



Desarrollo de un Sistema de Comunicación Basado en el Estándar CCSDS para Satélites

Autor:

Ing. Gonzalo Lavigna

Director:

Ing. Roberto Cibils (FIUBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 11 de marzo de 2025 y el 22 de abril de 2025.*

Índice

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

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	11 de marzo de 2025
1	Se completa hasta el punto 5 inclusive	16 de marzo de 2025

Acta de constitución del proyecto

Buenos Aires, 11 de marzo de 2025

Por medio de la presente, se acuerda con el Ing. Gonzalo Lavigna que su Trabajo Final de la Carrera de Especialización en Sistemas Embebidos se titulará “Desarrollo de un Sistema de Comunicación Basado en el Estándar CCSDS para Satélites” y consistirá en la implementación de un prototipo de comunicación satelital basado en normas CCSDS con una FPGA. El trabajo tendrá un presupuesto preliminar estimado de 640 horas y un costo estimado de mil dolares estadounidenses (USD 1000), con fecha de inicio el 11 de marzo de 2025 y fecha de presentación pública el 15 de diciembre de 2025.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Ing. Roberto Cibils
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

El presente proyecto aborda el diseño y la implementación de un sistema digital para comunicaciones satelitales, basado en el estándar internacional establecido por el *Consultative Committee for Space Data Systems* (CCSDS). La motivación principal radica en la creciente demanda de soluciones estandarizadas y confiables para la comunicación satelital, especialmente orientadas a pequeños satélites y misiones espaciales de bajo presupuesto, donde las soluciones tradicionales resultan prohibitivas en términos de costo y complejidad.

Este proyecto se enmarca en un esfuerzo personal y académico, como Trabajo Final de la Carrera de Especialización en Sistemas Embebidos. No existen condiciones especiales relacionadas con financiamiento externo, acuerdos de confidencialidad o propiedad intelectual específica y el objetivo es proporcionar un prototipo abierto y documentado para futuras implementaciones en diferentes organizaciones interesadas en el ámbito aeroespacial.

El desafío principal radica en implementar eficientemente las especificaciones del estándar CCSDS en *hardware Field-Programmable Gate Array* (FPGA), cumpliendo estrictos requerimientos de confiabilidad, robustez y eficiencia energética. El estado del arte actual muestra diversas implementaciones, generalmente complejas y costosas, orientadas a satélites de mayor tamaño o con mayores recursos disponibles. Este proyecto propone un enfoque simplificado y eficiente que destaca por su simplicidad, accesibilidad económica y facilidad de integración en satélites pequeños.

En la Figura 1 se presenta el diagrama en bloques del sistema, compuesto por los siguientes módulos principales:

- *Ethernet*: interfaz física externa *ethernet* aquí es donde la FPGA se conecta con *hardware* externo, ya sea una estación terrena o una red local.
- *Universal Asynchronous Receiver-Transmitter* (UART) : Es la interfaz de comunicación serial que sirve típicamente para enviar comandos al FPGA y recibir datos del estado del sistema, desde una PC o terminal externa.
- Bloque *ethernet*: gestiona la comunicación de datos *ethernet* a nivel digital dentro de la FPGA. Convierte las señales físicas *ethernet* en datos digitales que puedan ser procesados internamente por el sistema y viceversa.
- Control: se encarga del manejo interno de comandos recibidos desde el bloque UART y la generación de datos de estado del sistema. Es el bloque de gestión de mensajes, comandos y monitoreo del sistema.
- Enrutador de paquetes: este bloque cumple la función de distribución y conmutación interna. Es decir, recibe paquetes de datos *ethernet* desde el bloque *ethernet* y determina hacia dónde dirigirlos: hacia la cadena transmisora (codificador CCSDS) o la cadena receptora (decodificador CCSDS).
- Transformación CCSDS: convierte los datos desde el formato *ethernet* hacia tramas del estándar CCSDS Advanced Orbiting Systems (AOS). Básicamente, es una traducción hacia el protocolo estandarizado espacial.
- Codificador CCSDS : codifica las tramas CCSDS AOS generadas previamente, añadiendo información adicional para detección y corrección de errores (usualmente utilizando Reed-Solomon). Este bloque asegura robustez y confiabilidad en la comunicación.

