



NAVEGACIÓN, GUIADO Y CONTROL DE VEHÍCULOS AEROSPACIALES

Memoria del Proyecto Final de Carrera
de Ingeniería Informática
realizado por Alberto Pérez Marcos
y dirigido por Porfidio Hernández Budé
Bellaterra, 24 de Enero de 2008.



Escola Tècnica Superior d'Enginyeria

El abajo firmante, Porfidio Hernández Budé
Profesor de la Escuela Técnica Superior de Ingeniería de la UAB,

CERTIFICA:

Que el trabajo al que corresponde esta memoria ha sido realizado bajo su dirección por Alberto Pérez Marcos

I para que conste firma la presente.

Firmado: Porfidio Hernández Budé
Bellaterra, 24 de Enero de 2008

DIFUSIÓN	COPIAS	MEDIO
Tribunal ETSE – UAB	3	Impreso
Presidente del Tribunal – ETSE UAB	1	Electrónico

RESUMEN DE MODIFICACIONES			
Ed.	Fecha	Modificación	Autor
1.0	01/01/2008	Creación del documento	Alberto Pérez

- *Toda modificación en el documento que origine una evolución en la edición o en la revisión debe ser trazada en la tabla Resumen de Modificaciones.*

Índice

1	Introducción	6
1.1	Objetivos	6
1.2	Estructura de la memoria	7
1.3	Convenciones	8
1.4	Documentación de referencia	8
1.5	Definiciones, Abreviaturas y Acrónimos	8
2	Plan de Proyecto.....	10
2.1	Organización general	10
2.2	Ciclo de vida.....	11
2.3	Paquetes de trabajo	12
2.4	Análisis de riesgos.....	15
2.5	Planificación	15
3	Análisis	17
3.1	Análisis Científico	17
3.1.1	Descripción general del vehículo	17
3.1.2	Descripción de la misión	18
3.1.3	Sistemas de referencia	25
3.2	Análisis del sistema informático.....	27
3.2.1	Arquitectura del sistema	28
3.2.2	Protocolos de comunicación.....	35
3.3	Análisis Económico	53
4	Especificación.....	56
4.1	Casos de uso	56
4.1.1	Caso de uso: [CU_001] Iniciar Sistema.....	56
4.1.2	Caso de uso: [CU_010] Editar Misión.....	57
4.1.3	Caso de uso: [CU_020] Cargar Misión.....	58
4.1.4	Caso de uso: [CU_030] Ejecutar Misión	59
4.1.5	Caso de uso: [CU_040] Cancelar Misión.....	61
4.1.6	Caso de uso: [CU_050] Abortar Misión	61
4.1.7	Caso de uso: [CU_060] Guardar Histórico	62

4.2	Especificación de requisitos.....	62
4.2.1	Requisitos funcionales	64
4.2.2	Requisitos no funcionales.....	67
5	Diseño	72
5.1	Arquitectura software	72
5.2	Diseño preliminar	75
5.2.1	Centro de Control de Misión en Tierra	76
5.2.2	OnBoard Computer	85
5.2.3	Simulador Vehiculo Aeroespacial.....	87
5.3	Diseño detallado	87
6	Conclusiones.....	88
6.1	Análisis del desarrollo del proyecto	88
6.1.1	Plan de Proyecto.....	88
6.1.2	Análisis	88
6.1.3	Especificación	89
6.1.4	Diseño	90
6.2	Conclusión	90
6.3	Líneas Abiertas	91

Índice de Figuras

Figura 1:	Organigrama del proyecto	11
Figura 2:	Ciclo de vida	11
Figura 3:	Work Breakdown Structure.....	12
Figura 4:	Diagrama de Gant.....	16
Figura 5:	Vehículo aeroespacial	18
Figura 6:	Esquema de misión	20
Figura 7:	Secuencia de Eventos.....	24
Figura 8:	Sistemas de referencia	26
Figura 9:	Orientación Ejes Cuerpo.....	27
Figura 10:	Arquitectura del sistema	28
Figura 11:	Funcionamiento buffers.....	30
Figura 12:	Interfaces de comunicación	35

Figura 13: Diagrama de estados INS.....	37
Figura 14: Motores de gas frío.....	39
Figura 15: Intercambio de tramas CCMT-OBC.....	48
Figura 16: Intercambio de tramas OBC-SimRT-VA.....	52
Figura 17: Caso de uso: [CU_001] Iniciar Sistema	57
Figura 18: Caso de uso: [CU_010] Editar Misión	58
Figura 19: Caso de uso: [CU_020] Cargar Misión	59
Figura 20: Caso de uso: [CU_030] Ejecutar Misión	60
Figura 21: Caso de uso: [CU_040] Cancelar Misión	61
Figura 22: Caso de uso: [CU_050] Abortar Misión	61
Figura 23: Caso de uso: [CU_060] Guardar Histórico.....	62
Figura 24: Diagrama de componentes	73
Figura 25: Diagrama de despliegue.....	75
Figura 26: Maqueta IHM.....	77
Figura 27: Diagrama de clases IHM.....	78
Figura 28: Diagrama de clases CMT	80
Figura 29: Diagrama de clases MD.....	82
Figura 30: Modelo BD	83
Figura 31: Diagrama de clases T&T.....	84
Figura 32: Diagrama de clase CM	85
Figura 33: Diagrama de clase NGC.....	86
Figura 34: Diagrama de módulos SimRT-VA.....	87

Índice de Tablas

Tabla 1: Documentos de referencia.....	8
Tabla 2: Definiciones, Abreviaturas y Acrónimos	9
Tabla 3: Fichero de trayectoria en rampa.....	31
Tabla 4: Fichero de trayectoria en vuelo	31
Tabla 5: Formato de la cabecera de los ficheros de histórico.....	32
Tabla 6: Formato fichero de histórico Vector de Estado	32
Tabla 7: Formato fichero de histórico Eventos	32
Tabla 8: Modos de operación INS	37

Tabla 9: Comandos	37
Tabla 10: Fases de vuelos	38
Tabla 11: Activación motores de gas frío	39
Tabla 12: Información líneas digitales	40
Tabla 13: Trama Tele-Destrucción	41
Tabla 14: Trama Inicialización	42
Tabla 15: Trama Punto de Trayectoria en Rampa	42
Tabla 16: Trama Punto de Trayectoria Fuera de Rampa	43
Tabla 17: Trama Alive.....	43
Tabla 18: Trama Abortar Misión.....	44
Tabla 19: Trama Confirmación Tele-Destrucción	44
Tabla 20: Trama Confirmación Inicialización	44
Tabla 21: Trama Confirmación Trayectoria En Rampa	45
Tabla 22: Trama Confirmación Trayectoria Fuera de Rampa	45
Tabla 23: Trama Confirmación Alive	46
Tabla 24: Trama Confirmación Abortar Misión.....	46
Tabla 25: Trama Vector de Estado	47
Tabla 26: Trama Inicialización	49
Tabla 27: Trama Comandos	50
Tabla 28: Trama Señal de Control.....	50
Tabla 29: Trama Datos	51
Tabla 30: Trama protocolo HDLC	53
Tabla 31: Valoración coste RRHH.....	54
Tabla 32: Valoración coste licencias SW.....	55
Tabla 33: Valoración coste material HW.....	55
Tabla 34: Especificación de un requisito.....	64
Tabla 35: Descripción clases IHM.....	79
Tabla 36: Descripción clases CMT	80
Tabla 37: Descripción clases MD.....	81
Tabla 38: Descripción modelo BD	82
Tabla 39: Descripción clases T&T.....	84
Tabla 40: Descripción clases CM.....	85
Tabla 41: Descripción clases NGC	86

1 Introducción

Sistemas más complejos, menores ciclos de desarrollo y una considerable reducción en los presupuestos penalizan la entrada de nuevas empresas en un área como es la de integración de sistemas de navegación, guiado y control (NGC) para vehículos aeroespaciales.

Al mismo tiempo, y tras la aprobación de las tecnologías a emplear en Galileo, parece imparable la incorporación de nuevas herramientas de desarrollo que, en algunos casos, además de requerir un esfuerzo de formación importante, arrojan dudas sobre su capacidad para cubrir algunos requisitos de los sistemas NGC.

Igualmente, tal y como se ha venido constatando en todas aquellas áreas de actividad relacionadas con sistemas críticos, entendiéndose por tales, aquellos sistemas en los que un mal funcionamiento asociado a errores de diseño y/o implementación puede originar pérdidas económicas importantes y/o graves perjuicios para la integridad física de las personas, es necesario disponer de organismos independientes que garanticen, mediante la apropiada certificación, la adecuación del sistema a la función para la que se destina.

1.1 OBJETIVOS

El objetivo del proyecto propuesto es doble: por un lado se pretende adquirir experiencia en el desarrollo de software de navegación, guiado y control de vehículos aeroespaciales, mientras que, paralelamente, se pretende diseñar una plataforma de ensayos NGC.

La plataforma de ensayos constará de 3 subsistemas principales, a saber:

- **SimRT-VA:** Simulador Tiempo Real del Vehículo Aeroespacial.
- **OBC:** OnBoard Computer.
- **CCMT:** Centro de Control de Misión en Tierra.

El subsistema SimRT-VA es un simulador tiempo real de 6 grados de libertad que reproducirá la dinámica del vehículo real. El subsistema SimRT-VA integrará los módulos software y hardware necesarios para notificar los eventos producidos durante el vuelo y recibir los comandos del OBC.

El subsistema OBC integrará el sistema de control y guiado del vehículo, así como, los módulos de telemetría necesarios para establecer un enlace de comunicación bidireccional con el CCMT.

El subsistema CCMT permitirá configurar la misión y ejecutarla, realizar el seguimiento de la misma y registrar los eventos generados durante el vuelo del vehículo para su posterior análisis.

El desarrollo de los diferentes subsistemas permitirá poner en práctica nuevas herramientas no utilizadas en la actualidad en sistemas críticos, por ejemplo, se experimentará con la versión de *JavaRT* y se implementará un controlador digital basado en lógica difusa (FLC).

La plataforma sobre la que se ejecutará el subsistema CCMT será *Linux*, mientras que para los subsistemas OBC y SimRT-VA se valorará la posibilidad de utilizar un sistema operativo tiempo real como por ejemplo *LinuxRT*, *QNX* ó *VxWorks*.

1.2 ESTRUCTURA DE LA MEMORIA

La memoria se estructura entorno a 6 capítulos.

- **Capítulo 1. Introducción.**

El capítulo 1 introduce el contexto del proyecto y presenta los objetivos del mismo. Además, define las convenciones que aplican a la memoria, detalla los documentos de referencia y las definiciones, abreviaturas y acrónimos que aparecen a lo largo de la memoria.

- **Capítulo 2. Plan de Proyecto.**

El capítulo 2 presenta el plan de proyecto. En él se describe la organización general del proyecto, el ciclo de vida, los paquetes de trabajo, el análisis de riesgos y la planificación de las tareas a desarrollar.

- **Capítulo 3. Análisis.**

El capítulo 3 presenta el análisis del sistema desde diferentes puntos de vista. En primer lugar, se describe el análisis científico, el cuál nos permitirá comprender el problema y su contexto. A continuación, se presenta el análisis del sistema informático y la valoración económica.

- **Capítulo 4. Especificación.**

El objeto del capítulo 4 es presentar la especificación técnica del sistema. El capítulo se ha dividido en dos apartados, el primero presenta los casos de uso del sistema y el segundo la especificación de requisitos.

- **Capítulo 5. Diseño.**

El capítulo 5 presenta el diseño del software en tres etapas, la arquitectura del software, el diseño preliminar y el diseño detallado.

- **Capítulo 6. Conclusiones.**

El capítulo 6 está dedicado a las conclusiones extraídas fruto del trabajo realizado y ofrece una guía para futuras implementaciones ó posibles mejoras del sistema desarrollado.

1.3 CONVENCIONES

Para la redacción del presente documento se han establecido un conjunto de convenciones tipográficas, a saber:

- Los anglicismos, nombre técnicos relativos a productos comerciales y nombres de empresas citadas a lo largo del documento se escribirán en *itálicas*.
- La bibliografía referenciada en el documento sigue el formato Harvard APA.

1.4 DOCUMENTACIÓN DE REFERENCIA

Id.	Título	Edición	Fecha
[DR01]	Introducción a la dinámica espacial.	-	1991
[DR02]	An Introduction to Fuzzy Control.	2 nd .	1993
[DR03]	Spacecraft Systems Engineering.	3 rd .	2003
[DR04]	Spacecraft Navigation and Guidance.	-	1998
[DR05]	Discrete-time control systems.	2 nd .	1995

Tabla 1: Documentos de referencia

1.5 DEFINICIONES, ABREVIATURAS Y ACRÓNIMOS

ACK	Acknowledgement
APA	American Psychological Association
API	Application Program Interface
ISA	Internacional Standard Atmosphere
I+D+I	Investigación + Desarrollo + Innovación
CCMT	Centro de Control de Misión en Tierra
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CRC	Control de Redundancia Cíclica
FLC	Fuzzy Logic Controller
GPS	Global Positioning System
HDLC	High-Level Data Link Control
HLR	High Level Requirement
HWIL	HardWare In the Loop
INS	Inertial Navigation System
JNI	Java Native Interface
LLR	Low Level Requirement
MSB	Most Significant Byte
MVC	Model View Controller
NA	No aplica

NRC	National Research Council
NRZ	Non Return to Zero
OBC	OnBoard Computer
OSI	Open System Interconnection
PDF	Portable Document Format
RCS	Range Coordinate System
RT	Real Time
RTSJ	Real Time Specification for Java
SimRT-VA	Simulador Tiempo Real – Vehículo Aeroespacial.
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language
UTC	Universal Time Coordinated
UTM	Universal Transverse Mercator
WBS	Work Breakdown Structure
WP	Work Package
WPT	Way Point

Tabla 2: Definiciones, Abreviaturas y Acrónimos

2 Plan de Proyecto

El objeto de este capítulo es presentar la información necesaria para conocer el contexto en el que se va a desarrollar el proyecto.

El Plan de Proyecto presentado a continuación define todos los aspectos de organización y gestión del proyecto relacionados con el desarrollo de la plataforma de ensayos para vehículos aeroespaciales.

2.1 ORGANIZACIÓN GENERAL

La Figura 1 muestra el organigrama del proyecto previsto para el desarrollo de la plataforma de ensayos para vehículos aeroespaciales.

El rol de cada recurso introducido en el organigrama del proyecto es:

- Responsable de Proyecto:
 - Representa el proyecto en todos los aspectos técnicos y administrativos.
 - Responsable de la ejecución de las actividades definidas en el proyecto.
 - Interlocución con el cliente.
- Responsable de Calidad:
 - Aseguramiento de la calidad del producto.
 - Adaptación de los estándares de calidad de la empresa a los requeridos por el cliente.
- Responsable Técnico:
 - Responsable de la gestión técnica del proyecto.
 - Asignación de responsabilidades técnicas dentro del equipo de proyecto.
 - Interlocución técnica con el cliente.
 - Coordinación y supervisión de las actividades desde el punto de vista técnico.
- Analista Funcional:
 - Análisis y especificación técnica.
- Ingeniero de Proyecto:

- Responsable del diseño, codificación, test y validación del software.
- Responsable de los sistemas informáticos.
- Redacción de la documentación técnica del proyecto.
- Ingeniero de Control:
 - Responsable del diseño del sistema de control del vehículo.

