Desarrollo de un Sistema de Comunicación para Satélites basado en estándares espaciales

Autor:
Ing. Gonzalo Lavigna

Director:
Ing. Roberto Cibils (FIUBA)

Esta planificación fue realizada en el curso de Gestión de proyectos entre el 4 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 (*Product backlog*) **10**

8. Entregables principales del proyecto **11**

9. Desglose del trabajo en tareas **12**

10. Diagrama de Activity On Node **13**

11. Diagrama de Gantt **14**

12. Presupuesto detallado del proyecto **17**

13. Gestión de riesgos **17**

14. Gestión de la calidad **18**

15. Procesos de cierre **19**

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	4 de marzo de 2025
1	Se completa hasta el punto 5 inclusive	16 de marzo de 2025
2	Se completa hasta el punto 9 inclusive	27 de marzo de 2025

Acta de constitución del proyecto

Buenos Aires, 4 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 para Satélites basado en estándares espaciales" 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 dólares estadounidenses (U$D 1000), con fecha de inicio el 4 de marzo de 2025 y fecha de presentación pública en 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 fisica 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, consiste 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.

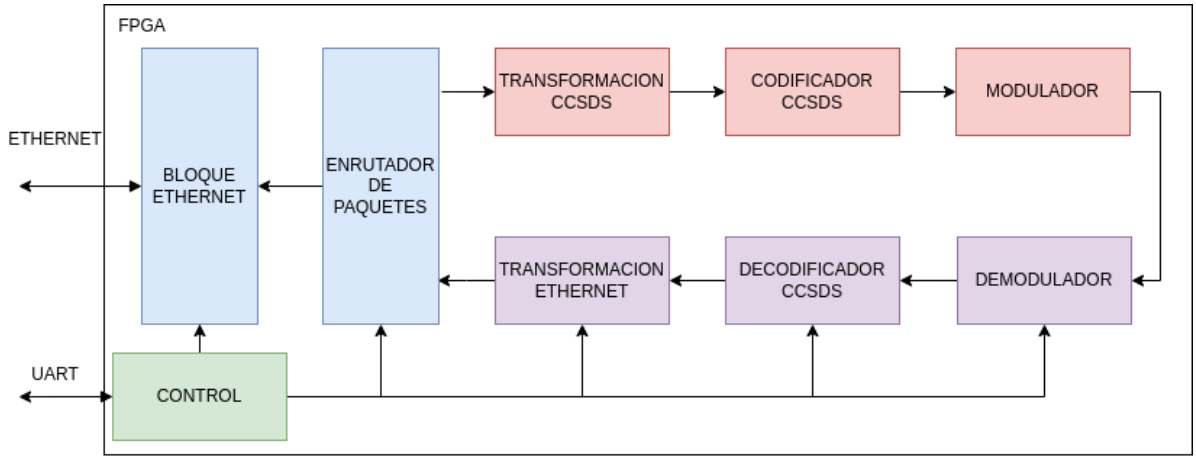

Figura 1. Diagrama en bloques del sistema.

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. 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

1. Requerimientos funcionales:
   1.1. El sistema debe tener una baja latencia en el procesamiento y transferencia de paquetes (menor a 15 ms).
   1.2. El usuario debe ser capaz de configurar todos los bloques del sistema a través de comando externos utilizando UART.
   1.3. El bloque Ethernet debe soportar transmisión y recepción de paquetes estándar de Ethernet (IEEE 802.3).
   1.4. El bloque Ethernet debe soportar una velocidad de comunicación de 100 Mbps.
   1.5. El bloque Ethernet debe tener la capacidad de manejar en simultáneo múltiples paquetes entrantes y salientes.

1.6. El bloque Ethernet debe tener la capacidad de generar estadísticas básicas de red: paquetes transmitidos, paquetes recibidos, errores detectados.

1.7. El bloque Ethernet debe tener la capacidad de identificar y gestionar errores básicos en paquetes Ethernet.

1.8. El bloque Ethernet debe brindar soporte para tramas Ethernet estándar de hasta 1500 bytes (*Maximum Transmission Unit* estándar).

1.9. El bloque enrutador debe ser capaz de recibir, procesar y enrutar paquetes Ethernet internamente entre los diferente módulos del sistema.