- Modulador: acondiciona las tramas codificadas digitalmente para simular su transmisión hacia un canal físico. En este caso, realiza una modulación lógica que simula el comportamiento real de transmisión.
- Decodificador CCSDS : decodifica las tramas recibidas, verificando su integridad y corrigiendo errores mediante los códigos Reed-Solomon introducidos por el codificador CCSDS.
- Transformación *ethernet*: convierte nuevamente las tramas CCSDS AOS decodificadas hacia paquetes *ethernet*, facilitando así su distribución posterior hacia interfaces externas o internas.

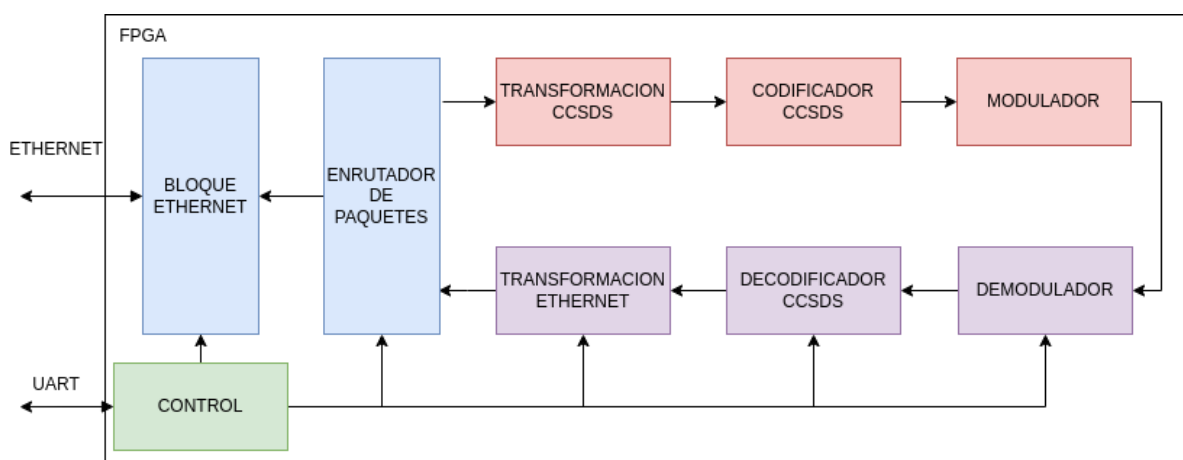


Figura 1. Diagrama en bloques del sistema.

Este proyecto se encuentra en una etapa inicial del ciclo de vida, enfocándose en la validación mediante simulaciones en un entorno controlado previo a cualquier implementación física más avanzada.

El cliente objetivo lo constituyen organizaciones o investigadores dedicados al desarrollo de misiones satelitales pequeñas, quienes valoran soluciones abiertas, económicas y estandarizadas para facilitar la integración y reducir riesgos en sus misiones.

La innovación principal del proyecto reside en simplificar el estándar CCSDS para aplicaciones de bajo costo y alta eficiencia, facilitando su adopción y garantizando interoperabilidad en futuras misiones espaciales.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Cliente	-	-	-
Responsable	Ing. Gonzalo Lavigna	FIUBA	Alumno
Colaboradores	-	-	-
Orientador	Ing. Roberto Cibils	FIUBA	Director del Trabajo Final
Usuario final	Industria satelital	-	-

- Orientador: el Ing. Roberto Cibils es experto en la temática y va a ayudar con la definición de los requerimientos y el desarrollo del *firmware* del embebido. Roberto Cibils cuenta con basta experiencia en sistemas robustos para aplicaciones espaciales y calificación de partes.
- Responsable: será quien dispondrá del conocimiento y desarrollo del proyecto.
- Usuario final: El usuario final de este proyecto serían principalmente, ingenieros, universidades y organizaciones dedicadas al desarrollo e implementación de misiones satelitales, especialmente aquellas orientadas a pequeños satélites o misiones de bajo costo.

