prueba.
  - 6.6. El sistema debe escribir los registros de configuración de los bloques implementados en la parte estática de la PS:
    - a. Generación de datos de excitación.
    - b. Configuración de test.
    - c. Adquisición de datos.

- 6.7. El sistema debe permitir al usuario la opción de ejecutar pruebas sin generar reportes o almacenar los datos generados.
- 6.8. El sistema debe permitir al operador actualizar los archivos que conforman el banco de filtros. Dichos archivos son binarios previamente generados en la herramienta de síntesis de HDL.
7. Requerimientos funcionales (ejecución de pruebas)
  - 7.1. El sistema debe permitir al usuario la solicitud de ejecución de pruebas en la PL. Tales pruebas pueden ser:
    - a. Respuesta de filtro a estímulos generados.
    - b. Comportamiento de la lógica en la PL estática cuando se realiza la configuración dinámica.
    - c. Evaluación del tiempo de re-configuración.
  - 7.2. El software del PS debe escribir los registros de los bloques de lógica programable de generación de datos de excitación, configuración de test y adquisición de datos implementados en la parte estática de la PS de modo tal que se dispare la ejecución de la prueba configurada.
  - 7.3. El software del PS debe capturar los datos adquiridos tras la ejecución de la prueba y almacenarlos en el sistema de archivos.
  - 7.4. El software del PS debe leer los registros de los bloques de generación de datos de excitación, configuración de test y adquisición de datos.
  - 7.5. El software del PS debe generar un reporte de pruebas conteniendo mínimamente la siguiente información:
    - a. Fecha de ejecución.
    - b. Tiempo de ejecución.
    - c. Configuración del sistema de pruebas.
    - d. Resultado de las pruebas realizadas.
    - e. Filtro seleccionado e instanciado en la PL dinámica.
    - f. Versión del archivo instanciado en la PL estática.
  - 7.6. El software del PS debe controlar que la prueba configurada por el usuario no genere datos superiores a un tamaño de 10 kBytes.
8. Requerimientos de rendimiento
  - 8.1. Una vez iniciada una prueba el reporte de la misma deberá ser generado en menos de un minuto.
9. Requerimientos de diseño
  - 9.1. Se debe utilizar la placa de desarrollo ZedBoard para implementar el desarrollo en un SoC. La placa de desarrollo podría ser modificada por una que contenga un MPSoC, de acuerdo a la disponibilidad de la misma en INVAP.
10. Requerimientos de confiabilidad
  - 10.1. El sistema debe asegurar su correcto funcionamiento en condiciones normales de operación durante al menos una semana de uso continuo (sin ser reiniciado).
11. Requerimientos de documentación
  - 11.1. Se debe generar un manual de usuario para el sistema de pruebas de bancos de filtros re-configurables.

## 7. Historias de usuarios (*Product backlog*)

A continuación se presentan cuatro historias de usuario para el proyecto. Vale la pena destacar que los *Story points*, que se le atribuyen a cada una de las historias, son calculados como el número de Fibonacci mayor o igual a la suma de la estimación de carga de trabajo, complejidad y conocimiento previo.

- Como *ingeniero de desarrollo* quiero *realizar la configuración dinámica en lógica programable* para evaluar su implementación en diseños complejos.

Carga de trabajo: 5 puntos. Complejidad: 3 puntos. Conocimiento previo: 3 puntos.

$$5 + 3 + 3 = 11 \rightarrow 13$$

*Story points: 13*, la historia de usuario permite evaluar la complejidad de generar diseños de lógica programable con re-configuración parcial, para con esto evaluar riesgos, ventajas y desventajas de su utilización en proyectos de alto costo y alcance.

- Como *ingeniero de desarrollo* quiero *realizar la configuración dinámica en lógica programable* para evaluar el tiempo que toma dicho proceso.