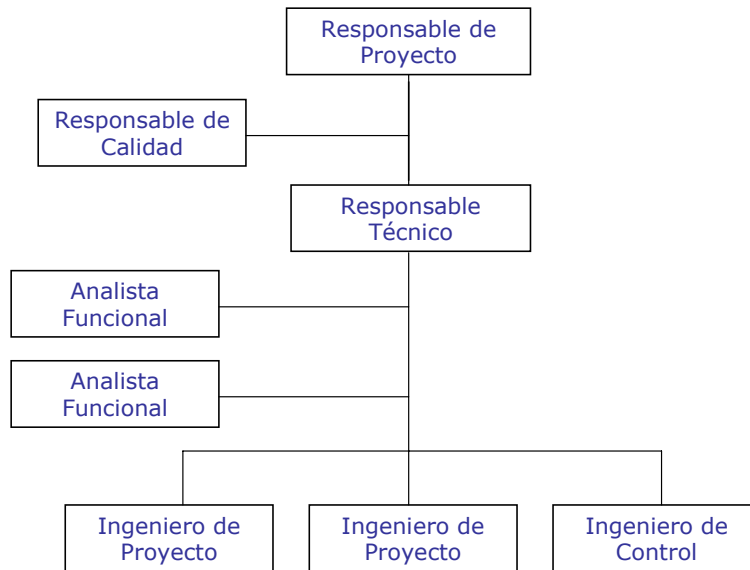


Figura 1: Organigrama del proyecto

2.2 CICLO DE VIDA

El ciclo de vida adoptado para desarrollar el proyecto es un ciclo de vida en V mostrado en la Figura 2.

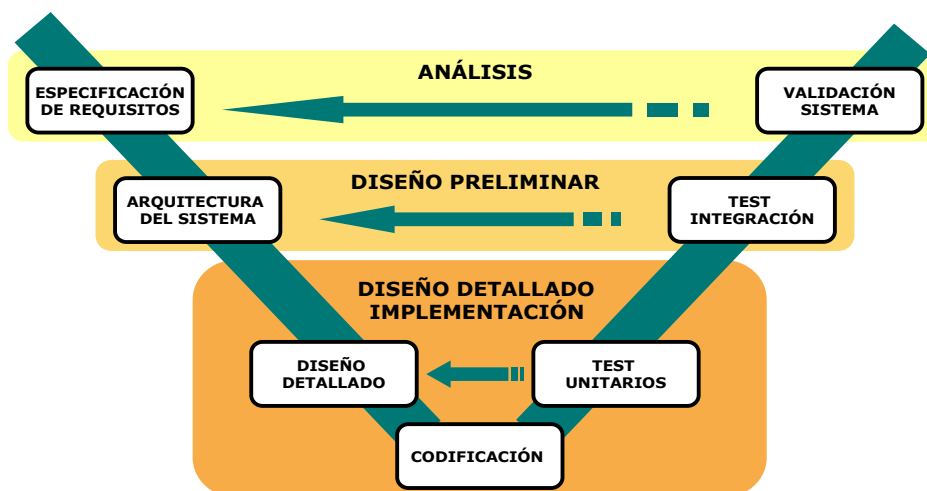


Figura 2: Ciclo de vida

2.3 PAQUETES DE TRABAJO

La descomposición del proyecto en paquetes de trabajo (WBS) se presenta en la Figura 3.

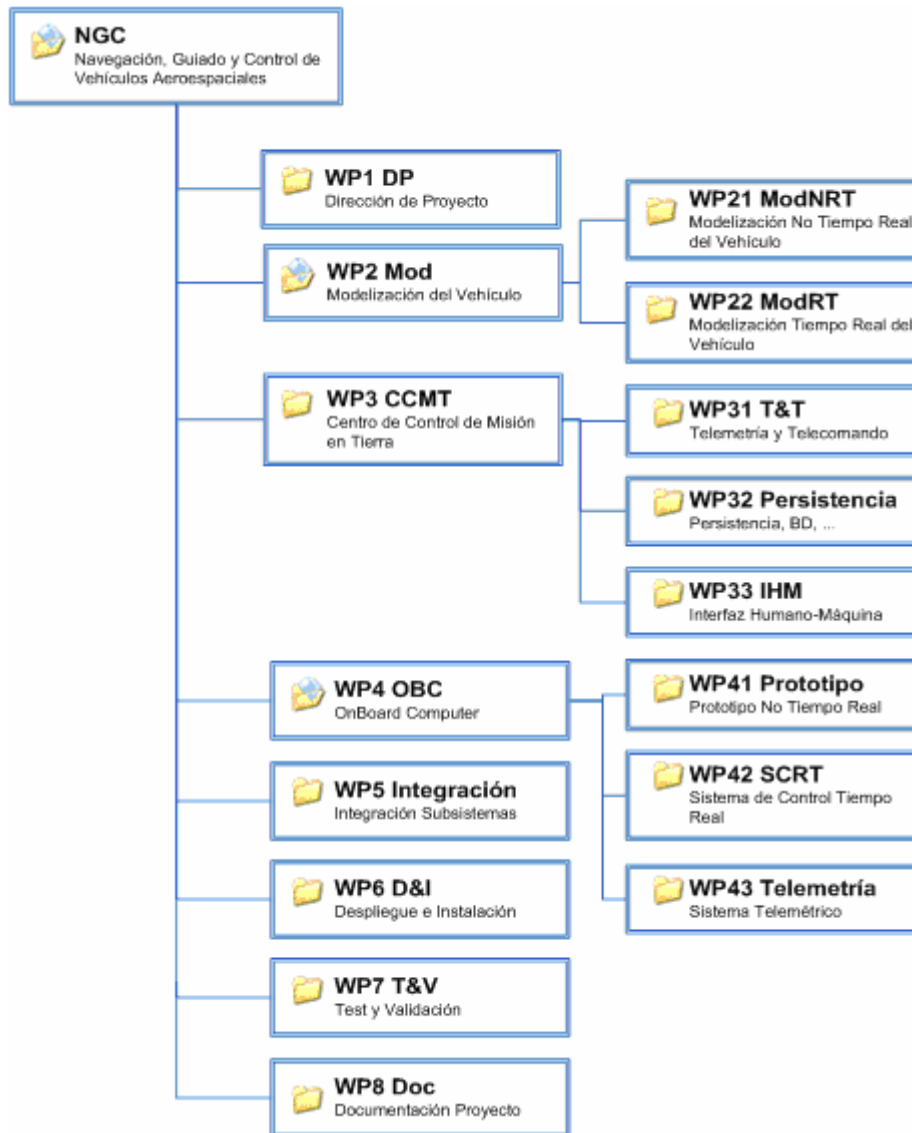


Figura 3: Work Breakdown Structure

El alcance de cada paquete de trabajo se describe a continuación:

- **WP1 DP:** Dirección de Proyecto.

El objetivo de este paquete de trabajo es dirigir y coordinar las actividades desarrolladas en el marco del proyecto.

Las tareas principales son: Coordinar el equipo de trabajo, organizar reuniones de seguimiento, gestionar las compras de equipos, licencias, etc.

- **WP2 Mod:** Modelización del Vehículo.

El objetivo de este paquete de trabajo consiste en definir e implementar los módulos de simulación del cohete ó vehículo aeroespacial en sus versiones no tiempo real y tiempo real.

El paquete de trabajo se ha dividido en dos paquetes:

- **WP21 ModNRT:** Modelización No Tiempo Real.
- **WP22 ModRT:** Modelización Tiempo Real.

Los modelos matemáticos integrados en el simulador del vehículo incluirán modelos: geofísicos, dinámico, geométrico, aerodinámico, propulsión y actuadores.

El paquete de trabajo incluye, además de la definición y diseño de los modelos, la implementación y la validación de los mismos.

Para el desarrollo del paquete de trabajo se requiere la información del vehículo a simular.

- **WP3 CCMT:** Centro de Control de Misión en Tierra.

El objetivo de este paquete de trabajo es desarrollar el subsistema CCMT.

Las tareas a realizar en el marco del paquete de trabajo son: definición de requisitos, prototipado de la interfaz de usuario, diseño, implementación y validación del subsistema.

El paquete de trabajo se ha subdividido en tres:

- **WP31 T&T:** Telemetría y Telecomando.
- **WP32 Persistencia:** Persistencia y Base de Datos.
- **WP33 IHM:** Interfaz Humano-Máquina.

El sistema de telemetría y telecomando incluye la definición del protocolo de comunicación y la implementación de los *drivers* de comunicación.

El WP32 comprende la definición y diseño de la base de datos y la implementación del componente encargado de la persistencia de los datos manejados por el sistema.

El WP33 abarca el desarrollo de la maqueta de la interfaz gráfica de usuario y su implementación e integración en el subsistema CCMT.

- **WP4 OBC:** OnBoard Computer.

El trabajo a desarrollar en el WP4 consiste en implementar el sistema embarcado del vehículo.

El objetivo de este paquete de trabajo es adquirir experiencia en el desarrollo de sistemas de control digital para vehículos aeroespaciales. El conocimiento relacionado con la dinámica de vehículos aeroespaciales es aplicable no sólo en el ámbito espacial sino también en los sectores aeronáutico y militar. La experiencia conseguida mediante el desarrollo del sistema de control del vehículo es aplicable en múltiples ámbitos de la

ingeniería. Al mismo tiempo, el equipo de ingenieros profundizará en el uso de herramientas de simulación y control.

Las tareas a realizar en este paquete de trabajo son: análisis y definición de requisitos, diseño, implementación, test y validación del subsistema.

El diseño del sistema embarcado OBC se realizará conforme al objetivo del proyecto, definiendo interfaces genéricas para permitir intercambiar ó integrar futuros sistemas de control para nuevos vehículos aeroespaciales.

Por el último, el WP4 se ha dividido en los siguientes paquetes:

- **WP41 Prototipo:** Prototipo No Tiempo Real.
- **WP42 SCRT:** Sistema de Control Tiempo Real.
- **WP43 Telemetría:** Sistema Telemétrico.

El WP41 se centrará en el desarrollo de un primer prototipo del sistema de control.

El WP42 estará focalizado en la implementación del sistema de control tiempo real para el vehículo aeroespacial seleccionado.

El WP43 consiste en implementar el sistema de telemetría integrado en el OBC.

- **WP5 Integración:** Integración Subsistemas.

La integración de todos los subsistemas desarrollados y los correspondientes test de integración se realizarán en el marco de este paquete de trabajo.

El resultado del WP5 será el sistema integrado y listo para instalar y el informe con los resultados de los test de integración.

- **WP6 D&I:** Despliegue e Instalación.

La instalación en la plataforma de ensayos y la sintonización del sistema se realiza en este paquete de trabajo.

Al finalizar el paquete de trabajo el sistema estará instalado en la plataforma de ensayos definitiva listo para ser validado.

- **WP7 T&V:** Test y Validación.

El WP7 tiene como objetivo validar el sistema. El equipo de desarrollo redactará el Plan de Validación y los Procedimientos de Test a ejecutar para validar el sistema completo.

- **WP8 Doc:** Documentación Proyecto

La documentación del proyecto se realizará en el marco de este paquete de trabajo.

2.4 ANÁLISIS DE RIESGOS

Los puntos que se vislumbran problemáticos en el proyecto y las soluciones previstas en caso de que se presenten son:

- **RISK 01:**
 - Riesgo: Coordinación de las actividades.
 - Solución: Reuniones de seguimiento periódicas.
- **RISK 02:**
 - Riesgo: Uso de nuevas tecnologías.
 - Solución: Formación.
- **RISK 03:**
 - Riesgo: Falta de licencias de herramientas, sistemas tiempo real, etc.
 - Solución: Adquisición de licencias.
- **RISK 04:**
 - Riesgo: Falta de motivación del equipo.
 - Solución: Fijar hitos y entregables.

2.5 PLANIFICACIÓN

El diagrama de Gantt de la Figura 4 muestra la planificación prevista para el desarrollo del proyecto.

La estimación prevista para el desarrollo completo del proyecto es de 125 días hábiles contando con el equipo de proyecto definido en el primer apartado del capítulo y con una dedicación del 100 % de los recursos asignados al proyecto.

Los ingenieros de proyecto no tienen tareas asignadas durante las tres primeras semanas, no obstante, dedicarán su tiempo a formación y a evaluar los COTS.

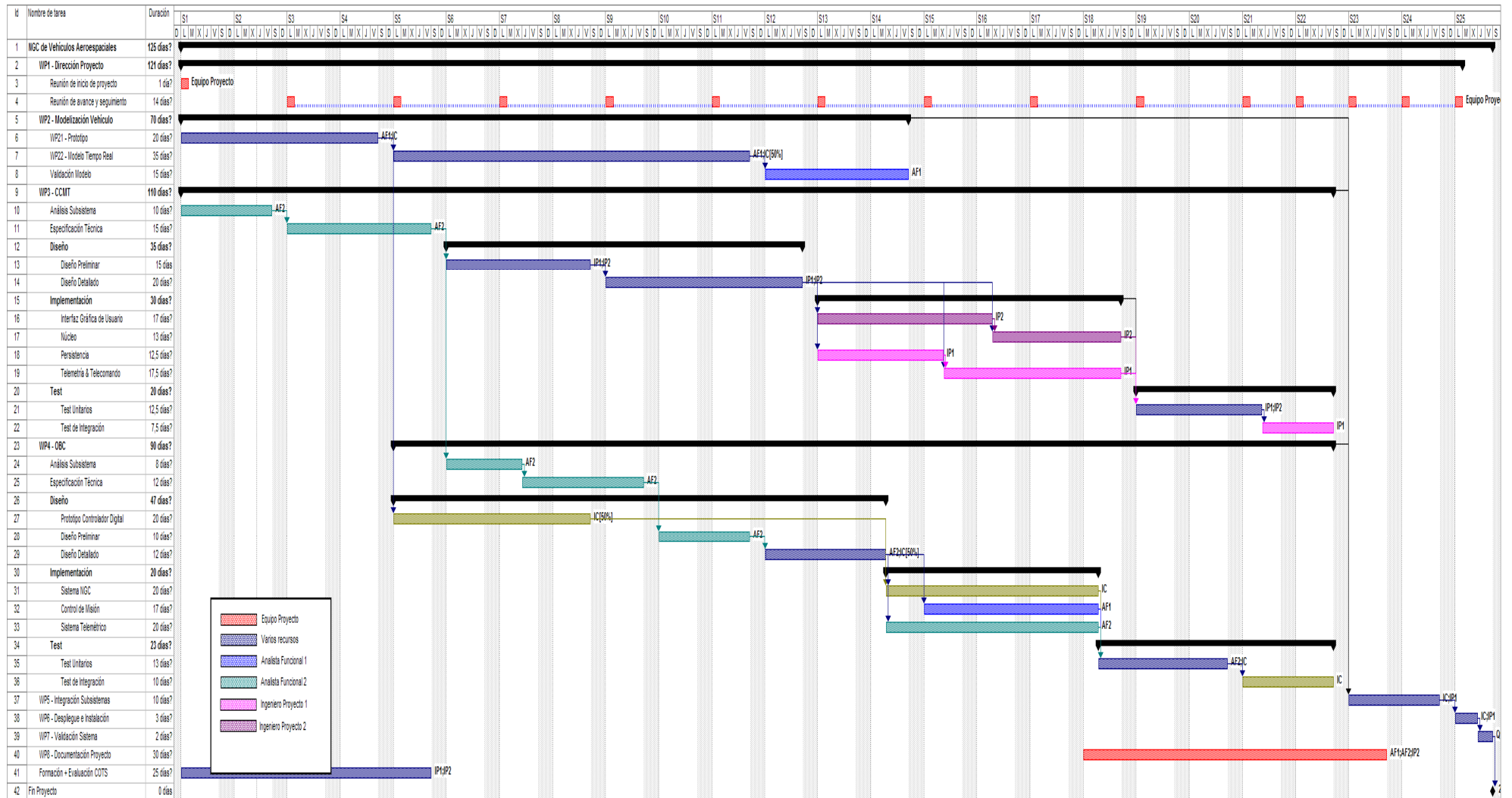


Figura 4: Diagrama de Gant

3 Análisis

El análisis del sistema presentado en este capítulo se estructura entorno a tres apartados. La primera parte del estudio recoge el análisis científico, la segunda está dedicada al análisis del sistema informático y la última sección detalla el análisis económico.