1.10. El bloque enrutador debe tener la capacidad de almacenamiento temporal de unos 4 paquetes para evitar la pérdida de datos en situaciones de congestión temporal.

1.11. El bloque enrutador deberá contar con un mecanismo interno para realizar una conexión logica en modo *loopback*, permitiendo que los paquetes Ethernet recibidos puedan ser automáticamente redirigidos a la cadena de transmisión de datos Ethernet. Este mecanismo facilitará pruebas y validaciones funcionales.

1.12. El sistema debe encapsular correctamente los paquetes Ethernet recibidos en tramas AOS según especificaciones del estándar CCSDS 732.0-B.

1.13. El sistema debe ser capaz de codificar y decodificar tramas CCSDS completas agregando códigos Reed-Solomon para detección y corrección de errores, según la configuración estándar (255,239).

1.14. El bloque modulador debe implementar un esquema de modulación digital compatible con los estándares utilizados en sistemas espaciales CCSDS: *Binary Phase Shift Keying* (BPSK) y *Quadrature Phase Shift Keying* (QPSK) según lo especificado en el estándar CCSDS 401.0-B.

2. Requerimientos de documentación:

2.1. Toda la documentación técnica del proyecto, incluyendo diagramas, especificaciones, manuales de usuario, y procedimientos de validación y configuración, deberán estar claramente redactados en español, utilizando un lenguaje formal y técnico.

2.2. La totalidad del código fuente generado, junto con su documentación asociada, deberá publicarse en repositorios públicos de acceso general garantizando así su disponibilidad abierta para cualquier usuario interesado.

2.3. Deberá incluirse una guía breve («README») en los repositorios públicos, que describa claramente el objetivo, características principales y licencia de uso.

2.4. Todo el código fuente generado en este proyecto, junto con su documentación asociada, deberá publicarse bajo la licencia de código abierto BSD de 3 cláusulas. Esta licencia permite explícitamente el uso libre, modificación, distribución y comercialización del software y derivados.

3. Requerimientos de verificación y validación:

3.1. Se deberán desarrollar bancos de pruebas específicos para cada módulo interno del sistema, utilizando lenguajes estándares como Python, VHDL o SystemVerilog, que permitan simular y verificar su correcto funcionamiento individual.

3.2. Los bloques Transformación CCSDS y Transformación Ethernet se verificarán en conjunto conectando los mismos en *loopback* en un banco de pruebas realizado para dicho propósito.

3.3. Los bloques Codificador CCSDS y Decodificador CCSDS se verificarán en conjunto conectando los mismos en *loopback* en un banco de pruebas realizado para dicho propósito.

3.4. Los bloques Modulador y Demodulador se verificarán en conjunto conectando los mismos en *loopback* en un banco de pruebas realizado para dicho propósito.

3.5. Todos las simulaciones funcionales deben realizar con herramientas reconocidas como Vivado Simulator, ModelSim o herramientas equivalentes.

3.6. La verificación del sistema deberá incluir simulaciones de integración, donde se validará la interoperabilidad de los módulos interconectados mediante interfaces AXI-Stream y AXI-Lite.

3.7. La validación del sistema deberá realizarse haciendo pruebas funcionales en la placa de desarrollo en modo *loopback* lógico, verificando el cumplimiento de los requisitos del estándar CCSDS y la correcta operación del flujo completo desde generación hasta recepción de datos.

3.8. Se debe incluir una simulación con la inserción controlada de errores (por ejemplo, ruido o corrupción artificial de paquetes) para verificar la robustez y la capacidad de corrección de errores del sistema implementado.

4. Requerimientos de interfaz:

4.1. El sistema deberá contar con una interfaz UART para comunicación serie externa, que permita monitorear el estado operativo y enviar comandos básicos para configurar y controlar la operación del sistema.

4.2. La interfaz UART deberá funcionar bajo una configuración estándar de comunicación (8 bits de datos, sin paridad y 1 bit de parada), soportando tasas configurables entre 9600 y 115200 baudios.

4.3. Deberá incluirse una interfaz Ethernet física compatible con el estándar IEEE 802.3, para la transmisión y recepción externa de paquetes Ethernet estándar (mínimo 100 Mbps).

4.4. Todos los módulos internos del sistema deberán utilizar el protocolo *AXI-Stream* como estándar para la comunicación e intercambios de paquetes, asegurando así la compatibilidad, simplicidad en la integración y estandarización del flujo interno del sistema.