Carga de trabajo: 3 puntos. Complejidad: 2 puntos. Conocimiento previo: 2 puntos.

$$3 + 2 + 2 = 7 \rightarrow 8$$

*Story points: 8*, la historia de usuario permite dimensionar el tiempo de re-configuración que es un dato de importancia para la evaluación de su uso en sistemas de procesamiento que sean demandantes en su tiempo de ejecución.

- Como *ingeniero de desarrollo* quiero *realizar la configuración dinámica en lógica programable* para evaluar el funcionamiento del resto del dispositivo de lógica programable durante el proceso de re-configuración.

Carga de trabajo: 2 puntos. Complejidad: 2 puntos. Conocimiento previo: 2 puntos.

$$2 + 2 + 2 = 6 \rightarrow 8$$

*Story points: 5*, la historia de usuario permite validar que no se afecta el funcionamiento de la lógica adyacente durante el proceso de re-configuración.

- Como *ingeniero de desarrollo* quiero *realizar pruebas de desempeño sobre filtros de procesamiento en hardware* para evaluar y caracterizar su funcionamiento.

Carga de trabajo: 2 puntos. Complejidad: 2 puntos. Conocimiento previo: 1 puntos.

$$2 + 2 + 1 = 5 \rightarrow 5$$

*Story points: 5*, la historia de usuario permite generar una métrica de funcionamiento de los filtros.

## 8. Entregables principales del proyecto

Los entregables del proyecto son:

- Código fuente c en el repositorio con control de versiones *Github*.
- Documentación del código c con *Doxygen*.
- Código fuente VHDL en el repositorio con control de versiones *Github*.
- Documentación del código VHDL con *TerosHDL*.
- Manual de usuario del sistema de pruebas.
- Ficha de lecciones aprendidas durante el desarrollo.
- Código fuente del firmware y software desarrollado para el PS.
- Código fuente del HDL generado para la PL estática.
- Código fuente de los HDLs generados para la PL dinámica.
- Memoria del Trabajo final.

## 9. Desglose del trabajo en tareas

### 1. Planificación

- 1.1. Estudio del problema general (30 horas)
- 1.2. Relevamiento y estudio de los requerimientos (20 horas)
- 1.3. Documentación de requerimientos y especificación del trabajo (20 horas)
- 1.4. Planificación del proyecto (20 horas)

### 2. Proceso de adquisición de conocimientos

- 2.1. Estudio del SoC/MPSoC (20 horas)
- 2.2. Estudio de técnicas de re-configuración parcial (40 horas)
- 2.3. Implementación de sistemas operativos y sistemas de archivos en SoC/MPSoC (40 horas)
- 2.4. Estudio de periférico UART (10 horas)
- 2.5. Estudio de periférico Ethernet (20 horas)
- 2.6. Estudio de periférico RAM (30 horas)
- 2.7. Estudio de periférico AXI (10 horas)
- 2.8. Estudio de periférico SD (20 horas)

### 3. Desarrollo de aplicaciones

- 3.1. Desarrollo de comunicación con PL (40 horas)
- 3.2. Desarrollo de consola intérprete de comandos de usuario (20 horas)
- 3.3. Desarrollo de configurador de PL estática (20 horas)
- 3.4. Desarrollo de configurador de PL dinámica (20 horas)
- 3.5. Desarrollo de configurador y ejecutor de pruebas (API) (40 horas)
- 3.6. Desarrollo de generador de reportes (40 horas)

### 4. Desarrollos de HDL

- 4.1. Desarrollo de partición PL (40 horas)
- 4.2. Diseño de puerto de comunicación en PL estática de funcionamiento independiente de la PS (10 horas)
- 4.3. Diseño de puerto de comunicación en PL estática de funcionamiento dependiente de la PS (20 horas)
- 4.4. Diseño del bloque de configuración y ejecución de pruebas (HDL) (40 horas)
- 4.5. Implementación de generador de datos de excitación (30 horas)
- 4.6. Implementación de adquirente de datos (30 horas)
- 4.7. Implementación de bus AXI e integración general (40 horas)
- 4.8. Implementación de filtros de procesamiento de datos para demo de funcionamiento del sistema (40 horas)