El objetivo del análisis científico es doble, por un lado, pretende introducir el contexto en el que se va a desarrollar el proyecto, a la vez que permite adquirir el *background* necesario para entender el problema al que nos enfrentamos.

Una vez comprendido el problema, estaremos en disposición de ofrecer una solución técnica desde el punto de vista de la ingeniería del software.

Por último, evaluaremos el coste económico que supone desarrollar el sistema informático propuesto.

3.1 ANÁLISIS CIENTÍFICO

El análisis científico se centra en el vehículo aeroespacial y en las características de una misión.

La descripción del vehículo y de la misión nos permitirá conocer la complejidad que conlleva el desarrollo de un sistema de navegación, guiado y control.

3.1.1 DESCRIPCIÓN GENERAL DEL VEHÍCULO

El objeto de este capítulo es describir el vehículo aeroespacial y los subsistemas que lo componen.

Los modelos matemáticos a desarrollar en el paquete de trabajo WP2 Modelización del Vehículo (Ver 2.3) estarán basados, lógicamente, en el vehículo real.

En primer lugar, describiremos el objetivo para el que fue concebido el vehículo aeroespacial a ensayar. En nuestro caso, simularemos un mini-lanzador cuyo objetivo es poner en órbita micro-satélites.

A continuación, definiremos el modelo de vehículo aeroespacial en relación al número de etapas que lo componen. El vehículo a ensayar, ya sea, misil, cohete, lanzador, etc..., puede ser de una, dos ó más etapas. Cada una de las etapas del vehículo estará propulsada por un modelo de combustible, por ejemplo, el vehículo podría ser un cohete bietapa propulsado por dos motores cohete.

Vehículos de más de una etapa dispondrán de sistemas de separación entre etapas. El sistema de separación interetapas es un mecanismo ó actuador que accionado electrónicamente por el sistema de control permitirá liberar el motor cohete propulsante de la fase de vuelo finalizada.

El vehículo aeroespacial estará equipado con diferentes actuadores que permitirán el control del mismo durante las diferentes fases de vuelo.

El vehículo aeroespacial simulado para validar la plataforma de ensayos estará equipado con tres sistemas, uno para cada fase de vuelo. Durante la primera fase de vuelo el sistema de control dispondrá de un sistema de aletas para estabilización y control en balance. La segunda fase de vuelo contará con un sistema de control del vector de empuje. Por último, la tercera fase de vuelo se controlará mediante un sistema de motores de gas frío, además, esta última fase de vuelo incluirá los actuadores que permitirán abrir la cofia para poner en órbita el satélite.

La Figura 5 muestra el vehículo aeroespacial a ensayar, las etapas de vuelo y las inter-etapas.

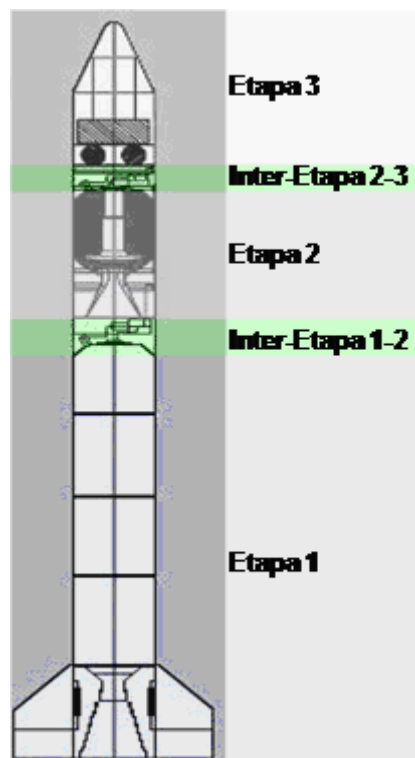


Figura 5: Vehículo aeroespacial

3.1.2 DESCRIPCIÓN DE LA MISIÓN

En este apartado se definirá una misión tipo como base para el desarrollo de los trabajos a realizar. La definición de una misión modelo nos permitirá conocer los eventos típicos durante el vuelo del vehículo.

La información necesaria para definir una misión es:

- Lugar de lanzamiento.
- Perfil de Misión.
- Trayectoria de lanzamiento.

3.1.2.1 Lugar de lanzamiento

El lugar de lanzamiento es un dato informativo para el sistema puesto que la plataforma de ensayos simulará el vuelo del vehículo.

En el caso real, el lugar de lanzamiento es un dato importante por varios motivos. El lector interesado puede consultar el documento de referencia [DR03].

Para especificar la posición del lugar de lanzamiento debemos definir el sistema de coordenadas utilizado. Los sistemas de referencia utilizados los podemos dividir en sistemas generalizados y sistemas de referencia específicos.

- Sistemas de referencia generalizados.

Los sistemas de referencia generalizados que utilizaremos a lo largo del proyecto son:

- Universal Transfer Mercator (UTM):
 1. X (Easting)
 2. Y (Northing)
 3. Z: (Cota)
 4. Huso
- Geográficas:
 1. Latitud
 2. Longitud
 3. Altura

Para los dos sistemas de referencia generalizados especificaremos el siguiente parámetro:

- Datum:
 1. Elipsoide de referencia.
 2. Traslación: Δx , Δy , Δz .
 3. Rotación: Θ_x , Θ_y , Θ_z .
- Sistema de referencia específico.

El sistema de referencia específico es un sistema de coordenadas propio. El sistema utilizado a lo largo del proyecto será el RCS (Range Coordinate System). Para definir el sistema de coordenadas RCS debemos detallar los siguientes parámetros:

- Origen del sistema de coordenadas (0, 0, 0) en el sistema de coordenadas general.
- Descripción del sistema. El sistema de coordenadas utilizado a lo largo del proyecto se define como un sistema ortogonal (por lo tanto no tiene en cuenta la curvatura de la tierra), el eje de coordenadas Y en la dirección del norte UTM y el eje Z en el sentido de las alturas positivas.

3.1.2.2 Perfil de misión

Entendemos por perfil de misión la definición de los valores más característicos de una misión, es decir, masa de la carga útil, el tiempo consumido en cada etapa, los eventos de mayor relevancia y los instantes de tiempo en los que sucederán.

La Figura 6 muestra un esquema de una misión típica indicando los hitos principales de la misión.

Posteriormente, se analizan brevemente cada una de las fases en las que se divide una misión típica del vehículo.

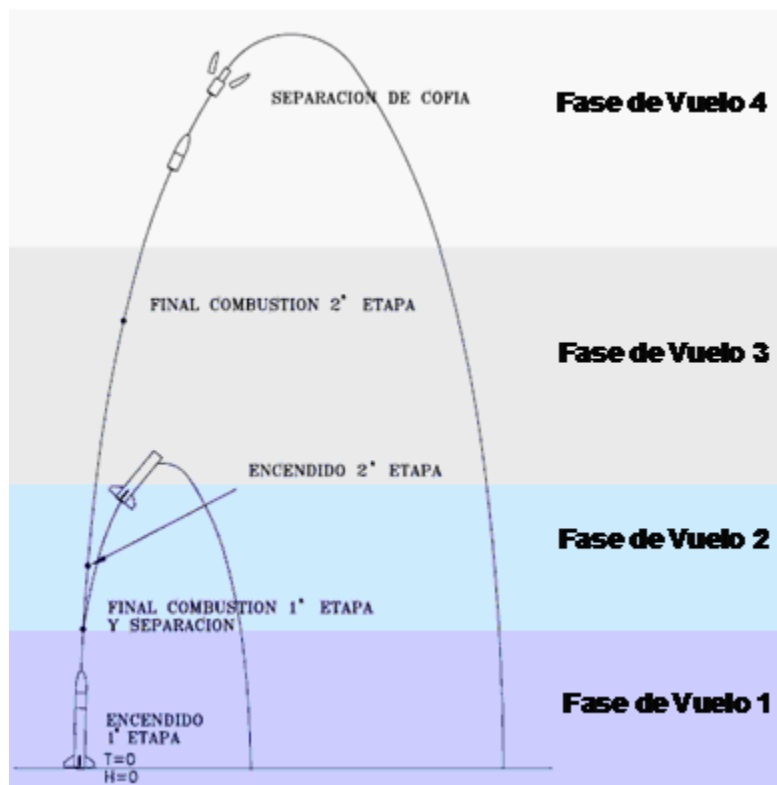


Figura 6: Esquema de misión

- **Fase 0**

La primera fase de la misión la denominaremos pre-vuelo. En esta primera fase se realizará el chequeo de todos los subsistemas que componen el vehículo.

- **Fase 1**

La fase de vuelo 1 corresponde a la primera etapa de vuelo del vehículo. Esta fase se puede descomponer en varias fases.

- Fase 1.1:

La fase 1.1 corresponde al desplazamiento del vehículo por la rampa de lanzamiento. Una vez accionado el mecanismo de ignición del pri-

mer motor cohete, el vehículo empezará a desplazarse por la rampa de lanzamiento debido a los gases generados por la combustión del motor cohete. En este momento, el vehículo se desconectará del cordón umbilical (El cordón umbilical no es más que un conjunto de señales digitales que interconectan el sistema CCMT y el vehículo; la función del mismo se definirá en los próximos apartados).

- Fase 1.2

Durante esta fase, el vuelo del vehículo será libre puesto que las aletas permanecerán fijas hasta pasados los primeros segundos. Por lo tanto, la trayectoria del vehículo carecerá de acción de control.

- Fase 1.3

Vuelo de la primera etapa del vehículo con control en balance. El sistema de control deberá accionar las aletas con el objetivo de limitar el espín del vehículo. El objetivo del sistema de control en esta fase de vuelo es mantener la velocidad en espín entorno a 0 rad/s.

Esta etapa de vuelo es crítica. Una acción de control no apropiada sobre las aletas del vehículo provocaría un incremento progresivo del espín, lo que inhabilitaría el navegador inercial y como consecuencia la misión debería abortarse.

- **Fase 1**

Fase de vuelo inter-etapas 1-2. Esta fase de vuelo se corresponde con el intervalo de vuelo no propulsado que se extenderá desde el instante en que finalice la combustión del primer motor cohete hasta el instante en que se inicie la propulsión de la segunda etapa.

Durante esta fase de vuelo se producirá la separación del motor cohete de la primera etapa del resto del vehículo.

El sistema de control deberá activar el sistema separación de la primera etapa.

El tiempo transcurrido entre la separación de la primera etapa y la activación del segundo motor cohete, es un parámetro que deberá obtenerse empíricamente durante el desarrollo del sistema de control. Este tiempo será el necesario para asegurar que la primera etapa inerte se encuentra lo suficientemente alejada del resto del vehículo para no interferir en el proceso de ignición de la segunda etapa.

La primera etapa se destruirá en el aire mediante una carga explosiva que se activará segundos después de la separación de las etapas.

La criticidad de la fase 2 radica en la variación instantánea de la masa y el desplazamiento del centro de gravedad del vehículo.

- **Fase 3**

Vuelo de la segunda etapa. La fase de vuelo 3 comienza con la ignición del segundo motor cohete.

El sistema de navegación, guiado y control del vehículo deberá actuar sobre el sistema de control del vector de empuje del motor cohete para mantener la trayectoria teórica planificada.

La señal de control aplicada al actuador del mecanismo que controla el vector de empuje modificará el *pitch* (cabeceo) y/o el *yaw* (guiñada) del vehículo.

Una vez completada la combustión del segundo motor cohete, el vehículo se encontrará en la capa atmosférica conocida como mesosfera y habrá alcanzado una altura superior a 60 Km. Por lo que el vuelo posterior se puede considerar exento de influencias atmosféricas.

- **Fase 4**

Fase de vuelo 4 ó vuelo cautivo. Finalizada la combustión del motor cohete de la segunda etapa, el vehículo continuará su trayectoria sin ningún tipo de propulsión. Al igual que ocurriera con la primera etapa, segundos después de finalizar la combustión del segundo motor se activará el sistema de separación de la segunda etapa. La segunda etapa se destruirá en el aire mediante una carga explosiva que se activará 5 segundos después de ocurrida la separación de la misma.

Durante esta fase de vuelo el control del vehículo se realizará exclusivamente mediante el sistema de control por motores de gas frío. A partir de este instante el vehículo se orientará en el espacio por medio de los motores de gas frío manteniendo la actitud del vehículo definida en la trayectoria teórica.

Esta fase termina con la apertura de la cofia y la satelización de la carga útil.

3.1.2.3 Secuencia de eventos

La secuencia de eventos descrita en este apartado es trascendental para el desarrollo del sistema de control del vehículo y está estrechamente relacionada con el perfil de misión.

La definición de la secuencia de eventos nos permitirá conocer el autómata ó máquina de estados a implementar en el subsistema OBC.

En primer lugar, definiremos que entendemos por evento.

- **Evento:** Se entiende por evento una situación definida y prevista que se produce durante la misión y de la que se derivan acciones que influyen o determinan la misión.

Desde el punto de vista de la generación de eventos la misión se divide en dos partes:

- **Prevuelo:** Esta fase comprende desde el encendido de los equipos hasta el envío de la señal de ignición del primer motor cohete.
- **Vuelo:** Esta fase se inicia con la recepción de la señal de ignición del primer motor cohete y finaliza con la satelización de la carga.

Los eventos definidos en este apartado sólo incluyen aquellos eventos relevantes desde el punto de vista de la simulación de una misión, es decir, detallaremos los eventos que deban ser considerados para el diseño de los subsistemas que conforman la plataforma de ensayos. Por lo tanto, obviaremos todos los eventos específicos del sistema real y que no sean necesarios en una simulación, por ejemplo, comprobaciones relacionadas con la seguridad, acciones a ejecutar en la lanzadera, etc.

Dicho esto, los eventos de la fase pre-vuelo correspondientes a los instantes previos al lanzamiento del vehículo son:

1. Arranque del sistema.
2. Chequeo de los subsistemas.
3. Chequeo de los modos de navegación.
4. Señal de ignición del motor cohete.