4.5. Todos los módulos internos del sistema deberán contar con una interfaz de control y monitoreo basada en el protocolo *AXI-Lite*. Esta interfaz permitirá realizar operaciones básicas de lectura y escritura en registros internos para configurar parámetros específicos de cada módulo y obtener información sobre el estado operativo en tiempo real.

7. Historias de usuarios (*Product backlog*)

Se utilizó la asignación de *story points* para evaluar las historias de usuario. El valor se determinó considerando los siguientes factores: tiempo, complejidad e incertidumbre asociada con la tarea.

- Tiempo.
- Complejidad.
- Riesgo.

Para las estimaciones mediante *story points* se utilizará como escala números de la serie de Fibonacci comprendidos en el rango [1, 8]. A cada criterio se le asigna un número dentro de esta escala. Luego, se sumarán los números de cada criterio para obtener una calificación final. Si este número no pertenece a un valor de la serie se aproxima al más cercano. Un número más grande indica mayor complejidad, riesgo y/o tiempo.

- Historia 1: Como operador del sistema quiero monitorear el estado operativo para verificar rápidamente que el sistema funciona correctamente. Tiempo: 2 - Complejidad: 2 - Riesgo: 1 - *story points*: 5.
- Historia 2: Como ingeniero de integración, quiero una interfaz Ethernet estándar (IEEE 802.3) para enviar y recibir paquetes desde sistemas externos en pruebas y validaciones. Tiempo: 3 - Complejidad: 3 - Riesgo: 2 - *story points*: 8.
- Historia 3: Como desarrollador de FPGA, quiero que todos los módulos internos se comuniquen mediante AXI-Stream para asegurar la interoperabilidad y escalabilidad del diseño. Tiempo: 2 - Complejidad: 2 - Riesgo: 1 - *story points*: 5.
- Historia 4: Como desarrollador FPGA, quiero utilizar bloques validados de codificación y decodificación Reed-Solomon provistos por terceros para acelerar el desarrollo y reducir riesgos técnicos. Tiempo: 2 - Complejidad: 1 - Riesgo: 8 - *story points*: 13.
- Historia 5: Como ingeniero de integración, quiero realizar simulaciones en modo *loopback* lógico para validar el funcionamiento completo del protocolo CCSDS sin *hardware* externo adicional. Tiempo: 2 - Complejidad: 1 - Riesgo: 8 - *story points*: 13.
- Historia 6: Como usuario final , quiero acceder al código fuente y documentación completa del sistema en repositorios públicos para adaptarlo y reutilizarlo en futuras misiones satelitales. Tiempo: 1 - Complejidad: 1 - Riesgo: 1 - *story points*: 3.
- Historia 7: Como desarrollador de FPGA, quiero configurar fácilmente los módulos internos mediante AXI-Lite para adaptar rápidamente los parámetros operativos del sistema en diferentes escenarios de prueba. Tiempo: 1 - Complejidad: 3 - Riesgo: 1 - *story points*: 5.
- Historia 8: Como desarrollador de FPGA, quiero contar con bancos de prueba automatizados para realizar simulaciones rápidas y repetibles de los módulos individuales del sistema.Tiempo: 5 - Complejidad: 3 - Riesgo: 5 - *story points*: 13.

8. Entregables principales del proyecto

Los entregables del proyecto son:

- Informe de avance del proyecto.
- Código fuente completo del sistema implementado (con licencia BSD-3).
- Bancos de pruebas utilizados para simulación y validación.
- Documentación técnica del diseño, diagramas y especificaciones técnicas.
- Manual de uso y guía rápida para configuración y operación del prototipo desarrollado.

- Resultados y documentación de las pruebas realizadas en placa, con el prototipo del sistema implementado y funcionando en una placa de desarrollo de FPGA.
- Informe final, incluyendo resultados obtenidos, análisis del desempeño, conclusiones y recomendaciones.