### 5. Proceso de cierre

- 5.1. Escritura de memoria de Trabajo final (40 horas)
- 5.2. Escritura de manual de usuario (30 horas)

- 5.3. Confección de ficha de lecciones aprendidas (10 horas)
- 5.4. Confección de presentación y defensa pública (10 horas)

Cantidad total de trabajo estimado para el proyecto: (800 horas)

## 10. Diagrama de Activity On Node

En la Figura 2 se encuentran las tareas componentes de la WBS, en donde el tiempo de realización de las mismas está expresado en horas. El diagrama de AoN deja de manifiesto el camino crítico mediante líneas continuas y tareas griseadas. El mismo consta de un tiempo de realización de 380 horas.

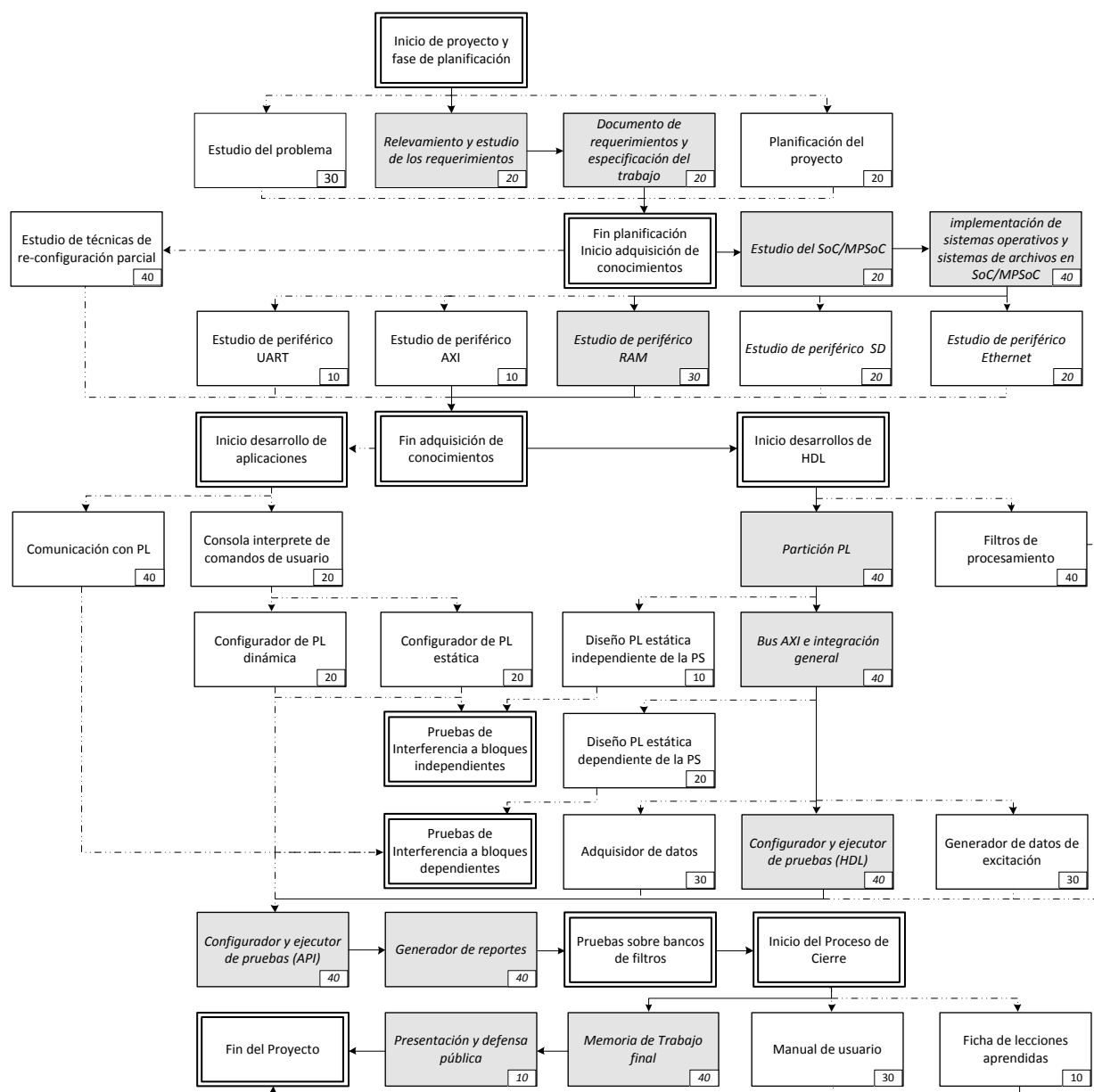


Figura 2. Diagrama en Activity on Node.



## 11. Diagrama de Gantt

En la figura 3 y la figura 4 se muestra el diagrama de Gantt del proyecto, el mismo fue obtenido configurando la jornada laboral de 15 horas semanales.

En el diagrama de Gantt no se observan tareas realizadas de manera paralela, esto se debe a que el proyecto cuenta con un único recurso. Por lo que, queda evidenciada una diferencia con el diagrama de AoN, en donde queda de manifiesto la dependencia entre las tareas según su relación técnica óptima.