3. Propósito del proyecto

El propósito del proyecto es el desarrollo de un prototipo de un sistema de comunicación basado en el estándar CCSDS dentro de una FPGA que tiene como objetivo proporcionar una solución tecnológica eficiente y accesible para aplicaciones satelitales, especialmente orientada a misiones de satélites pequeños y de bajo costo. De esta forma, se pretende reducir significativamente la complejidad y los costos asociados a sistemas tradicionales, favoreciendo la investigación, el desarrollo y el despliegue de futuras misiones espaciales por parte de organizaciones académicas y científicas.

4. Alcance del proyecto

El presente proyecto incluye:

- Diseño e implementación de un sistema digital de comunicación satelital basado en el estándar CCSDS.
- Integración de algoritmos de codificación y decodificación Reed-Solomon según CCSDS.
- Desarrollo integral del sistema sobre una placa de desarrollo de FPGA.
- Simulación del canal de comunicación utilizando un esquema de *loopback* lógico.
- Validación funcional mediante herramientas de simulación.
- Validación de la implementación utilizando herramientas de síntesis como Vivado.
- Desarrollo de una interfaz mínima utilizando UART para control y monitoreo del sistema.
- Documentación técnica completa del diseño y código fuente.
- Liberación del código bajo licencia de código abierto.

El presente proyecto no incluye:

- Implementación física de *hardware* de radiofrecuencia (RF), ni diseño de antenas.
- Integración real con *hardware* satelital.
- Realización de pruebas en ambientes espaciales reales o estaciones terrenas operativas.

- Desarrollo de software adicional para análisis avanzado o procesamiento de datos fuera del alcance del sistema propuesto.
- Utilización de conversor analógico digital.
- Utilización de conversor digital analógico.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Disponibilidad de recursos:
 - Se dispondrá oportunamente de una placa de desarrollo que contenga una FPGA.
 - Cable *Universal Serial Bus* USB a UART, que permitirá establecer comunicación entre la FPGA y una terminal externa (por ejemplo, una computadora).
 - Cable *ethernet* (categoría 5e o superior), con conectores RJ-45 en ambos extremos.
 - El proyecto contempla la utilización de módulos proporcionados por terceros para realizar las funciones específicas de codificación y decodificación Reed-Solomon según lo establecido en el estándar CCSDS. Estos módulos externos, previamente validados y disponibles abiertamente, permitirán acelerar el desarrollo del sistema.
- Experiencia técnica:
 - El responsable posee un nivel adecuado de conocimiento y experiencia previa en leguajes de descripción de *hardware* como VHDL o Verilog, necesarios para el diseño, implementación y validación lógica del sistema propuesto.
 - El responsable dispone de la capacidad técnica para integrar y validar adecuadamente módulos de terceros, como los bloques de codificación y decodificación Reed-Solomon.
 - El responsable cuenta con conocimientos sólidos en el manejo del estándar CCSDS y familiaridad con las herramientas de desarrollo FPGA, como Vivado y ModelSim.
 - Se dispondrán de las 640 horas requeridas para realizar el proyecto.

6. Requerimientos

Los requerimientos deben enumerarse y de ser posible estar agrupados por afinidad, por ejemplo:

1. Requerimientos funcionales:

- 1.1. El sistema debe...
- 1.2. Tal componente debe...
- 1.3. El usuario debe poder...

2. Requerimientos de documentación:

- 2.1. Requerimiento 1.
- 2.2. Requerimiento 2 (prioridad menor)

3. Requerimiento de testing...
4. Requerimientos de la interfaz...
5. Requerimientos interoperabilidad...
6. etc...

Leyendo los requerimientos se debe poder interpretar cómo será el proyecto y su funcionalidad.