Los eventos durante la fase de vuelo son:

5. Inicio del movimiento (t_0).
6. Inicio del control en balance (sistema de aletas) (t_1).
7. Detección del fin de combustión de la 1ª etapa (t_2).
8. Fin del control en balance del sistema de aletas (t_3).
9. Armado electrónico del sistema pirotécnico interetapa 1-2 (t_4).
10. Activación del sistema de separación de la 1ª etapa (t_5)
11. Detección de la separación de la 1ª etapa (t_6)
12. Activación del sistema de ignición del segundo motor (t_7).
13. Detección del inicio de la combustión del motor (t_8).
14. Inicio del control en cabeceo y guiñada (sistema de control del vector de empuje) (t_9).
15. Detección del fin de combustión de la 2ª etapa (t_{10}).
16. Fin del control en cabeceo y guiñada (sistema de control del vector de empuje) (t_{11}).
17. Armado electrónico del sistema pirotécnico interetapa 2-3 (t_{12}).
18. Activación del sistema de separación de la 2ª etapa (t_{13}).
19. Detección de la separación de la 2ª etapa (t_{14}).
20. Activación del sistema de apertura de la cofia (t_{15}).
21. Detección de la separación de la cofia (t_{16}).
22. Inicia del control en tres ejes por el sistema de motores de gas frío (t_{17}).
23. Maniobra programa según misión (t_{18}).

La Figura 7 muestra el diagrama de estados del vehículo, resume la secuencia de eventos detallada anteriormente e indica que eventos generan las transiciones entre estados.

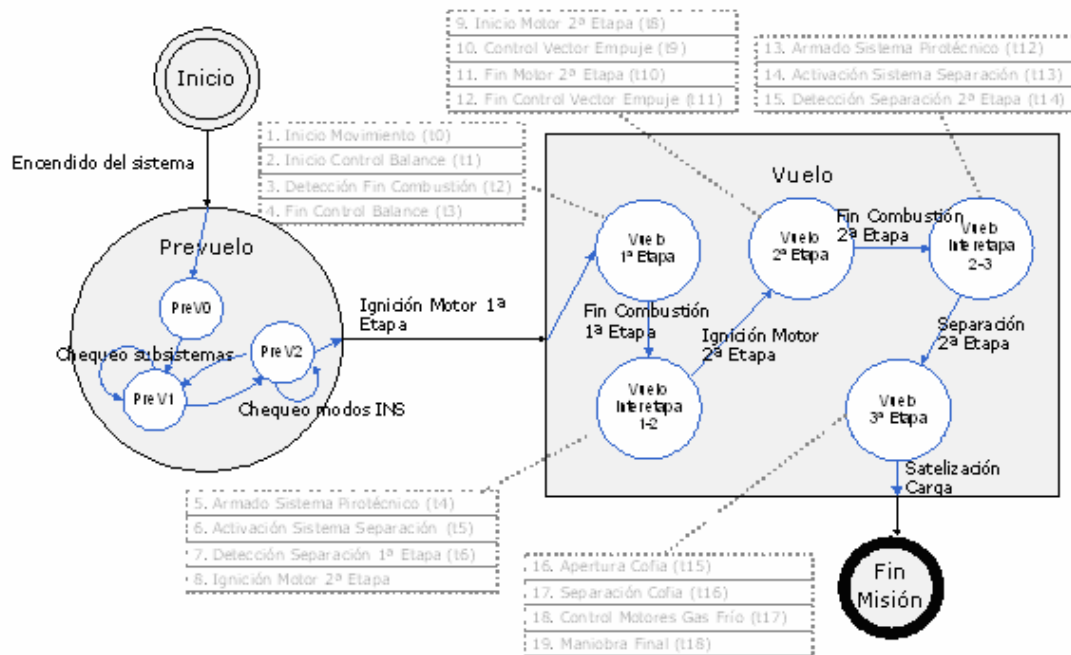


Figura 7: Secuencia de Eventos

3.1.2.4 Trayectoria de referencia

En este apartado detallaremos los parámetros que definen la trayectoria de referencia a seguir por el vehículo.

Los parámetros descritos a continuación son de vital importancia para el diseño del sistema de control integrado en el Omboard Computer.

La trayectoria la dividiremos en dos tramos, el primer tramo corresponde a la trayectoria del vehículo en la rampa de lanzamiento y el segundo al resto de la trayectoria.

- Parámetros de la trayectoria en rampa:
 - TIEMPO: Tiempo en segundos (s).
 - DIST: Distancia en metros (m).
 - VEL: Velocidad en metros por segundo (m/s).
 - ACEL: Aceleración en metros por segundo al cuadrado (m/s^2).
 - EMPUJE: Empuje en Newtons (N).
 - MASA: Masa del vehículo en kilogramos (Kg).
 - ROZ: Rozamiento en Newtons (N).
 - RES: Resistencia aerodinámica en Newtons (N)
- Parámetros de la trayectoria fuera de rampa:
 - TIEMPO: Tiempo en segundos (s).

- ALCANCE: Distancia recorrida sobre la superficie terrestre en kilómetros (Km).
- ALTURA: Distancia a la superficie terrestre según la vertical local en kilómetros (Km).
- ACEL: Aceleración longitudinal del vehículo en ejes cuerpo (compone en el eje X) en metros por segundo al cuadrado (m/s^2).
- VEL: Módulo del vector velocidad en metros por segundo (m/s).
- EMPUJE: Empuje en Newtons (N).
- MASA: Masa del vehículo en kilogramos (Kg).
- CD: Coeficiente de resistencia aerodinámico del vehículo.
- ANG: Ángulo entre la vertical del punto de lanzamiento y la vertical del lugar en grados sexagesimales.
- PD: Presión dinámica en N/cm^2 ($PD=1/2 \rho v^2$)
- BETA (β): Ángulo formado por la vertical del punto de lanzamiento y el vector velocidad en grados sexagesimales.
- THETA (θ): Ángulo de asiento. Ángulo formado entre el eje del vehículo y la horizontal local del punto de lanzamiento en grados sexagesimales.
- GAMMA (γ): Ángulo de asiento del vector velocidad. Ángulo formado entre el vector velocidad y la horizontal local del punto de lanzamiento.
- ALFA (α): Ángulo de ataque ($\alpha=\gamma-\theta$) en grados sexagesimales.

3.1.3 SISTEMAS DE REFERENCIA

Los sistemas de referencia a utilizar durante en el desarrollo del proyecto se resumen en la Figura 8 y se describen a continuación en este mismo apartado.

- **Sistema Inercial (O_i, X_i, Y_i, Z_i).**

El sistema de coordenadas inercial definido para el sistema tiene el origen del sistema O_i situado en el centro de la Tierra. El eje de coordenadas Z_i coincide con el eje de giro de la Tierra. El eje X_i es la intersección del plano del meridiano que pasa por el punto de lanzamiento con el plano del ecuador. El eje Y_i completa un sistema de coordenadas ortogonal *dextrosum*.

- **Sistema Local (O_L, X_L, Y_L, Z_L).**

El sistema de coordenadas local es un sistema móvil cuyo origen de coordenadas O_L está en el centro de la Tierra. El eje de coordenadas Z_L coincide en todo instante con la línea que une el centro de la Tierra con el centro de gravedad del vehículo y el sentido positivo hacia arriba. El eje X_L está contenido en el plano definido por Z_L y el eje de giro de la Tierra, siendo positivo hacia el Sur. Por último, el eje Y_L completa el triedro a derechas.

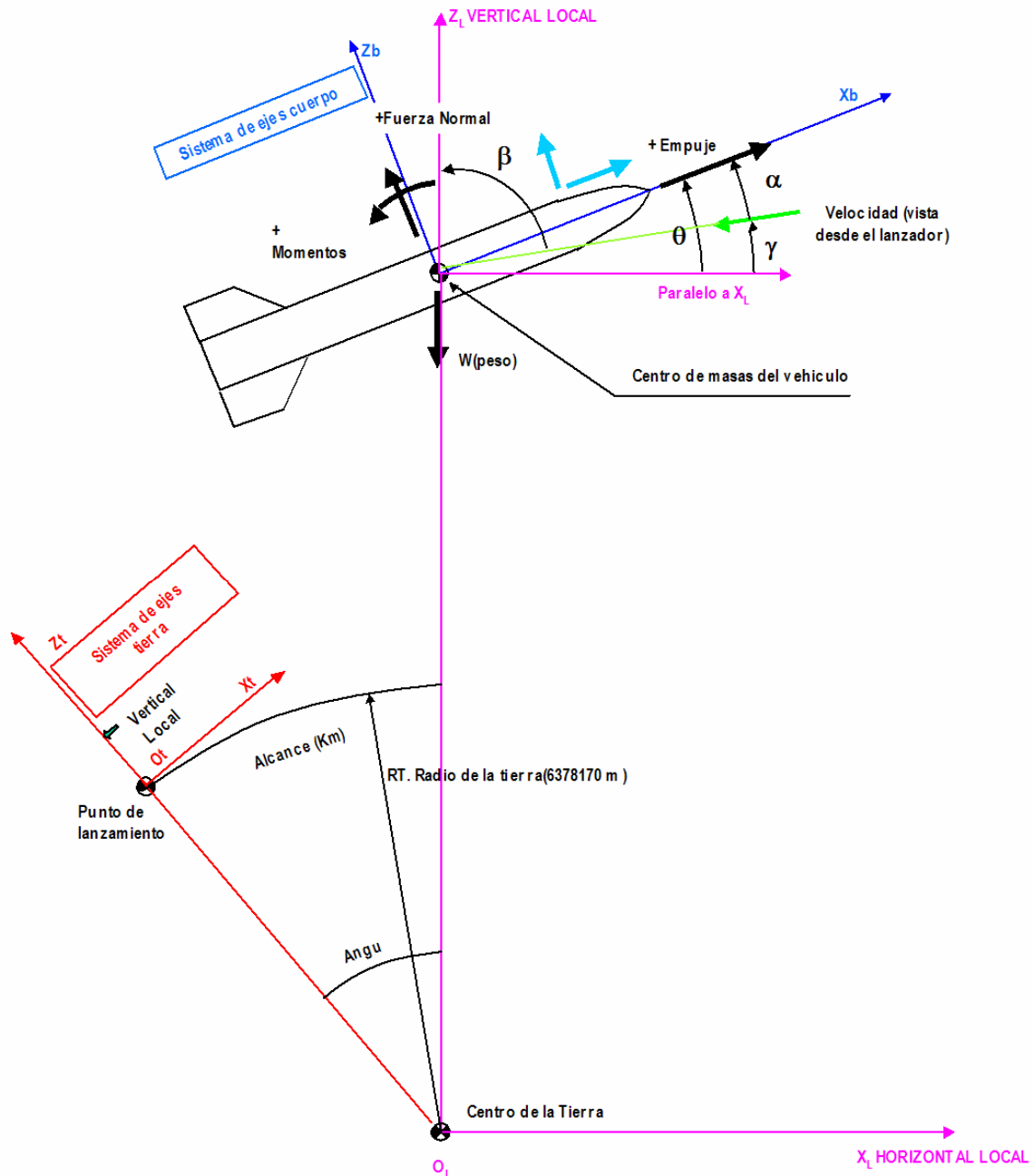


Figura 8: Sistemas de referencia

- **Sistema de Ejes Tierra (O_T , X_T , Y_T , Z_T).**

El origen del sistema de coordenadas de ejes Tierra O_T está en el punto de lanzamiento. Los ejes de coordenadas X_T e Y_T están contenidos en el plano tangente a la superficie de la Tierra en el punto de lanzamiento, estando X_T determinado por el ángulo de azimut del vehículo en la rampa de lanzamiento medido desde el norte y positivo en el sentido de las agujas del reloj. El eje Z_T dirigido según la vertical geográfica en el punto de lanzamiento y positivo hacia arriba. El eje Y_T completa el triedro a derechas definiendo un sistema ortogonal dextrosom.

- **Sistema de Ejes Plataforma (O_P, X_P, Y_P, Z_P).**

El origen de coordenadas del sistema de ejes plataforma O_P está situado en el centro de gravedad de la plataforma y su orientación se mantiene en todo momento constante respecto a un sistema de referencia inercial, coincidiendo inicialmente con la orientación del sistema de ejes Tierra.

- **Sistema de Ejes Cuerpo (O_B, X_B, Y_B, Z_B).**

El sistema de ejes cuerpo es un sistema de coordenadas fijo al vehículo cuyo origen del sistema O_B se encuentre en el centro de gravedad del vehículo sobre el eje longitudinal. El eje de coordenadas X_B según el eje longitudinal del vehículo y positivo en el sentido del empuje. Los ejes Y_B y Z_B están contenidos en un plano normal al eje longitudinal por O_B orientados según la dirección de cada uno de los pares de aletas, Y_B hacia la derecha de la dirección de lanzamiento tal y como se muestra en la Figura 9.

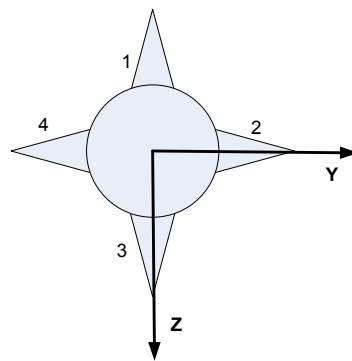


Figura 9: Orientación Ejes Cuerpo

- **Sistema de Ejes Guiado (O_G, X_G, Y_G, Z_G).**

El sistema de coordenadas ejes guiado tiene el origen del sistema O_G situado en el mismo punto que el origen del sistema de ejes Tierra orientando el eje X_G en la dirección del eje del vehículo en el momento que finaliza la combustión del motor cohete de la 2ª etapa.

3.2 ANÁLISIS DEL SISTEMA INFORMÁTICO

El proyecto trata de desarrollar una plataforma de ensayos que permita simular el vuelo de cualquier vehículo aeroespacial.

La plataforma de simulación presentada en este documento permitirá integrar modelos matemáticos de vehículos aeroespaciales, integrar y testear el sistema de control diseñado para guiar el vehículo modelado, visualizar la trayectoria de referencia y simulada para una misión predefinida, generar históricos para su posterior análisis, etc.

La plataforma de ensayos proporcionará un entorno de simulación completo que permitirá enfocar el trabajo de los ingenieros en el modelado del vehículo y en el sistema de navegación, guiado y control, despreocupándose del desarrollo de interfaces de usuario, protocolos de comunicación, base de datos, etc.

En lo que resta de capítulo presentaremos la arquitectura propuesta para el sistema y los protocolos de comunicación diseñados para el intercambio de información entre los subsistemas que forman la plataforma de simulación.

3.2.1 ARQUITECTURA DEL SISTEMA

El sistema a desarrollar está compuesto por tres subsistemas, a saber:

- **SimRT-VA:** Simulador Tiempo Real del Vehículo Aeroespacial.
- **OBC:** OnBoard Computer.
- **CCMT:** Centro de Control de Misión en Tierra.

La Figura 10 muestra la arquitectura del sistema y los módulos principales que componen los subsistemas de la plataforma de ensayos objeto de desarrollo.