9. Desglose del trabajo en tareas

1. Gestión y documentación del proyecto (100 h)
   1.1. Definición inicial del alcance del proyecto (8 h).
   1.2. Revisión y aprobación del alcance definitivo (8 h).
   1.3. Informes preliminares de avance del proyecto (8 h).
   1.4. Elaboración de documentación técnica general, diagrama en bloque y documentación de la arquitectura (10 h).
   1.5. Escribir memoria técnica (40 h).
   1.6. Escribir y preparar presentación final (10 h).
   1.7. Crear documento de lecciones aprendidas (8 h).
   1.8. Publicación de documentación en repositorios públicos (8 h).
2. Diseño conceptual y arquitectura (60 h)
   2.1. Investigación inicial de los diversos estándares CCSDS (20 h).
   2.2. Análisis de requerimientos técnicos CCSDS a implementar en la FPGA (20 h).
   2.3. Diseño conceptual del sistema (4 h).
   2.4. Diseño detallado del sistema y diagramas finales (16 h).
3. Configuración del entorno de desarrollo (40 h)
   3.1. Instalación y configuración inicial del software Vivado (4 h).
   3.2. Instalación y configuración del simulador *ModelSim* o equivalente.(4 h).
   3.3. Preparación de proyecto inicial (estructura del proyecto, repositorios Git) (4 h).
   3.4. Configuración del entorno para integración continua y gestión de código fuente (28 h).
4. Desarrollo de las interfaces del sistema (60 h)
   4.1. Diseño del módulo interfaz UART (10 h).
   4.2. Implementación y validación módulo UART (10 h).
   4.3. Diseño del módulo interfaz Ethernet (20 h).
   4.4. Implementación y validación módulo Ethernet (20 h).
5. Implementación del protocolo AXI interno (40 h)
   5.1. Diseño e implementación interfaces AXI-Stream (20 h).
   5.2. Diseño e implementación interfaces AXI-Lite (20 h).
6. Integración bloques Reed-Solomon (120 h)

6.1. Integración bloque codificador Reed-Solomon (20 h).
6.2. Validación y ajustes codificador Reed-Solomon (20 h).
6.3. Validación cruzada del módulo codificador Reed-Solomon contra modelo Python (20 h).
6.4. Integración bloque decodificador Reed-Solomon (20 h).
6.5. Validación y ajustes decodificador Reed-Solomon (20 h).
6.6. Validación cruzada del módulo decodificador Reed-Solomon contra modelo Python (20 h).

7. Implementación módulos conversión Ethernet/AOS (60 h)
   7.1. Diseño e implementación módulo Ethernet a AOS (20 h).
   7.2. Pruebas módulo Ethernet a AOS (10 h).
   7.3. Diseño módulo AOS a Ethernet (20 h).
   7.4. Pruebas módulo AOS a Ethernet (10 h).
8. Implementación bloques modulación/demodulación (60 h)
   8.1. Diseño e implementación del módulo modulador CCSDS (10 h).
   8.2. Pruebas módulo modulador CCSDS (10 h).
   8.3. Diseño e implementación del módulo demodulador CCSDS (20 h).
   8.4. Pruebas módulo demodulador CCSDS (20 h).
9. Validación y pruebas (100 h)
   9.1. Diseño y desarrollo de bancos de prueba en *loopback* (20 h).
   9.2. Ejecución y validación de bancos de prueba en *loopback* (20 h).
   9.3. Síntesis e implementación del sistema completo en placa FPGA (40 h).
   9.4. Ejecución de pruebas de validación sobre el prototipo implementado (20 h).

Cantidad total de horas: 640.

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.

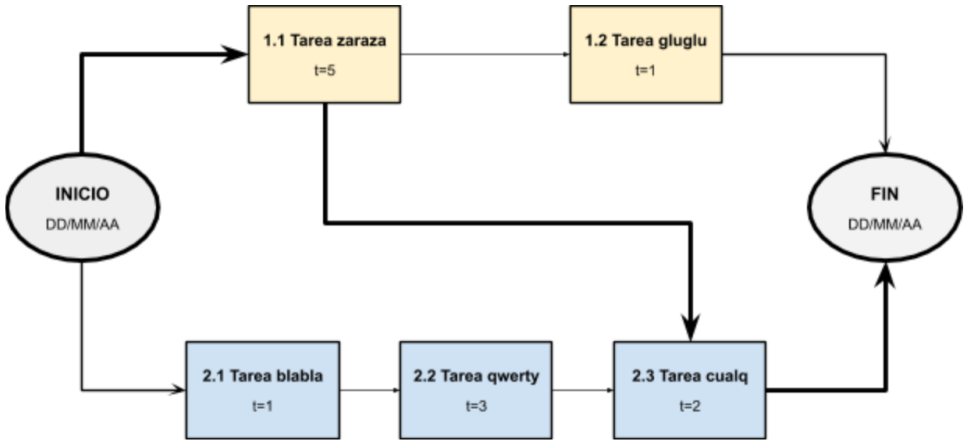

Figura 2. Diagrama de *Activity on Node*.