Indicar claramente cuál es la prioridad entre los distintos requerimientos y si hay requerimientos opcionales.

!!!No olvidarse de que los requerimientos incluyen a las regulaciones y normas vigentes!!!

Y al escribirlos seguir las siguientes reglas:

- Ser breve y conciso (nadie lee cosas largas).
- Ser específico: no dejar lugar a confusiones.
- Expresar los requerimientos en términos que sean cuantificables y medibles.

7. Historias de usuarios (*Product backlog*)

Descripción: en esta sección se deben incluir las historias de usuarios y su ponderación (*history points*). Recordar que las historias de usuarios son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema. La ponderación es un número entero que representa el tamaño de la historia comparada con otras historias de similar tipo.

Se debe indicar explícitamente el criterio para calcular los *story points* de cada historia.

El formato propuesto es:

1. “Como [rol] quiero [tal cosa] para [tal otra cosa].”
Story points: 8 (complejidad: 3, dificultad: 2, incertidumbre: 3)

8. Entregables principales del proyecto

Los entregables del proyecto son (ejemplo):

- Manual de usuario.
- Diagrama de circuitos esquemáticos.
- Código fuente del firmware.

- Diagrama de instalación.
- Memoria del trabajo final.
- etc...

9. Desglose del trabajo en tareas

El WBS debe tener relación directa o indirecta con los requerimientos. Son todas las actividades que se harán en el proyecto para dar cumplimiento a los requerimientos. Se recomienda mostrar el WBS mediante una lista indexada:

1. Grupo de tareas 1 (suma h)
 - 1.1. Tarea 1 (tantas h)
 - 1.2. Tarea 2 (tantas h)
 - 1.3. Tarea 3 (tantas h)
2. Grupo de tareas 2 (suma h)
 - 2.1. Tarea 1 (tantas h)
 - 2.2. Tarea 2 (tantas h)
 - 2.3. Tarea 3 (tantas h)
3. Grupo de tareas 3 (suma h)
 - 3.1. Tarea 1 (tantas h)
 - 3.2. Tarea 2 (tantas h)
 - 3.3. Tarea 3 (tantas h)
 - 3.4. Tarea 4 (tantas h)
 - 3.5. Tarea 5 (tantas h)

Cantidad total de horas: tantas.

¡Importante!: la unidad de horas es h y va separada por espacio del número. Es incorrecto escribir “23hs”.

Se recomienda que no haya ninguna tarea que lleve más de 40 h. De ser así se recomienda dividirla en tareas de menor duración.

10. Diagrama de Activity On Node

Armar el AoN a partir del WBS definido en la etapa anterior.

Una herramienta simple para desarrollar los diagramas es el Draw.io (<https://app.diagrams.net/>). Draw.io

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semi críticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color.

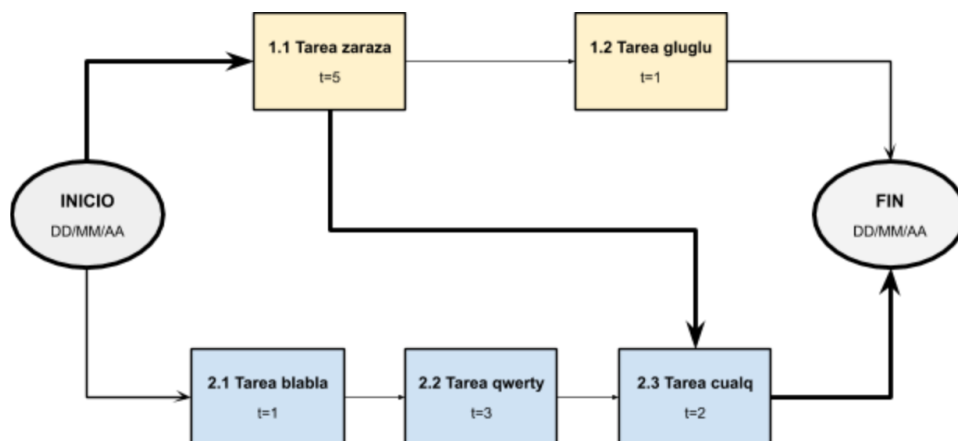


Figura 2. Diagrama de *Activity on Node*.