Los subsistemas citados anteriormente pueden agruparse en dos componentes atendiendo a la localización teórica del sistema real, es decir, Segmento Tierra y Sistema Embarcado.

El Segmento Tierra corresponde al CCMT mientras que el Sistema Embarcado está compuesto por el vehículo simulado, subsistema SimRT-VA, y el subsistema OBC.

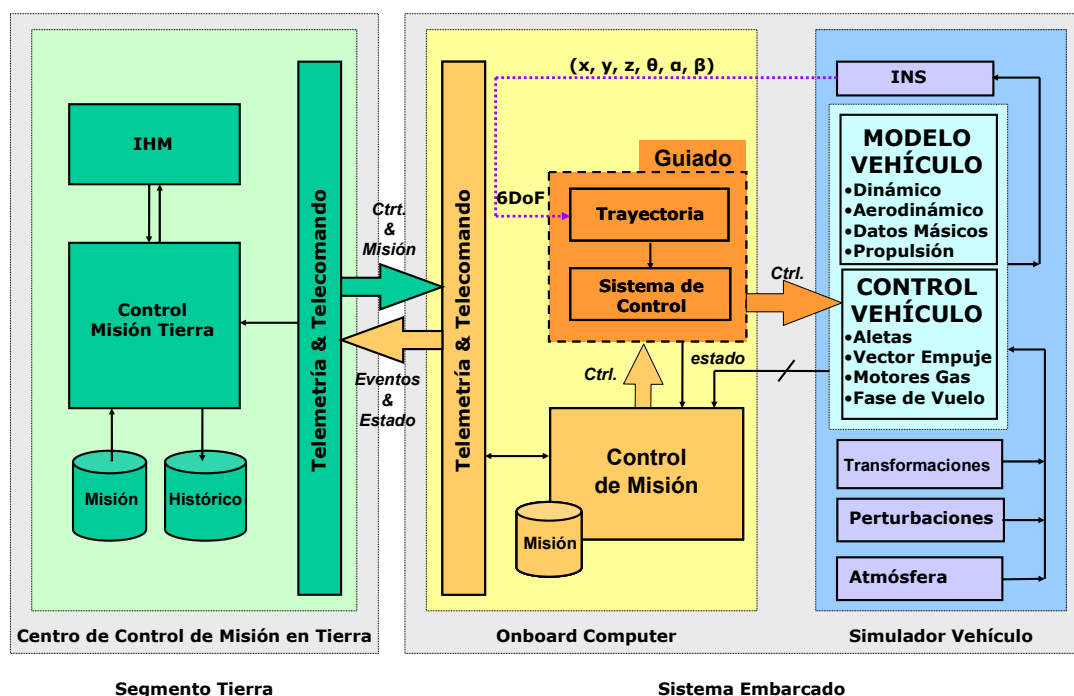


Figura 10: Arquitectura del sistema

3.2.1.1 Centro de Control de Misión en Tierra

El Centro de Control de Misión en Tierra se ha concebido como un sistema de monitorización y control que debe implementar las siguientes funcionalidades:

- Interfaz gráfica de usuario.
- Visualización del vector de estado del vehículo simulado.
- Visualización gráfica de la trayectoria de referencia y real durante la fase de vuelo.
- Importación de trayectorias teóricas.
- Configuración de misión.
- Gestión de la base de datos de misiones.
- Generación de históricos de misión.
- Envío de comandos al Onboard Computer: Cargar Misión, Iniciar Misión, Abortar Misión, Autodestrucción, Testear Subsistemas, etc.

La arquitectura software del subsistema CCMT implementará el patrón de diseño Modelo Vista Controlador (MVC).

El subsistema CCMT se puede descomponer en los siguientes componentes software:

- **IHM**: Interfaz-Humano-Máquina. Este componente implementa la interfaz gráfica de usuario.
- **T&T**: Telemetría y Telecomando. Es el componente encargado de las comunicaciones. La función principal de este componente es la de asegurar el intercambio de datos entre los subsistemas CCMT y OBC. El intercambio de datos debe permitir:
 - Sincronizar la hora del OBC a partir de la hora del sistema CCMT.
 - Recibir la información sobre Eventos, Estados y Vector de Estado.
 - Enviar datos de configuración y comandos al OBC.
 - Informar del estado de las comunicaciones.

El sistema CCMT se comunicará con el OBC a través de los siguientes enlaces:

- Línea serie RS-422; Protocolo HDLC.
- Líneas digitales.

El sistema de comunicación dispondrá de tres *buffers* de datos para mantener la información procedente del OBC. La Figura 11 muestra un gráfico descriptivo de la gestión de los buffers de datos.

Por otro lado, el componente publicará una interfaz con los métodos necesarios para permitir la lectura de los *buffers* de datos y configurar el hardware. El componente lanzará eventos para notificar cambios de estado, errores, etc.

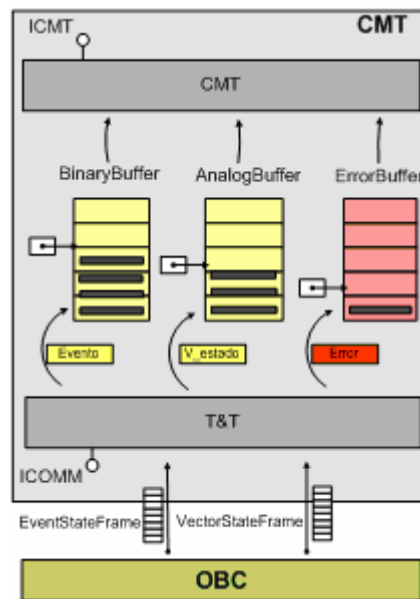


Figura 11: Funcionamiento buffers

- **DATA:** Modelo de datos del sistema. La función de este componente es:
 - Definir las estructuras de datos que utilizará el subsistema CCMT.
 - Gestionar la base de datos. El sistema CCMT integrará una base de datos para almacenar la información generada a lo largo de una misión. Este componente implementará la gestión de la base de datos del subsistema CCMT.
 - Gestionar la E/S de ficheros. El componente se encargará de la gestión de E/S del subsistema CCMT. Este módulo de software debe permitir importar la trayectoria nominal de la misión y exportar el histórico de una misión almacenada en la base de datos.
 - Mantener en memoria la información de la misión en curso.
- **CMT:** El Controlador de Misión en Tierra es el núcleo del subsistema CCMT. Implementa la lógica de la aplicación. La interacción con el resto de componentes de la aplicación se resume a continuación:
 - Instancia, inicializa y destruye el resto de componentes de la aplicación.
 - Gestiona los comandos ejecutados por el usuario a través de la interfaz de usuario.
 - Distribuye la información entre los componentes.

3.2.1.1.1 Definición de los ficheros de trayectorias

La trayectoria nominal del vehículo para una misión se almacenará en un fichero *Excel* (.xls). El formato del fichero será el siguiente:

- El fichero *Excel* deberá contener la trayectoria nominal en la primera hoja del libro (siguiendo la nomenclatura de la aplicación *Microsoft Excel*).

- Los parámetros que definen una trayectoria se almacenarán en columnas individuales y consecutivamente empezando en la primera columna de la hoja.
- La trayectoria deberá empezar en la primera fila de la hoja *Excel*. Este primer registro de la tabla coincidirá con el instante de tiempo $t_0 = 0$.
- La **iError! No se encuentra el origen de la referencia.** resume los parámetros de la trayectoria en el tramo correspondiente al desplazamiento del vehículo en la rampa de lanzamiento.

Parámetro	Descripción
EVENT	Evento asociado.
ID_EVENT	Identificador de evento
T	Tiempo (s).
DIST	Distancia (m).
VEL	Velocidad (m/s).
ACEL	Aceleración (m/s^2)
EMPUJE	Empuje (N).
MASA	Masa del vehículo (kg).
ROZ	Rozamiento (N).
RES	Resistencia aerodinámica (N).

Tabla 3: Fichero de trayectoria en rampa

- La Tabla 4 resume los parámetros de la trayectoria durante el vuelo del vehículo.

Parámetro	Descripción
EVENT	Evento asociado.
ID_EVENT	Identificador de evento
TIEMPO	Tiempo en segundos (s).
ALCAN	Alcance en Kilómetros (km)
ALTURA	Altura en Kilómetros (km).
VELOC	Módulo del vector velocidad (m/s).
MASA	Masa del vehículo en Kilogramos (kg).
EMPUJE	Empuje en Newtons (N).
ACELAT	Aceleración longitudinal del vehículo en ejes cuerpo (componente en el eje X) (m/s^2).
MACH	Número Mach. (VELOC / Velocidad del sonido)
CD	Coeficiente de resistencia aerodinámico del vehículo.
BETA	Ángulo formado por la vertical del punto de lanzamiento y el vector velocidad (Sexagesimales).
ANG	Ángulo entre la vertical del punto de lanzamiento y la vertical del lugar (Sexagesimales).
PD	Presión dinámica en N/cm^2 ($PD = \frac{1}{2} \rho v^2$)
THETA	Ángulo de asiento (Sexagesimal).
GAMMA	Ángulo de asiento del vector velocidad (Sexagesimal).
ALFA	Ángulo de ataque ($\alpha = \gamma - \theta$) (Sexagesimales).

Tabla 4: Fichero de trayectoria en vuelo

3.2.1.1.2 Definición de los ficheros de históricos

La aplicación generará dos ficheros de datos con el histórico de la misión. En uno de los ficheros se guardará los diferentes eventos acaecidos durante la ejecución de la misión, tanto aquellos que se generan en el OBC como aquellos que se generan en el CCMT al realizar una acción puntual. El segundo fichero de datos recogerá el denominado vector de estado del vehículo, obteniendo así, un registro de los datos de posición, velocidad, aceleración y ángulos que ha ido caracterizando el movimiento del vehículo a lo largo de la misión. La información registrada incluirá la señal discreta de control calculada por el controlador en cada instante de tiempo.

La estructura de los ficheros de datos con el histórico de la misión se presenta a continuación.

Ambos ficheros compartirán una misma cabecera con el formato siguiente:

Identificador	Valor
Nombre Misión	M-01
Descripción	Satélite Alpha
Fecha Lanzamiento	28/12/2003
Hora Lanzamiento	18h35 (GMT)
Carga	190 Kg
Fichero de Trayectoria	traj_MI-01.xls

Tabla 5: Formato de la cabecera de los ficheros de histórico

El formato del fichero de datos históricos del vector de estado del vehículo será el siguiente:

Identificador	Valor
Tiempo	Segundos
Distancia	Metros
Altura	Metros
Velocidad	Metros / Segundo
Aceleración	Metros / Segundo ²
Señal de Control	--

Tabla 6: Formato fichero de histórico Vector de Estado

Por su parte, el fichero de datos históricos que recoge los eventos sucedidos a lo largo de la misión tendrá el formato siguiente:

Identificador	Valor
Tiempo	Segundos
Identificador de Evento	--
Estado	--

Tabla 7: Formato fichero de histórico Eventos

3.2.1.2 Onboard Computer

El Onboard Computer es el sistema de monitorización y control embarcado en el vehículo.

Las funciones principales que debe implementar el Onboard Computer son:

- Monitorización y control del vehículo.
- Control de la misión.
- Comunicación con el CCMT.

El diseño del subsistema OBC deberá permitir la integración de sistemas NGC para una gama de vehículos heterogénea. Por su parte, el desarrollo del sistema de NGC de un vehículo aeroespacial es homogéneo en un gran número de aspectos.

El sistema de navegación, guiado y control debe inicializarse con la misión a ejecutar. La misión debe incluir la trayectoria nominal de la misión. La trayectoria definida en la sección 3.2.1.1.1 incluye todos los parámetros necesarios para el guiado del vehículo.

La monitorización del vehículo se realizará a través de un protocolo de comunicación con objeto de aislar el diseño del simulador del vehículo y el diseño del OBC. Por lo tanto, el OBC recibirá los eventos generados por el vehículo a través de un enlace de comunicaciones serie.

El diseño del controlador digital es específico y dependerá del vehículo aeroespacial a guiar. Sin embargo, la señal de entrada que cierra el lazo de control es común a todos los controladores, será el vector de estado del vehículo. El vector de estado del vehículo contiene toda la información necesaria para calcular la señal de control a enviar al vehículo.

La señal de control generada por el controlador digital variará en número de variables de salida. Para resolver este problema definiremos un vector de n variables como señal de control a enviar al vehículo simulado.

Por último, la frecuencia de funcionamiento del controlador digital será 100Hz. Esta frecuencia será la frecuencia mínima exigida al sistema. Una vez desarrollado el sistema y en función del rendimiento del mismo intentaremos conseguir una frecuencia de funcionamiento de 1KHz.

Dicho esto, pasamos a comentar los principales módulos de software que compondrán el subsistema OBC.

- **Control de misión:** El sistema de control de misión es el módulo principal del OBC. El control de misión mantendrá toda la información de la misión. Es el módulo encargado de coordinar el resto de componentes del OBC.
- **Sistema NGC:** El sistema de navegación, guiado y control del vehículo aeroespacial es el módulo de software encargado de implementar el controlador digital. El sistema de guiado y control de nuestro vehículo constará de tres controladores digitales totalmente independientes, una para cada etapa de vuelo.
- **T&T:** El sistema de telemetría y telecomando implementará el sistema de comunicación ente el CCMT y el OBC.

3.2.1.3 Simulador Tiempo Real del Vehículo Aeroespacial

El programa de simulación dinámica del vehículo aeroespacial simulará las trayectorias del vehículo en tiempo real.

El simulador del vehículo incluirá un conjunto de módulos estándar, reutilizables en futuros desarrollos, y una serie de módulos específicos del vehículo concreto a simular.

Los módulos genéricos integrados en el simulador del vehículo serán:

- **Modelo atmosférico:** El modelo atmosférico que integrará el simulador del vehículo será el modelo atmosférico estándar ISA. El modelo atmosférico ISA es un modelo creado por la Organización de Aviación Civil para uso principalmente en navegación aérea.
- **Perturbaciones:** El módulo de perturbaciones externas sobre el vehículo pretende simular la dinámica de la atmósfera introduciendo vientos, ráfagas y turbulencias al simulador.
- **Transformaciones:** El módulo de transformaciones implementará las ecuaciones para realizar transformaciones entre los diferentes sistemas de referencia.
- **INS:** Sistema de navegación inercial. Este módulo simulará el navegador inercial y el sistema GPS instalado en el vehículo real. El navegador inercial proporcionará los ángulos del vehículo y la posición instantánea del vehículo. El módulo introducirá una pequeña perturbación a la posición generada y permitirá transformar las posiciones al *datum* seleccionado por el OBC.