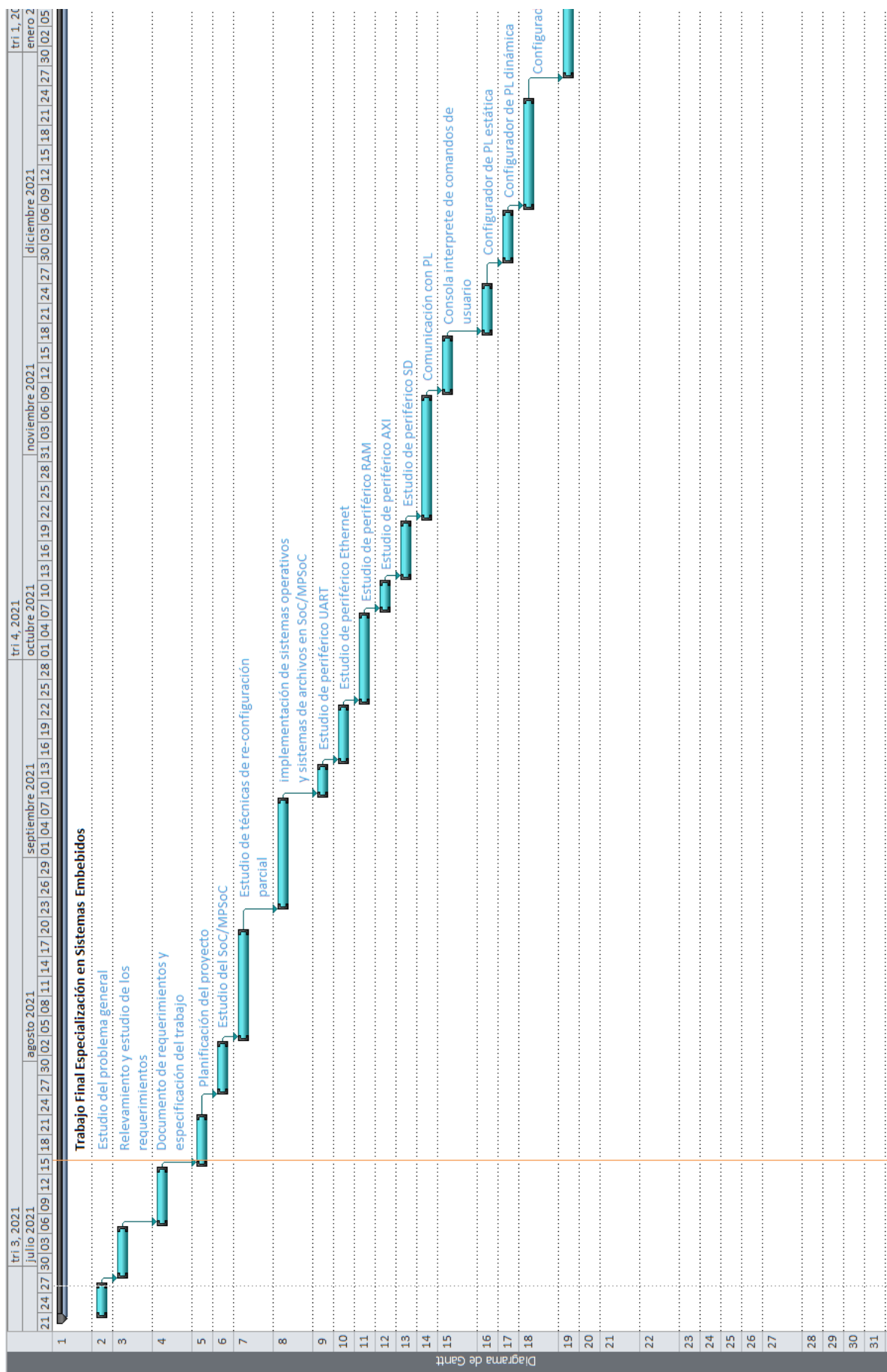


Figura 3. Diagrama de Gantt fase inicial.