11. Diagrama de Gantt

Existen muchos programas y recursos *online* para hacer diagramas de Gantt, entre los cuales destacamos:

- Planner
- GanttProject
- Trello + *plugins*. En el siguiente link hay un tutorial oficial:
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately, herramienta online colaborativa.
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>
- Se puede hacer en latex con el paquete *pgfgantt*
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Pegar acá una captura de pantalla del diagrama de Gantt, cuidando que la letra sea suficientemente grande como para ser legible. Si el diagrama queda demasiado ancho, se puede pegar primero la “tabla” del Gantt y luego pegar la parte del diagrama de barras del diagrama de Gantt.

Configurar el software para que en la parte de la tabla muestre los códigos del EDT (WBS).
Configurar el software para que al lado de cada barra muestre el nombre de cada tarea.
Revisar que la fecha de finalización coincida con lo indicado en el Acta Constitutiva.

En la figura 3, se muestra un ejemplo de diagrama de gantt realizado con el paquete de *pgfgantt*. En la plantilla pueden ver el código que lo genera y usarlo de base para construir el propio.

Las fechas pueden ser calculadas utilizando alguna de las herramientas antes citadas. Sin embargo, el siguiente ejemplo fue elaborado utilizando [esta hoja de cálculo](#).

Es importante destacar que el ancho del diagrama estará dado por la longitud del texto utilizado para las tareas (Ejemplo: tarea 1, tarea 2, etcétera) y el valor *x unit*. Para mejorar la apariencia del diagrama, es necesario ajustar este valor y, quizás, acortar los nombres de las tareas.