Los módulos específicos del simulador tiempo real del vehículo son:

- **Módulo de comunicaciones:** El módulo de comunicación permitirá el intercambio de información con el OBC. Este módulo proporcionará los datos de inicialización de la misión al vehículo y al entorno, así como la secuencia de eventos y señales de control generados por el OBC.
- **Módulo de dinámica de sólido rígido:** El vehículo simulado con objeto de validar la plataforma de ensayos se modelará como un sólido rígido. No se contemplará el efecto de la deflexión del sistema de control del vector de empuje de la segunda etapa sobre la posición del centro de masas. Las leyes de variación del centro de masas en función del tiempo sólo considerarán el consumo de combustible de los motores cohete y la separación de las etapas. Este módulo implementará los modelos siguientes:
 - Dinámica traslacional de sólido rígido.
 - Dinámica rotacional de sólido rígido.
- **Módulo aerodinámico:** Este módulo implementará el modelo aerodinámico del vehículo en términos de coeficientes de fuerzas y momentos.
- **Módulo de propulsión:** Este módulo generará el vector de empuje de cada etapa en función del tiempo.
- **Actuadores:** El módulo de actuadores pretende simular los diferentes actuadores del vehículo. El vehículo a simular dispone de tres tipos de ac-

tuadores, aletas en la primera etapa, control del vector de empuje en la segunda etapa y el sistema de motores de gas frío de la tercera y última fase de vuelo.

3.2.2 PROTOCOLOS DE COMUNICACIÓN

El objeto de este capítulo es definir los protocolos de comunicación a implementar para interconectar los diferentes subsistemas que componen la plataforma de simulación.

La definición de los protocolos de comunicación incluye la definición de las interfaces entre los subsistemas de la plataforma de simulación.

Para la descripción de los parámetros a intercambiar entre los subsistemas se ha establecido un conjunto de convenciones presentadas en la segunda sección del capítulo.

El capítulo finaliza analizando el ancho de banda requerido por el sistema para intercambiar el volumen de datos definido en los protocolos a la frecuencia requerida por el sistema de control.

3.2.2.1 Interfaces

La Figura 12 muestra las interfaces de comunicación entre los tres subsistemas. Los subsistemas CCMT y OBC se interconectarán utilizando un enlace serie RS-422 y un enlace digital. El número de líneas del enlace digital dependerá del número de comandos a codificar. La interconexión entre los subsistemas OBC y SimRT-VA se realizará mediante un enlace serie RS-422.

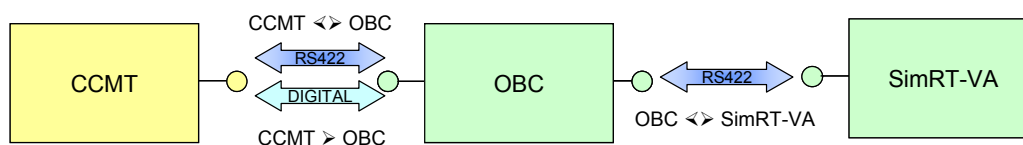


Figura 12: Interfaces de comunicación

3.2.2.2 Nomenclatura de parámetros

Las variables que utilizaremos para describir un parámetro en lo que resta de capítulo será la siguiente:

- Identificador (ID): Identificador del parámetro.
- Descripción (DS): Breve descripción del parámetro.
- Tamaño (T): Número de bits.
- Formato (F): Formato del parámetro.
 - NCF32/NCF64: Número Coma Flotante de 32/64 bits.
 - ECS8/ ECS16/ECS32/ECS64: Entero Con Signo de 8/16/32/64 bits.
 - ESS8/ESS16/ESS32/ESS64: Entero Sin Signo de 8/16/32/64 bits.
- Unidad (U): Unidades en que se expresa el parámetro.

El identificador del parámetro se codificará siguiendo en el siguiente patrón:

AB-C-DES

Donde:

- A: Representa el subsistema de origen del mensaje. Los posibles valores serán:
- B: Representa el subsistema de destino del mensaje.
- C: Representa el tipo de mensaje.
- DES: Indica 3 caracteres descriptivos del contenido del parámetro.

En cuanto al origen o destino del mensaje podemos tener:

- C: **C**entro de Control de Misión Terrestre.
- O: **O**nBoard Computer.
- S: **S**imulador Tiempo Real del Vehículo Aeroespacial.

En cuanto al tipo de mensaje podemos tener:

- C: **C**omando.
- P: **P**etición de información.
- N: **N**otificación.
- D: **D**atos.
- A: **A**cknowledge.

Los parámetros en coma flotante utilizarán el formato definido en el estándar IEEE 754 para su transmisión.

La codificación de los parámetros en las tramas será MSB (*Most Significant Byte*), es decir, el byte más significativo del parámetro se codificará en la primera posición de la trama. Por ejemplo, el valor en coma flotante de precisión simple ó de 32 bits 3.0 se transmitirá $msg[n] = 0x64$, $msg[n+1] = 0x64$ $msg[n+2] = 0x0$ y $msg[n] = 0x0$.

3.2.2.3 Codificación de datos

Este apartado describe la codificación utilizada para identificar los comandos, estados y eventos que se intercambiarán los diferentes subsistemas de la plataforma de simulación.

- **Modo de operación del navegador inercial (INS):**

La codificación de los modos de operación de la INS sirve tanto para la notificación del estado de la misma, en los mensajes enviados por el SimRT-VA al OBC, como para la codificación de los comandos a enviar desde el OBC al SimRT-VA.

La Tabla 8 resume los códigos utilizados para identificar los estados del navegador inercial y los comandos que permiten transitar entre los estados del navegador.

La Figura 13 muestra el diagrama de estados del navegador inercial integrado en el simulador del vehículo.

Modo de operación INS	Código (Hex)	Código (Dec)
Modo Operativo LPO	0x4E	78
Modo Operativo ALIN	0x48	72
Modo Operativo TEST	0x4A	74
Modo Operativo NAV	0x47	71

Tabla 8: Modos de operación INS

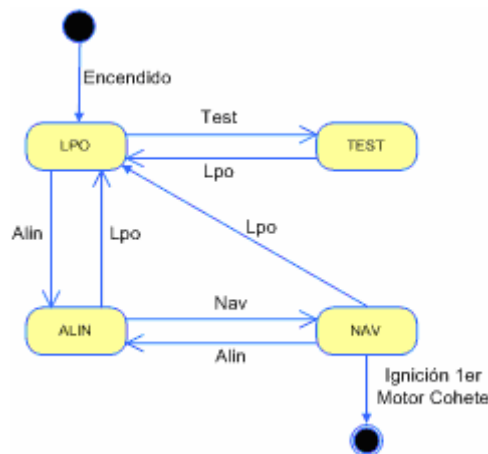


Figura 13: Diagrama de estados INS

• **Comandos:**

La codificación de los comandos correspondientes a la ignición de los motores cohete y la activación del sistema de separación de las interetapas se muestra en la Tabla 9. Además, se ha añadido un comando de autodestrucción del vehículo.

Comando	Código (Hex)	Código (Dec)
Ignición Motor Cohete 1	0xC4	196
Separación Interetapas 1-2	0xDC	220
Ignición Motor Cohete 2	0xE0	224
Separación Interetapas 2-3	0xE3	227
Teledestrucción	0x63	99

Tabla 9: Comandos

• **Fases de vuelo:**

La Tabla 10 muestra la codificación de las fases de vuelo. Las fases de vuelo presentadas en la tabla corresponden a:

- Fase 0: Pre-vuelo. Para transitar a la siguiente fase de vuelo se deberá enviar el comando de ignición del primer motor cohete. Para poder enviar dicho comando se deberá de poner en modo NAV (Navegación) la INS (Ver Figura 13).

- Fase 1: Vehículo propulsado por el primer motor cohete. El vehículo notificará la transición a la fase de vuelo 1 enviado el evento ignición del primer motor cohete.
- Fase 2: Primer motor cohete extinguido. A esta fase se llega tras el evento de fin de combustión del primer motor cohete.
- Fase 3: Vuelo de la segunda etapa después de la separación de la primera etapa y antes de iniciarse el segundo motor cohete. Para alcanzar esta fase de vuelo el sistema de control enviará la orden de separación de la primera etapa. La recepción del evento de detección de la separación de la primera etapa confirmará la transición a la fase de vuelo 3.
- Fase 4: Vehículo propulsado por el segundo motor cohete. El evento de ignición del segundo motor cohete confirmará la transición a la fase de vuelo 4.
- Fase 5: Vuelo de la segunda etapa no propulsada. La transición a esta fase de vuelo está ligada al evento de fin de combustión del segundo motor cohete.
- Fase 6: Vuelo carga útil. Esta fase de vuelo se alcanza tras la recepción del evento de detección de la separación de la segunda etapa.

Comando	Código (Hex)	Código (Dec)
Fase 0	0xA4	164
Fase 1	0xC4	196
Fase 2	0xD4	212
Fase 3	0xDC	220
Fase 4	0xE0	224
Fase 5	0xE2	226
Fase 6	0xE3	227

Tabla 10: Fases de vuelos

- **Sistema de aletas de la 1ª etapa:**

La codificación del sistema de aletas se utilizará para identificar el actuador del vehículo que controla la deflexión de las aletas del vehículo.

El código utilizado será el valor hexadecimal: **0x1E**.

- **Sistema de control del vector de empuje:**

La codificación del sistema del vector de empuje se utilizará para identificar el actuador del vehículo que controla el vector de Empuje.

El código utilizado será el valor hexadecimal: **0x2E**.

- **Motores de gas frío:**

La Tabla 11 define la codificación de los comandos de activación de los motores de gas frío que permiten controlar y guiar el vehículo durante la fase de vuelo 6.

El código identificador del actuador de la última etapa de vuelo es: **0x3E**.

El vehículo dispone de cuatro pares de motores de gas frío separados 90° tal y como muestra la Figura 14.

La codificación utilizada para activar los motores de gas frío relaciona cada bit del byte con un motor. Para activar un motor asignaremos un "1" al bit correspondiente y un "0" para desactivarlo. Esta codificación nos permite activar más de un motor de gas frío utilizando un único byte.

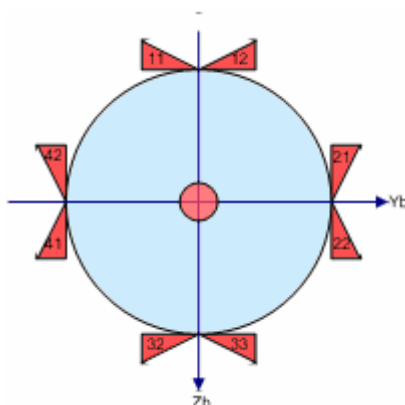


Figura 14: Motores de gas frío

Activación motor	Código (Hex)	Código (Dec)
Activar motor 11	0x01	1
Activar motor 12	0x02	2
Activar motor 21	0x04	3
Activar motor 22	0x08	8
Activar motor 31	0x10	16
Activar motor 32	0x20	32
Activar motor 41	0x40	64
Activar motor 42	0x80	128

Tabla 11: Activación motores de gas frío

3.2.2.4 Comunicación CCMT-OBC

El protocolo de comunicación entre los subsistema CCMT y OBC se describe en este apartado. Dividiremos la descripción en dos secciones de acuerdo a las interfaces físicas entre los dos subsistemas. En primer lugar detallaremos el protocolo que se implementará para intercambiar información a través de las líneas digitales. Finalizaremos con la descripción del protocolo de comunicaciones serie.

3.2.2.4.1 Líneas Digitales

Las líneas digitales estarán asociadas al umbilical que une el vehículo con la rampa de lanzamiento durante los instantes iniciales previos al lanzamiento del mismo. A través de estas líneas se comandarán los diferentes estados del navegador inercial INS (LPO, TEST, ALIN, NAV), se dará la orden de encendido al sistema OBC y se transmitirá la orden efectiva de ignición del primer motor cohete.

Cada comando tendrá asociados 3 bits o líneas digitales, con la siguiente configuración:

101 → Activación de Comando

010 → Desactivación de Comando

La confirmación (*ACK*) de cada comando tendrá una línea digital asociada. El OBC activará la línea correspondiente cuando el comando haya sido ejecutado.

La Tabla 12 muestra las líneas de la tarjeta de E/S digitales utilizadas.

Id.	Descripción	#Bits	Pin	Valor defecto
CO-C-SON	Sistema ON	3	0-2	010
CO-C-SEG	Test seguridad	3	3-5	010
CO-C-ALN	Alineamiento	3	6-8	010
CO-C-LPO	Inicio	3	9-11	010
CO-C-NAV	Navegación	3	12-14	010
CO-C-IGN	Lanzamiento	3	15-17	010
OC-N-SON	Estado Sistema	1	18	0
OC-N-LPO	Estado <i>LPO</i>	1	19	0
OC-N-SEG	Estado <i>TEST</i>	1	20	0
OC-N-ALN	Estado <i>ALIN</i>	1	21	0
OC-N-NAV	Estado <i>NAV</i>	1	22	0

Tabla 12: Información líneas digitales

3.2.2.4.2 Línea serie RS-422 HDLC

Este apartado describe las características del protocolo de comunicación serie a implementar entre los sistemas CCMT y OBC.

La descripción del protocolo presentada en este capítulo corresponde a la capa de aplicación.

La capa física del protocolo de comunicación dependerá de las tarjetas de comunicaciones seleccionadas. El tipo de conector, el pinado del mismo y el cable necesario para interconectar los subsistemas físicamente vendrá impuesto por el modelo de tarjeta, y será el fabricante de las tarjetas de comunicación el que proporcionará la documentación referente a la capa física del protocolo de comunicaciones.

En cuanto a la capa de enlace, el protocolo HDLC (*High-Level Data Link Control*) implementado por las tarjetas de comunicación se encargará del nivel de enlace de datos ofreciendo una comunicación punto a punto fiable entre transmisor y receptor. El protocolo de comunicación HDLC ofrece múltiples configuraciones. La configuración del protocolo para los subsistemas CCMT y OBC se cargará de disco. La configuración final se decidirá en función del rendimiento que ofrezcan las diferentes configuraciones.

Las tramas del protocolo de comunicación a enviar por el subsistema CCMT son:

- **Tele-Destrucción**

La trama Tele-Destrucción se enviará una vez por orden del operador del subsistema CCMT para ordenar la destrucción del vehículo tras detectar algún problema durante la misión.

El formato del campo de datos de la trama se especifica en la Tabla 13.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0x11	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)

Tabla 13: Trama Tele-Destrucción

- **Inicialización**

La trama Inicialización contiene los parámetros de inicialización de la misión. La trama se enviará al subsistema OBC al ejecutar el comando Carga Misión.

El formato del campo de datos de la trama se especifica en la Tabla 14.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0x22	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
CO-D-IHU	Huso	8	BYTE
CO-D-ISF	Esferoide	8	BYTE
CO-D-IPE	Posición Este	32	NCF32 (m)
CO-D-IPN	Posición Norte	32	NCF32 (m)
CO-D-IPA	Posición Altitud	32	NCF32 (m)
CO-D-IPX	Posición X_i	32	NCF32 (m)
CO-D-IPY	Posición Y_i	32	NCF32 (m)
CO-D-IPZ	Posición Z_i	32	NCF32 (m)
CO-D-IVX	Velocidad X_B	32	NCF32 (m/s)
CO-D-IVY	Velocidad Y_B	32	NCF32 (m/s)
CO-D-IVZ	Velocidad Z_B	32	NCF32 (m/s)
CO-D-ITH	Ángulo Θ_i	32	NCF32 (rad)
CO-D-ICH	Ángulo χ_i	32	NCF32 (rad)
CO-D-IPH	Ángulo Φ_i	32	NCF32 (rad)
CO-D-INP	Velocidad angular P_B	32	NCF32 (rad/s)
CO-D-INQ	Velocidad angular Q_B	32	NCF32 (rad/s)
CO-D-INR	Velocidad angular R_B	32	NCF32 (rad/s)
CO-D-E01	Evento E_1	8	NA
CO-D-T01	Tiempo nominal E_1	32	NA
CO-D-W01	Posible variación E_1	32	NA

** El número de eventos puede variar entre misiones. Se defi-*

<i>ne un número máximo de eventos por trama de 25.</i>			
CO-D-EON	Evento E_N	8	NA
CO-D-TON	Tiempo nominal E_N	32	NA
CO-D-WON	Posible variación E_N	32	NA

Tabla 14: Trama Inicialización

- **Puntos de Trayectoria en Rampa**

La trama Puntos de Trayectoria en Rampa contiene varios puntos de la trayectoria nominal correspondiente al tramo de la trayectoria en la rampa de lanzamiento. Esta trama se enviará cuando el operador ejecute, en el subsistema CCMT, el comando correspondiente para cargar los datos de configuración de una misión en el subsistema OBC.