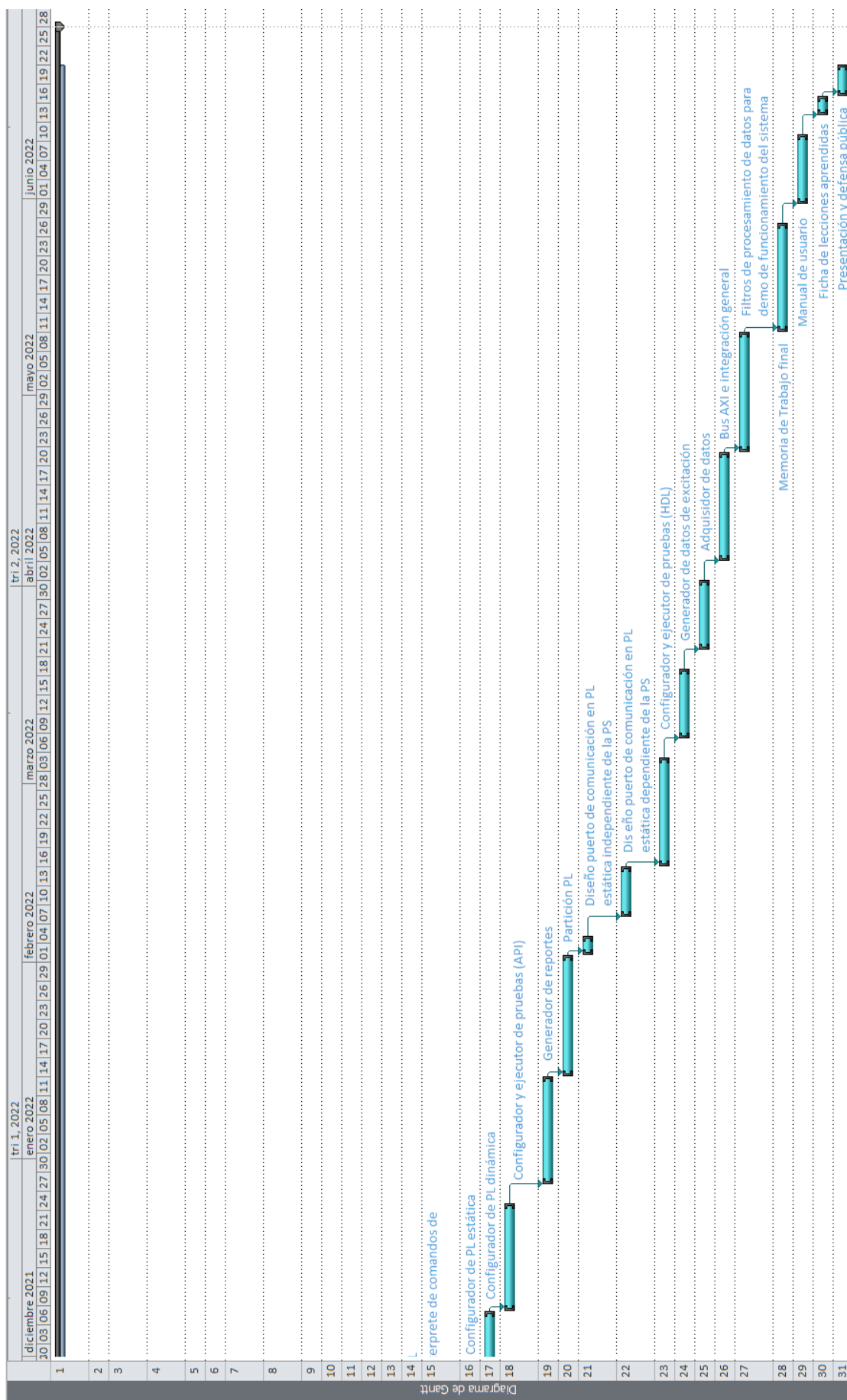


Figura 4. Diagrama de Gantt fase final.

## 12. Presupuesto detallado del proyecto

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario [\$]	Valor total [\$]
Ingeniería	800 horas	3.000	2.400.000
Imprevistos			160.000
SUBTOTAL			2.560.000
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario [\$]	Valor total [\$]
Costos indirectos			1.320.000
SUBTOTAL			1.320.000
MATERIALES			
Descripción	Cantidad	Valor unitario [\$]	Valor total [\$]
ZCU102 Development Board / ZedBoard	1	249.000	249.000
PC de desarrollo	1	300.000	300.000
SUBTOTAL			549.000
TOTAL			4.429.000

Cuadro 2. Presupuesto del proyecto.

## 13. Gestión de riesgos

### 1. Identificación de los riesgos y estimación de sus consecuencias:

Riesgo 1: Se cuenta con solo una ZedBoard propiedad del responsable del trabajo.

- Severidad (S): 9  
La pérdida, extravío o rotura de la placa de desarrollo generaría una imposibilidad de continuar con el avance del diseño en etapas avanzadas del proyecto.
- Probabilidad de ocurrencia (O): 4  
El hardware de desarrollo puede sufrir averías dado que es utilizado en un escritorio y no en un laboratorio. Así mismo, puede extraviarse o ser robado.

Riesgo 2: Recurso único de ejecución del proyecto.