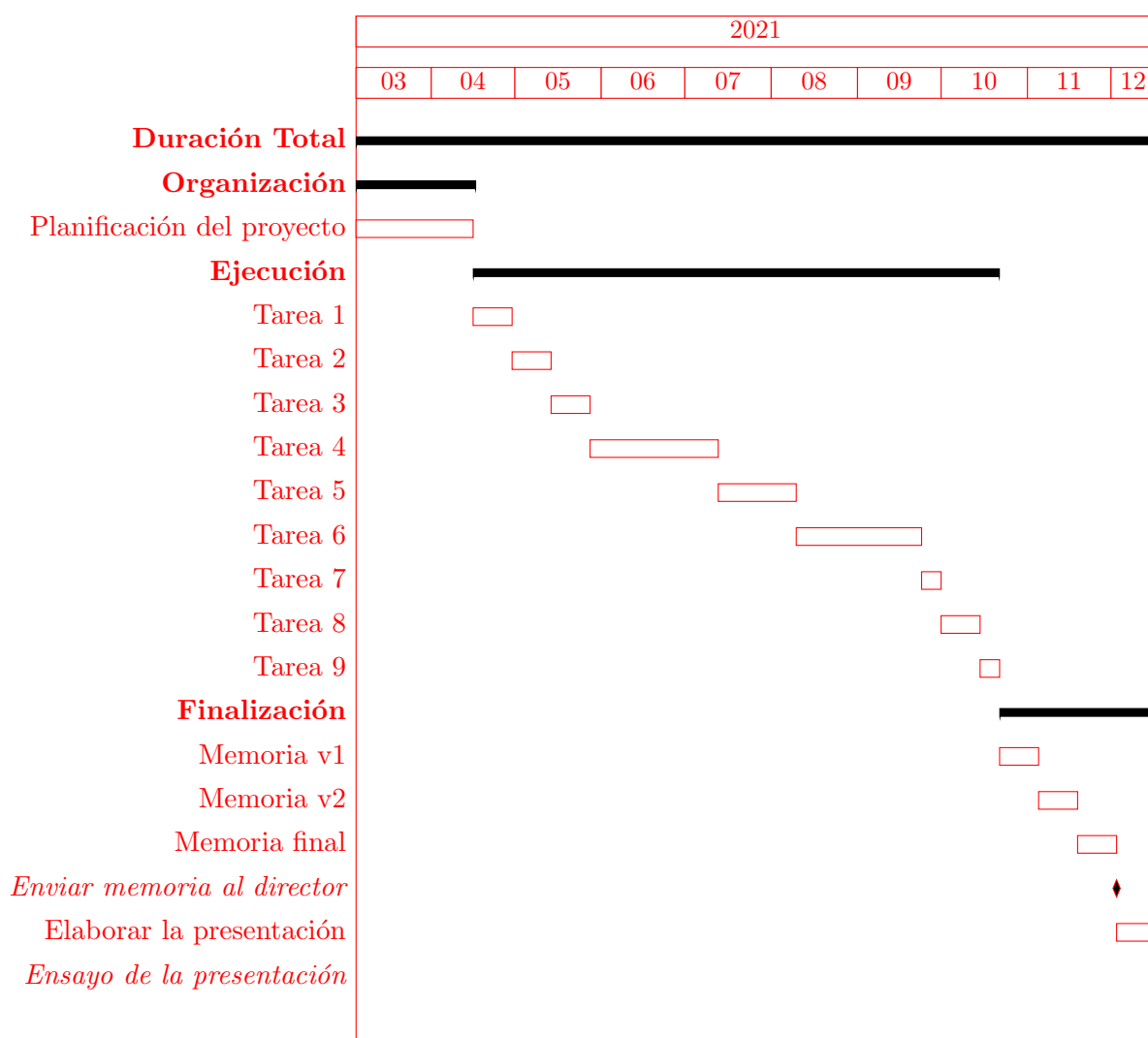


Figura 3. Diagrama de gantt de ejemplo



Figura 4. Ejemplo de diagrama de Gantt (apaisado).

12. Presupuesto detallado del proyecto

Si el proyecto es complejo entonces separarlo en partes:

- Un total global, indicando el subtotal acumulado por cada una de las áreas.
- El desglose detallado del subtotal de cada una de las áreas.

IMPORTANTE: No olvidarse de considerar los **COSTOS INDIRECTOS**.

Incluir la aclaración de si se emplea como moneda el peso argentino (ARS) o si se usa moneda extranjera (USD, EUR, etc). Si es en moneda extranjera se debe indicar la tasa de conversión respecto a la moneda local en una fecha dada.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
SUBTOTAL			
TOTAL			

13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S): X.
Justificación...

- Ocurriencia (O): Y.
Justificación...

Riesgo 3:

- Severidad (S): X.
Justificación...
- Ocurriencia (O): Y.
Justificación...

b) Tabla de gestión de riesgos: (El RPN se calcula como $RPN=S \times O$)

Riesgo	S	O	RPN	S*	O*	RPN*

Criterio adoptado:

Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).
Nueva asignación de S y O, con su respectiva justificación:

- Severidad (S*): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O*): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

14. Gestión de la calidad

Elija al menos diez requerimientos que a su criterio sean los más importantes/críticos/que aportan más valor y para cada uno de ellos indique las acciones de verificación y validación que permitan asegurar su cumplimiento.

- Req #1: copiar acá el requerimiento con su correspondiente número.

- Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar.
- Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar.

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc.

Las acciones de verificación suelen considerar al entregable como “caja blanca”, es decir se conoce en profundidad su funcionamiento interno.

En cambio, las acciones de validación suelen considerar al entregable como “caja negra”, es decir, que no se conocen los detalles de su funcionamiento interno.

15. Procesos de cierre

Establecer las pautas de trabajo para realizar una reunión final de evaluación del proyecto, tal que contemple las siguientes actividades:

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento a aplicar.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, los problemas que surgieron y cómo se solucionaron:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento para dejar registro.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores:
 - Indicar esto y quién financiará los gastos correspondientes.