El formato del campo de datos de la trama se especifica en la Tabla 15.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0x33	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
CO-D-TR1	Tiempo relativo a T_0 WP1	32	NCF32 (s)
CO-D-DI1	Distancia WP1	32	NCF32 (m)
CO-D-VE1	Velocidad WP1	32	NCF32 (m/s)
CO-D-AC1	Aceleración WP1	32	NCF32 (m/s^2)
CO-D-EM1	Empuje WP1	32	NCF32 (N)
CO-D-MA1	Masa WP1	32	NCF32 (Kg)
CO-D-RO1	Rozamiento WP1	32	NCF32 (N)
CO-D-RE1	Resistencia 1	32	NCF32 (N)
<i>*El número de puntos de la trayectoria en rampa puede variar entre misiones. Se define un número máximo de puntos por trama de 15.</i>			
CO-D-TRN	Tiempo relativo a T_0 WPN	32	NCF32 (s)
CO-D-DIN	Distancia WPN	32	NCF32 (m)
CO-D-VEN	Velocidad WPN	32	NCF32 (m/s)
CO-D-ACN	Aceleración WPN	32	NCF32 (m/s^2)
CO-D-EMN	Empuje WPN	32	NCF32 (N)
CO-D-MAN	Masa WPN	32	NCF32 (Kg)
CO-D-RON	Rozamiento WPN	32	NCF32 (N)
CO-D-REN	Resistencia WPN	32	NCF32 (N)

Tabla 15: Trama Punto de Trayectoria en Rampa

- **Puntos de Trayectoria Fuera de Rampa**

La trama Puntos de Trayectoria en Fuera de Rampa contiene varios puntos de la trayectoria nominal correspondiente al tramo de la trayectoria fuera de la rampa de lanzamiento. Esta trama se enviará cuando el operador ejecute, en

el subsistema CCMT, el comando correspondiente para cargar los datos de configuración de una misión en el subsistema OBC.

El formato del campo de datos de la trama se especifica en la Tabla 16.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0x44	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
CO-D-OTR	Tiempo relativa a T_0	32	NCF32 (s)
CO-D-OAL	Alcance	32	NCF32 (m)
CO-D-OAL	Altitud	32	NCF32 (m)
CO-D-OVE	Módulo velocidad	32	NCF32 (m)
CO-D-OMS	Masa del vehículo	32	NCF32 (kg)
CO-D-OEM	Empuje	32	NCF32 (N)
CO-D-OAD	Aceleración	32	NCF32 (m/s^2)
CO-D-OMA	Mach	32	NCF32 (NA)
CO-D- OCD	Cd : Coef. Resist. aerodinámica	32	NCF32 (NA)
CO-D-OBE	Beta	32	NCF32 (rad)
CO-D-OAN	Ángulo	32	NCF32 (rad)
CO-D-OPD	Presión dinámica	32	NCF32 (N/cm^2)
CO-D-OTH	Theta	32	NCF32 (rad)
CO-D-OGA	Gamma	32	NCF32 (rad)
CO-D-OAC	Aceleración longitudinal	32	NCF32 (m/s^2)
CO-D-OPX	Posición X_L	32	NCF32 (m)
CO-D-OPY	Posición Y_L	32	NCF32 (m)
CO-D-OPZ	Posición Z_L	32	NCF32 (m)
<i>Se definen un máximo de 6 puntos por trama</i>		576*N	...

Tabla 16: Trama Punto de Trayectoria Fuera de Rampa

- **Alive**

La trama Alive es una trama periódica que se enviará con una frecuencia de 50 Hz, es decir, una trama cada 20 milisegundos. El objetivo de esta trama es comprobar el estado del enlace de comunicaciones.

El subsistema OBC responderá a la trama Alive con el correspondiente ACK.

El formato del campo de datos de la trama se especifica en la Tabla 17.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0x55	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)

Tabla 17: Trama Alive

- **Abortar Misión**

La trama Abortar Misión se enviará una vez por orden del operador del subsistema CCMT para ordenar la cancelación de la misión cargada en el subsistema OBC. Lógicamente, esta trama sólo tendrá sentido antes de ejecutar la misión. Una vez enviada la orden de ignición del primer motor cohete no se podrá enviar una orden para abortar la misión.

El formato del campo de datos de la trama se especifica en la Tabla 18.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0x66	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)

Tabla 18: Trama Abortar Misión

Las tramas del protocolo de comunicación a enviar por el subsistema OBC son:

- **Confirmación Tele-Destrucción**

Esta trama la enviará el subsistema OBC como respuesta a la recepción de la trama Tele-Destrucción.

El formato del campo de datos de la trama se especifica en la Tabla 19.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0x99	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)

Tabla 19: Trama Confirmación Tele-Destrucción

- **Confirmación Inicialización**

Esta trama se enviará al CCMT como respuesta a la recepción de la trama Inicialización.

El formato del campo de datos de la trama se especifica en la Tabla 20.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0xAA	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
OC-A-AIN	ACK Inicialización	8	NA

Tabla 20: Trama Confirmación Inicialización

El campo *OC-A-AIN* de la tabla anterior contiene un código cuyo valor puede ser:

0101 0101: La trama recibida es correcta y ha sido procesada.

1010 1010: La trama recibida es incorrecta y el subsistema *OBC* ha descartado la trama.

- **Confirmación Trayectoria En Rampa**

Esta trama confirma la recepción de una trama Trayectoria en rampa.

El formato del campo de datos de la trama se especifica en la Tabla 21.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0xBB	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
OC-A-ATI	ACK Trayectoria en rampa	8	NA

Tabla 21: Trama Confirmación Trayectoria En Rampa

El campo *OC-A-ATI* de la tabla anterior contiene un código cuyo valor puede ser:

0101 0101: La trama recibida es correcta y ha sido procesada.

1010 1010: La trama recibida es incorrecta y el subsistema *OBC* ha descartado la trama.

- **Confirmación Trayectoria Fuera de Rampa**

Esta trama confirma la recepción de una trama Trayectoria fuera de rampa.

El formato del campo de datos de la trama se especifica en la Tabla 22.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0xCC	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
OC-A-ATO	ACK Trayectoria en rampa	8	NA

Tabla 22: Trama Confirmación Trayectoria Fuera de Rampa

El campo *OC-A-ATO* de la tabla anterior contiene un código cuyo valor puede ser:

0101 0101: La trama recibida es correcta y ha sido procesada.

1010 1010: La trama recibida es incorrecta y el subsistema *OBC* ha descartado la trama.

- **Confirmación Alive**

Esta trama confirma la recepción de una trama Alive.

El formato del campo de datos de la trama se especifica en la Tabla 23.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0XDD	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)

Tabla 23: Trama Confirmación Alive

- **Confirmación Abortar Misión**

Esta trama confirma la recepción de una trama Abortar Misión.

El formato del campo de datos de la trama se especifica en la Tabla 24.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0XEE	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)

Tabla 24: Trama Confirmación Abortar Misión

- **Vector de Estado**

Esta trama contiene la información de estado del vehículo. Es una trama periódica que se enviará al CCMT al finalizar la carga de la misión y se continuará enviando de forma periódica mientras transcurra la misión. La frecuencia de envío de esta trama será 100Hz.

El formato del campo de datos de la trama se especifica en la Tabla 25.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0XFF	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
OC-E-NEV	Número de eventos detectados simultáneamente.	8	NA
OC-E-EI1	Identificador del evento detectado	8	NA
*El número de eventos detectados puede variar.			
OC-E-EIN	Identificador del evento detectado	8	NA
OC-E-COS	Estado comunicación	8	NA

	OBC y SimRT-VA		
OC-D-PSX	Posición X	32	NCF32 (m)
OC-D-PSY	Posición Y	32	NCF32 (m)
OC-D-PSZ	Posición Z	32	NCF32 (m)
OC-D-THE	Ángulo de cabeceo	32	NCF32 (rad)
OC-D-CHI	Ángulo de guiñada	32	NCF32 (rad)
OC-D-PHI	Ángulo de balanceo	32	NCF32 (rad)
OC-D-VEX	Velocidad s. ejes cuerpo X	32	NCF32 (m/s)
OC-D-VEY	Velocidad s. ejes cuerpo Y	32	NCF32 (m/s)
OC-D-VEZ	Velocidad s. ejes cuerpo Z	32	NCF32 (m/s)
OC-D-VEP	Velocidad angular p	32	NCF32 (rad/s)
OC-D-VEQ	Velocidad angular q	32	NCF32 (rad/s)
OC-D-VER	Velocidad angular r	32	NCF32 (rad/s)
OC-D-ACX	Aceleración traslación x	32	NCF32 (m/s ²)
OC-D-ACY	Aceleración traslación y	32	NCF32 (m/s ²)
OC-D-ACZ	Aceleración traslación z	32	NCF32 (m/s ²)
OC-D-ACP	Aceleración angular p $\dot{}$	32	NCF32 (rad/s ²)
OC-D-ACQ	Aceleración angular q $\dot{}$	32	NCF32 (rad/s ²)
OC-D-ACR	Aceleración angular Rdo.	32	NCF32 (rad/s ²)
OC-D-FAS	Fase de vuelo del vehículo	8	NA
OC-D-INS	Estado de la INS	8	NA
OC-D-FLA	Ángulo deflexión aletas (4 aletas; 1ª etapa)	32	NCF32 (rad)
OC-D-TVY	Ángulo deflexión del sistema de control del vector de empuje en el plano Y=0	32	NCF32 (rad)
OC-D-TVZ	Ángulo deflexión del sistema de control del vector de empuje en el plano Z=0	32	NCF32 (rad)
OC-D-TGF	Tobera gas frío	8	NA

Tabla 25: Trama Vector de Estado

Una vez definidas las tramas del protocolo de comunicación entre los subsistemas CCMT y OBC pasamos a describir la secuencia de envío de tramas (Ver Figura 15).

Las tramas Tele-Destrucción y Abortar Misión se enviarán al subsistema OBC al recibir el comando Abortar ejecutado por el usuario del CCMT. La trama Abortar Misión se enviará al OBC si el vehículo se encuentra en la fase de vuelo 0, es decir, antes de iniciar el vuelo. En otro caso, se enviará la trama Tele-Destrucción.

La trama Alive es una trama periódica que permite monitorizar el estado del enlace de comunicación. Esta trama se enviará al OBC desde el arranque del sistema hasta la recepción del comando Cargar Misión en el CCMT. El subsistema OBC responderá con la trama Confirmación Alive, identificada en la Figura 15 con la etiqueta Ack Alive[n].

La ejecución del comando Cargar Misión detendrá el envío de la trama Alive. Seguidamente, el CCMT iniciará el envío de la misión a ejecutar por el vehículo. En primer lugar enviará la trama Inicialización, a la que el OBC responderá con la trama Confirmación Inicialización, a continuación, el CCMT enviará las tramas Puntos de Trayectoria en Rampa y Puntos de Trayectoria Fuera de Rampa, identificadas en la figura con las etiquetas TER y TFR. El número de tramas variará de una misión a otra dependiendo del número de puntos de la trayectoria en rampa y fuera de rampa. El OBC responderá a estas tramas con las correspondientes tramas de confirmación.

Por último, inmediatamente después de finalizar la carga de la misión, el subsistema OBC iniciará el envío periódico de la trama Vector de Estado (VE en la figura).

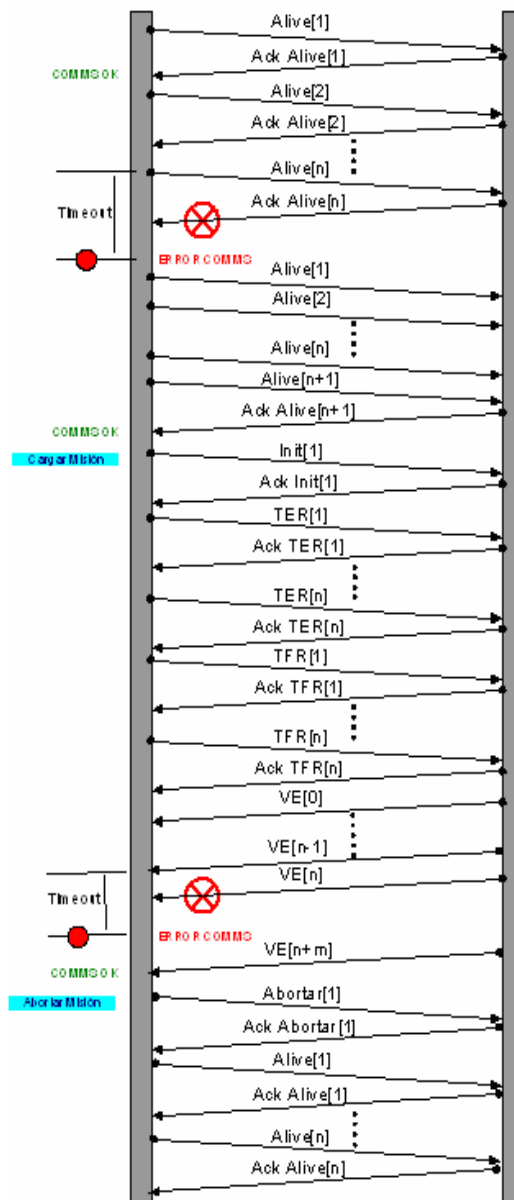


Figura 15: Intercambio de tramas CCMT-OBC

3.2.2.5 Comunicación OBC-SimRT-VA

En este apartado describiremos el protocolo de comunicación entre los subsistemas OBC y SimRT-VA.

La interconexión física entre los dos subsistema se realizará a través de un enlace serie RS-422. El protocolo de la capa de enlace será HDLC.

La capa física del protocolo de comunicación vendrá impuesta por el modelo de las tarjetas de comunicaciones utilizadas.

La implementación del protocolo de comunicación HDLC del nivel de enlace, según el modelo OSI, estará integrado en las tarjetas de comunicación.

La configuración del hardware y del modo de funcionamiento del protocolo HDLC se decidirá durante la fase de codificación basándonos en el rendimiento ofrecido por las tarjetas en los diferentes modos de operación.

A continuación, definiremos el protocolo de comunicaciones del nivel de aplicación. Identificaremos las tramas que intercambiarán los subsistemas OBC y SimRT-VA, describiremos el formato de las mismas y explicaremos la secuencia de envío de las tramas.