- Severidad (S): 10  
Se cuenta con un solo recurso para realizar las tareas. El mismo trabaja 45 horas semanales. Por lo que, las 15 horas semanales para la ejecución del proyecto provienen de su tiempo libre y podrían surgir inconvenientes de índole personal.
- Probabilidad de ocurrencia (O): 2  
Se debe a que son hechos fortuitos, no conocidos previamente por lo que la probabilidad de ocurrencia es baja.

Riesgo 3: Licencias de entornos de desarrollo.

- Severidad (S): 7  
Entornos de desarrollo como Vivado y HDL Coder cuentan con licencias de pago para su utilización. No contar con las mismas podrían hacer que el tiempo de desarrollo suba.
- Probabilidad de ocurrencia (O): 4  
INVAP cuenta con las licencias en estos momentos, pero la renovación de las mismas son externas a este proyecto.

Riesgo 4: Recurso sin experiencia en el manejo de sistemas operativos y aplicaciones en sistemas operativos.

- Severidad (S): 6  
El desarrollo necesita que se generen varias aplicaciones que se ejecutan dentro de un sistema operativo.
- Probabilidad de ocurrencia (O): 8  
El recurso ejecutor del proyecto tiene poca experiencia manejando sistemas operativos.

Riesgo 5: Dificultades funcionales entre PL estática y PL dinámica.