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.

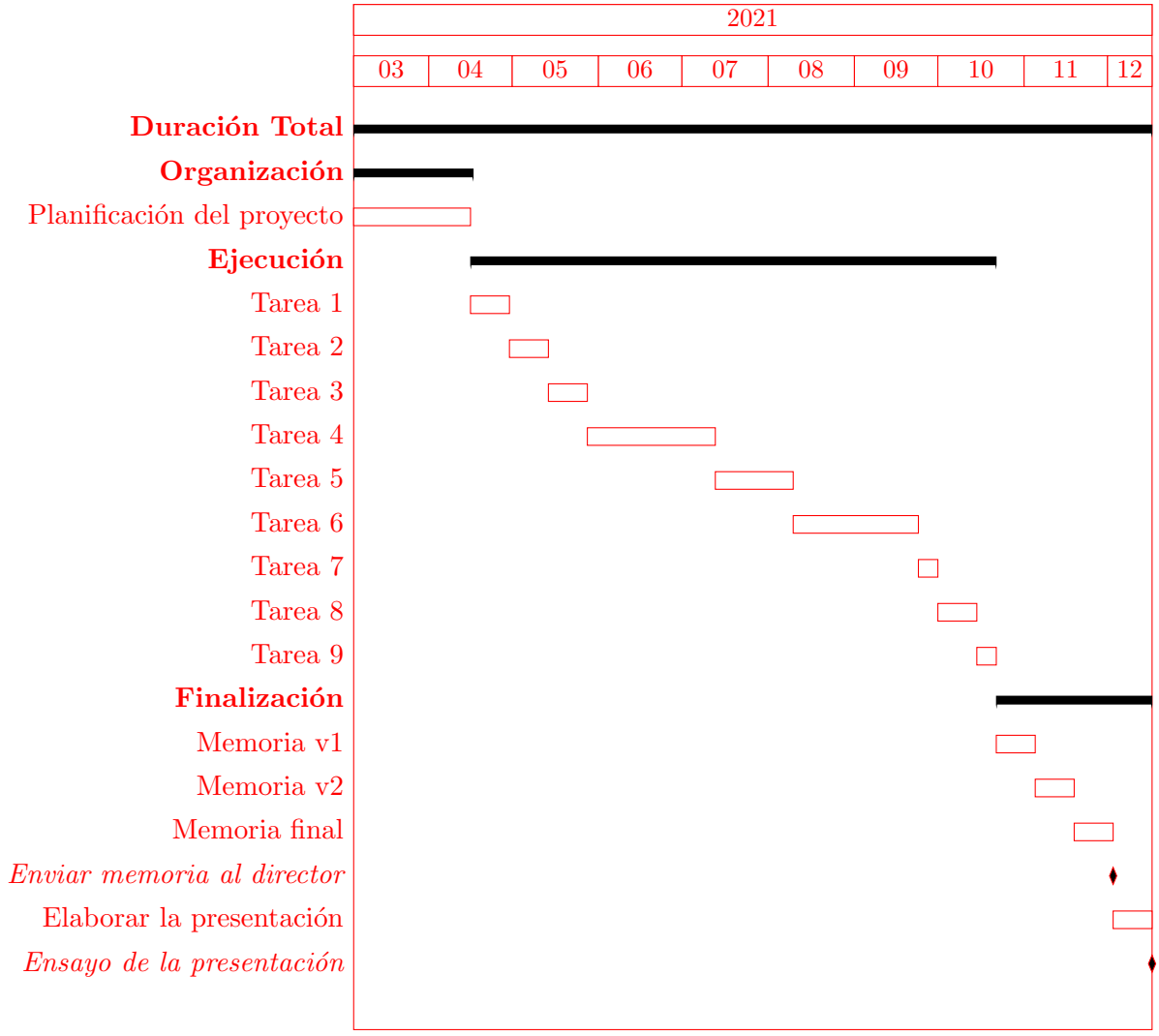

Figura 3. Diagrama de gantt de ejemplo

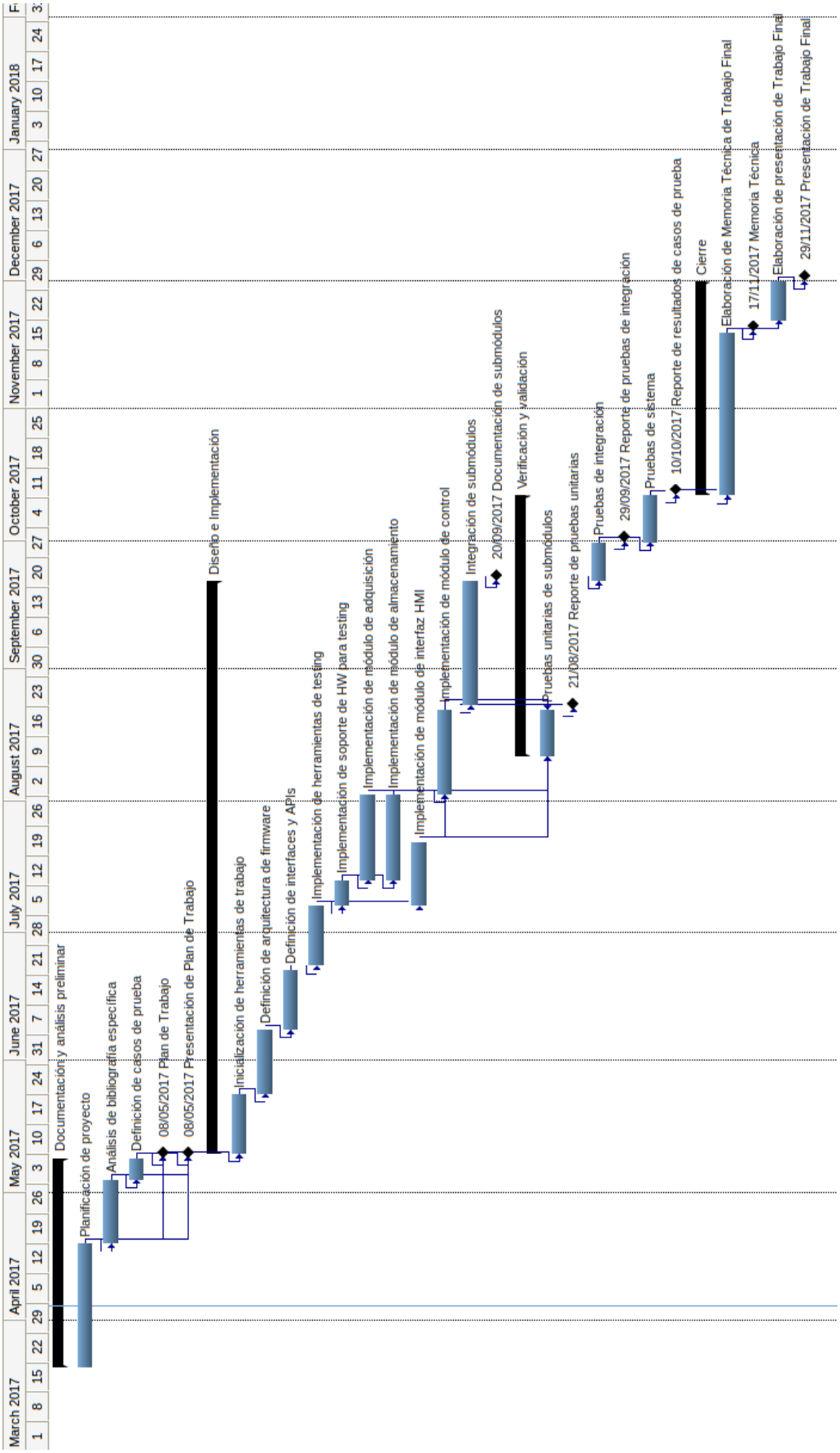

Figura 4. Ejemplo de diagrama de Gantt (apaisado).

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...

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

Riesgo 3:

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

b) Tabla de gestión de riesgos: (El RPN se calcula como RPN=SxO)

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.