- Severidad (S): 7  
Se pueden encontrar limitaciones en la comunicación de ambas particiones de la lógica programable, esto podría generar un cambio en la arquitectura del banco de pruebas.
- Probabilidad de ocurrencia (O): 4  
Se leyeron publicaciones que indican que la forma de comunicar ambas particiones es correcta. Este hecho hace que la probabilidad de ocurrencia de esto sea baja.

Riesgo 6: Tiempos de desarrollo.

- Severidad (S): 7  
Tiempos mal estimados en la planificación pueden influir directamente en la fecha de finalización del proyecto.
- Probabilidad de ocurrencia (O): 7  
En revisiones del plan de trabajo se hizo notar que probablemente el tiempo estimado para las tareas pueda ser muy optimista.

2. Tabla de gestión de riesgos:

Riesgo	S	O	RPN	S*	O*	RPN*
Una sola ZedBoard propiedad del responsable del trabajo	9	4	<b>36</b>	9	1	<i>9</i>
Recurso único de ejecución del proyecto	10	2	20	10	2	<i>20</i>
Licencias de entornos de desarrollo	7	4	<b>28</b>	5	4	<i>20</i>
Recurso sin experiencia en el manejo de SO	6	8	<b>40</b>	6	2	<i>12</i>
Dificultades funcionales entre PL estática y PL dinámica	7	4	<b>28</b>	4	4	<i>16</i>
Tiempos de desarrollo	7	7	<b>49</b>	5	4	<i>20</i>

Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a  $RPN = 25$ . Nota: los valores marcados con (\*) en la tabla corresponden luego de haber aplicado la mitigación.

3. Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: Se cuenta con solo una ZedBoard propiedad del responsable del trabajo.

- Severidad (S): 9  
No se registran cambios.
- Probabilidad de ocurrencia (O): 1  
*Mitigación:* INVAP cuenta con numerosas placas de desarrollo que podrían ser utilizadas para el proyecto.

Riesgo 2: Recurso único de ejecución del proyecto.

- Severidad (S): 10  
No se registran cambios.
- Probabilidad de ocurrencia (O): 2  
No se registran cambios.

Riesgo 3: Licencias de entornos de desarrollo.

- Severidad (S): 5  
*Mitigación:* se puede limitar el banco de filtros inicial a implementaciones simples que se encuentran en las librerías de IP Cores que no necesitan licencia de Vivado o HDL Coder.
- Probabilidad de ocurrencia (O): 4  
No se registran cambios.

Riesgo 4: Recurso sin experiencia en el manejo de sistemas operativos y aplicaciones en sistemas operativos.

- Severidad (S): 6  
No se registran cambios.
- Probabilidad de ocurrencia (O): 2  
*Mitigación:* durante el transcurso de la especialización existen tres asignaturas sobre sistemas operativos y hay una optativa sobre aplicaciones para sistemas operativos.

Riesgo 5: Dificultades funcionales entre PL estática y PL dinámica.

- Severidad (S): 4  
*Mitigación:* se realiza de manera temprana esta tarea para estimar si hay necesidad de ajustar la arquitectura del banco de pruebas.
- Probabilidad de ocurrencia (O): 4  
No se registran cambios.

Riesgo 6: Tiempos de desarrollo.

- Severidad (S): 4  
*Mitigación:* se pueden disminuir las funcionalidades del primer prototipo del banco de pruebas y se tiene una flotabilidad en la ejecución del desarrollo.
- Probabilidad de ocurrencia (O): 5  
*Mitigación:* se puede hacer una evaluación temprana de los tiempos de desarrollo de acuerdo a lo que hayan llevado las tareas iniciales.

## 14. Gestión de la calidad

### 1. Requerimientos de interfaces externas

- *Verificación:* durante el desarrollo se realizan pruebas unitarias y de integración sobre cada una de las interfaces requeridas.
- *Validación:* el cliente utilizará las interfaces externas para el uso del prototipo.

### 2. Requerimientos de interfaces internas

- *Verificación:* restricciones de diseño que serán utilizadas para el desarrollo.
- *Validación:* no son requerimientos funcionales por lo que el cliente no necesariamente debe validarlos.

### 3. Requerimientos de almacenamiento de información (sistema de archivos)

- *Verificación:* durante el desarrollo se realizan pruebas unitarias y de integración sobre cada una de las funcionalidades requeridas.
- *Validación:* el cliente desarrolla pruebas sobre las funcionalidades sobre el producto final.

### 4. Requerimientos generales de HDL

- *Verificación:* estas restricciones de diseño serán tenidas en cuenta durante la realización de los bloques funcionales de HDL.
- *Validación:* el cliente puede inspeccionar el código y validar los requerimientos de implementación.

### 5. Requerimientos de HDL estático

- *Verificación:* durante el desarrollo se realizan pruebas unitarias y de integración sobre cada bloques de HDL requeridos.
- *Validación:* el cliente desarrolla pruebas sobre las funcionalidades sobre el producto final.

### 6. Requerimientos de configuración del entorno de pruebas

- *Verificación:* durante el desarrollo se realizan pruebas unitarias y de integración sobre cada una de las funcionalidades requeridas.
- *Validación:* el cliente desarrolla pruebas sobre las funcionalidades sobre el producto final.

### 7. Requerimientos funcionales (ejecución de pruebas)

- *Verificación:* durante el desarrollo se realizan pruebas unitarias y de integración sobre cada una de las funcionalidades requeridas.
- *Validación:* el cliente desarrolla pruebas sobre las funcionalidades sobre el producto final.

### 8. Requerimientos de rendimiento

- *Verificación:* se generan reportes en donde se muestra el tiempo de procesamiento.
- *Validación:* se generan reportes en donde se muestra el tiempo de procesamiento.



9. Requerimientos de diseño

- *Verificación:* por inspección visual se constata la utilización de la ZedBoard (o la placa dispuesta por el cliente).
- *Validación:* por inspección visual se constata la utilización de la ZedBoard (o la placa dispuesta por el cliente).

10. Requerimientos de confiabilidad

- *Verificación:* durante el desarrollo se harán pruebas parciales.
- *Validación:* el cliente puede realizar un uso continuo del banco de pruebas para validar el requerimiento de confiabilidad.

11. Requerimientos de documentación

- *Verificación:* durante el transcurso del desarrollo se generará información técnica parcial que será utilizada para confeccionar documentación requerida: Memoria Técnica, Ficha de lecciones aprendidas y Manual de usuario.
- *Validación:* el cliente realizará la lectura y revisión de los documentos a entregar: Plan de Trabajo, Memoria Técnica, Ficha de lecciones aprendidas y Manual de usuario.

## 15. Procesos de cierre

Los aspectos a trabajar durante todo el proceso de cierre son los siguientes:

- Presentación oral de los principales resultados alcanzados.
- Contrastar el Plan de Trabajo con la Memoria Técnica, los documentos entregables e hitos parciales.
- Determinar el grado de cumplimiento de la planificación original. En lo referente a las estimaciones de esfuerzo previamente realizadas como los riesgos contemplados.
- Analizar el grado de cumplimiento de los requerimientos del proyecto.
- Identificar técnicas y procedimientos útiles e inútiles que se hayan implementado.
- Mostrar los problemas que surgieron y las soluciones encontradas para los mismos.
- Se archivará toda la documentación de la gestión del proyecto para ser fácilmente reutilizada en la planificación de futuros proyectos.
- Establecer pautas para desarrollos futuros.
- Realizar los agradecimientos correspondientes.

La presentación de los tópicos previamente mencionados estará a cargo del Ing. Carlos Jorge Maffrand.

Al finalizar el proyecto se realiza una defensa publica donde estarán presentes:

- El director del proyecto (Esp. Ing. Gonzalo Lavigna).
- Los jurados.
- El equipo de trabajo del proyecto.
- El Responsable del proyecto (Ing. Carlos Jorge Maffrand).

La misma estará a cargo del Dr. Ing. Ariel Lutenberg en su calidad de Director de la Especialización en Sistemas Embebidos de la FIUBA.