Para el intercambio de información definiremos las siguientes tramas:

- **Inicialización:**

La trama Inicialización contiene los parámetros iniciales de la misión. Es una trama periódica que se enviará con una frecuencia de 100Hz hasta que se ejecute el comando de ignición del primer motor cohete.

El formato de la trama se define en la Tabla 26.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0X50	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
OS-D-IPX	Posición inicial X_I	32	NCF32 (m)
OS-D-IPY	Posición inicial Y_I	32	NCF32 (m)
OS-D-IPZ	Posición inicial Z_I	32	NCF32 (m)
OS-D-IVX	Velocidad inicial X_B	32	NCF32 (m/s)
OS-D-IVY	Velocidad inicial Y_B	32	NCF32 (m/s)
OS-D-IVZ	Velocidad inicial Z_B	32	NCF32 (m/s)
OS-D-ITH	Ángulo inicial Θ_I	32	NCF32 (rad)
OS-D-ICH	Ángul inicial χ_I	32	NCF32 (rad)
OS-D-IPH	Ángulos inicial Φ_I	32	NCF32 (rad)
OS-D-INP	Velocidad angular inicial P_B	32	NCF32 (rad/s)
OS-D-INQ	Velocidad angular inicial Q_B	32	NCF32 (rad/s)
OS-D-INR	Velocidad angular inicial R_B	32	NCF32 (rad/s)

Tabla 26: Trama Inicialización

- **Comando:**

La trama Comando la enviará el subsistema OBC. La trama contiene una orden a ejecutar por el vehículo. Por ejemplo, modificar el estado del navegador inercial INS, activar el motor cohete ó el sistema de separación interetapas, etc.

El formato de la trama se especifica en la Tabla 27.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0X60	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
OS-C-CMD	Identificador co- mando	8	NA

Tabla 27: Trama Comandos

• **Señal de Control:**

La trama Señal de Control contiene la información calculada por el sistema de guiado y control. Es una señal dirigida a uno de los actuadores del vehículo. El número de parámetros dependerá del tipo de actuador.

Esta trama se enviará periódicamente al subsistema SimRT-VA. La frecuencia de envío será 100Hz.

El formato de la trama se especifica en la Tabla 28.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0X70	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
OS-C-IDA	Identificador actua- dor	8	NA
<i>*El identificador de actuador del campo anterior identifica el número de parámetros necesarios.</i>			
OS-C-PA1	Parámetro 1	X	NA
<i>*El numero de parámetros y el formato de los mismos (#bits+formato) dependerá del tipo de actuador. Por ejemplo:</i>			
<ul style="list-style-type: none"> • <i>Aletas: 1 parámetro - NFC32 - rad.</i> • <i>Control Vector de Empuje: 2 parámetros - NFC32 x 2 - rad.</i> • <i>Motores gas frío: 1 parámetro - 8 bits - NA</i> 			
OS-C-PAN	Parámetro N	X	NA

Tabla 28: Trama Señal de Control

• **Datos:**

La trama Datos es la única trama enviada por el subsistema SimRT-VA. La trama se enviará con una frecuencia de 100Hz.

La información que contiene la trama es el vector de estado, citado repetidas veces a lo largo del documento.

El formato de la trama se especifica en la Tabla 29.

Id.	Descripción	#Bits	Formato
ID_TRAMA	0X80	8	NA
RESERVADO	Reservado	8	NA
TIME_STAMP	Tiempo UTC	64	NCF64 (ms)
SO-D-E01	Detección fin combustión 1ª etapa	1	NA
SO-D-E02	Detección fin combustión 2ª etapa	1	NA
SO-D-EXX	Reservado	6	NA
SO-D-PPX	Posición X_0	32	NFC32 (m)
SO-D-PPY	Posición Y_0	32	NFC32 (m)
SO-D-PPZ	Posición Z_0	32	NFC32 (m)
SO-D-PTH	Ángulo de cabeceo	32	NFC32 (rad)
SO-D-PCH	Ángulo de guiñada	32	NFC32 (rad)
SO-D-PPH	Ángulo de balanceo	32	NFC32 (rad)
OS-D-VLX	Velocidad X_0	32	NFC32 (m/s)
OS-D-VLY	Velocidad Y_0	32	NFC32 (m/s)
OS-D-VLZ	Velocidad Z_0	32	NFC32 (m/s)
OS-D-VLP	Velocidad angular p	32	NFC32 (rad/s)
OS-D-VLQ	Velocidad angular q	32	NFC32 (rad/s)
OS-D-VLR	Velocidad angular r	32	NFC32 (rad/s)
OS-D-ACX	Aceleración traslación X	32	NFC32 (m/s ²)
OS-D-ACY	Aceleración traslación Y	32	NFC32 (m/s ²)
OS-D-ACZ	Aceleración traslación Z	32	NFC32 (m/s ²)
DN-D-AAP	Aceleración angular pdot	32	NFC32 (rad/s ²)
DN-D-AAQ	Aceleración angular qdot	32	NFC32 (rad/s ²)
DN-D-AAR	Aceleración angular rdot.	32	NFC32 (rad/s ²)
DN-D-FLA	Estado ángulo deflexión aletas	32	NFC32 (rad)
DN-D-VEY	Estado Ángulo deflexión del vector de empuje en plano Y=0	32	NFC32 (rad)
DN-D-VEZ	Estado Ángulo deflexión del vector de empuje en plano Z=0	32	NFC32 (rad)
DN-D-MGF	Estado motores gas frío	8	NA
DN-D-INS	Estado de la INS	8	NA
DN-D-FAS	Fase de vuelo del vehículo	8	NA

Tabla 29: Trama Datos

La Figura 16 resume el intercambio de tramas entre los subsistema OBC y SimRT-VA.

La secuencia de intercambio de tramas se inicia con el envío de la trama Inicialización desde el OBC. La recepción de la trama en el otro extremo indicará que el enlace de comunicación funciona correctamente y el SimRT-VA responderá con la trama Datos. Este intercambio de tramas se mantendrá hasta que el OBC envíe el comando de ignición del primer motor cohete. A partir de este momento, el OBC enviará la trama Señal de Control.

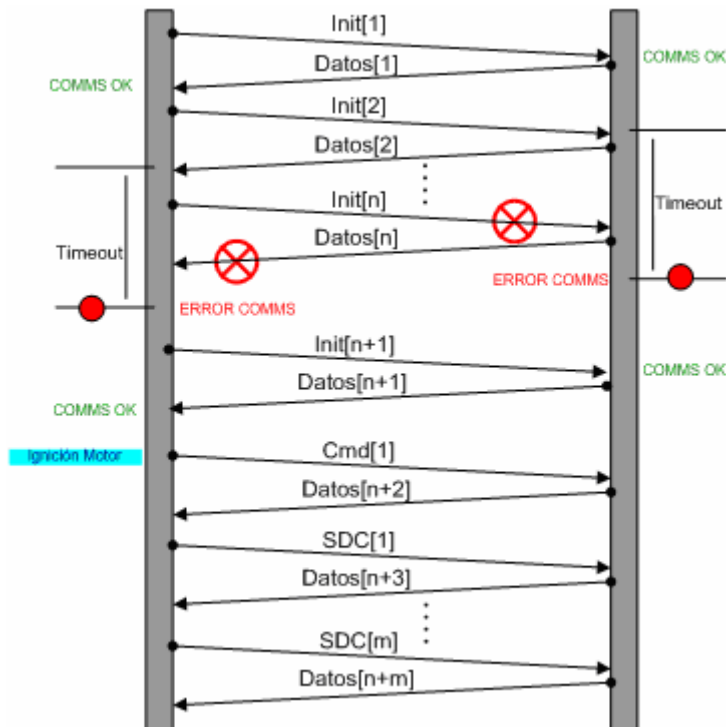


Figura 16: Intercambio de tramas OBC-SimRT-VA

3.2.2.6 Ancho de banda

El ancho de banda requerido para las tarjetas de comunicación serie RS-422 HDLC se calcula a continuación.

- La frecuencia de envío definida en los dos enlaces de comunicación, CCMT - OBC y OBC - SimRT-VA es de 100 Hz. Por lo tanto, se enviará una trama cada 10 ms.
- El tamaño máximo de una trama del protocolo de comunicación CCMT - OBC es de 1480 bits (Vector de Estado).
- El tamaño máximo de una trama del protocolo de comunicación OBC-SimRT-VA es de 1456 bits (Datos).
- El *overhead* introducido por el protocolo *HDLC* en la capa de enlace será inferior a 72 bits (Ver Tabla 30).
- El canal de comunicación *RS-422* es *full-duplex*.

Por lo tanto, el protocolo requiere un ancho de banda superior a 155200 bps para el enlace CCMT – OBC y 152800 bps para el enlace OBC – SimRT-VA.

Identificador de campo	Tamaño (en bits)
Flag (inicio de trama)	8 bits
Dirección	8 bits
Control	8 ó 16 bits
Información	Datos (variable)
Checksum	16 ó 32 bits
Flag (final de trama)	8 bits

Tabla 30: Trama protocolo HDLC

3.3 ANÁLISIS ECONÓMICO

La valoración económica considera los siguientes puntos:

- Coste de los recursos humanos.
- Coste de las licencias de software.
- Coste del hardware.

El coste de los recursos humanos implicados en el desarrollo del sistema se detalla en la Tabla 31.

La Tabla 31 muestra los roles identificados en el apartado 2.1 Organización general.

- **RP:** Responsable de Proyecto.
- **RC:** Responsable de Calidad.
- **RT:** Responsable Técnico.
- **AF:** Analista Funcional.
- **IP:** Ingeniero de Proyecto.
- **IC:** Ingeniero de Control

Para cada rol se detalla:

- El número de **recursos** requeridos.
- El número de **días/horas** estimadas según la planificación realizada. (Ver 2.5 Planificación).
- El **CPH** (Coste de Producción por Hora).
- El **CH** (Coste por Hora). $CH = CPH + CPH * 1.15$.
- El **CP** (Coste de Producción): $CP = Horas * CPH$.
- El **Coste**. Coste = Horas * CH.
- El coste de producción y de venta **total**.

El CPH es el coste que le representa a la empresa destinar un recurso determinado al proyecto. El CPH incluye el salario del trabajador más el *overhead* de la empresa. Por lo general, el departamento financiero será el encargado de calcular el CPH.

El CH representa el precio de la hora de ingeniería vendida por la empresa. Para calcular el CH se ha incrementado el CPH en un 115%. El objetivo es doblar el precio de la hora de producción. El 15% restante se ha incluido para absorber posibles desviaciones en la planificación del proyecto.

Rol	Recursos	Días	Horas	CPH (€)	CH (€)	CP (€)	Coste (€)
RP	1,0	25,0	200,0	32,00	68,80	6.400,00	13.760,00
RC	1,0	10,0	80,0	26,00	55,90	2.080,00	4.472,00
RT	1,0	60,0	480,0	26,00	55,90	12.480,00	26.832,00
AF	2,0	125,0	1000,0	22,00	47,30	22.000,00	47.300,00
IP	2,0	125,0	1000,0	18,00	38,70	18.000,00	38.700,00
IC	1,0	125,0	1000,0	18,00	38,70	18.000,00	38.700,00
Total						78.960,00	169.764,00

Tabla 31: Valoración coste RRHH

La Tabla 32 detalla el coste de las licencias software requeridas para el desarrollo y despliegue del proyecto. La información mostrada en la tabla es la siguiente:

- El **producto** software requerido.
- La **empresa** desarrolladora del producto.
- El **PL** (Precio por Licencia).
- El **número de licencias** necesarias.
- El **coste** total por producto.
- El **coste total** en licencias.

El tipo de licencia de los productos requeridos puede ser compartida ó individual (*node-locked*). El precio de las licencias detalladas en la Tabla 32 corresponde a licencias de tipo individual.

El precio por licencia mostrado en la Tabla 32 corresponde al precio ofertado por las diferentes empresas en enero de 2008.

Producto	Empresa	PL (€)	#Lic.	Coste (€)
MATLAB	THE MATHWORKS	500	2	1000
MATLAB, Simulink	THE MATHWORKS	1000	2	2000
MATLAB, Simulink Control Design	THE MATHWORKS	200	1	200
MATLAB, Control System Toolbox	THE MATHWORKS	200	1	200
MATLAB, Fuzzy Logia Toolbox	THE MATHWORKS	200	1	200
MATLAB, Aerospace Toolbox	THE MATHWORKS	200	1	200
MATLAB, Aerospace Blockset	THE MATHWORKS	200	1	200

MATLAB, Compiler	THE MATHWORKS	500	1	500
MATLAB, xPC Target	THE MATHWORKS	500	1	500
JamaicaVM (JavaRT)	AICAS	4500	1	4500
JClass	QUEST SOFTWARE	1500	1	1500
FuzzyJ Toolkit	NRC of Canada	0	1	0
			Total	11000

Tabla 32: Valoración coste licencias SW

La Tabla 33 detalla la valoración económica en lo referente al material hardware necesario para instalar la plataforma de simulación. La información mostrada en la tabla es la siguiente:

- La **descripción** del material hardware requerido.
- La **empresa** fabricante del producto.
- El **PU** (Precio por Unidad).
- El **número de unidades** requeridas.
- El **coste** por elemento hardware.
- El **coste total** del hardware.

Descripción	Empresa	PU (€)	Unid	Coste (€)
HP XW6200 WS 2.80GHz 1GB/80GB	HP	1400	3	4200
Monitor LCD HP VS19b 19"	HP	280	1	280
DIO Card PCI-7296/7248/7224	ADLINK	500	2	1000
SYNCLINK WAN Adapters For Linux GT4 2 Ports	MICROGATE	644	3	1932
Cable RS-530 Null Modem	--		2	0
			Total	7412

Tabla 33: Valoración coste material HW

El coste total del desarrollo de la plataforma de simulación es: **188.176,00 €**.

El beneficio estimado es: **90.804,00 €**

4 Especificación

El objeto del presente capítulo es definir, de manera explícita y formal, todos los requisitos que debe satisfacer la plataforma de simulación de vuelos aeroespaciales.

El capítulo se estructura entorno a dos apartados.

El primer apartado introduce los casos de uso del sistema. La definición de los casos de uso facilitará la comprensión de los diferentes escenarios que debe contemplar el sistema informático.

El segundo apartado presenta la especificación formal de requisitos. La especificación de requisitos constituirá la base de trabajo para las próximas etapas de desarrollo.

4.1 CASOS DE USO

Los casos de uso de la plataforma de ensayos de vehículos aeroespaciales se describen en este apartado.

4.1.1 CASO DE USO: [CU_001] INICIAR SISTEMA

El propósito del caso de uso Iniciar Sistema es dotar al Operador del CCMT de los mecanismos necesarios para arrancar el sistema. El caso de uso se descompone en dos funciones principales: Cargar Configuración e Iniciar Controlador de Misión en Tierra.

El caso de uso Cargar Configuración cargará en memoria los parámetros de configuración del CCMT.

El objetivo del caso de uso Iniciar CMT es crear e inicializar los componentes software que configuran el CCMT, establecer la comunicación con el OBC y crear la conexión con la base de datos del sistema.

Los actores que intervienen son: Operador CCMT, BBDD y OBC.

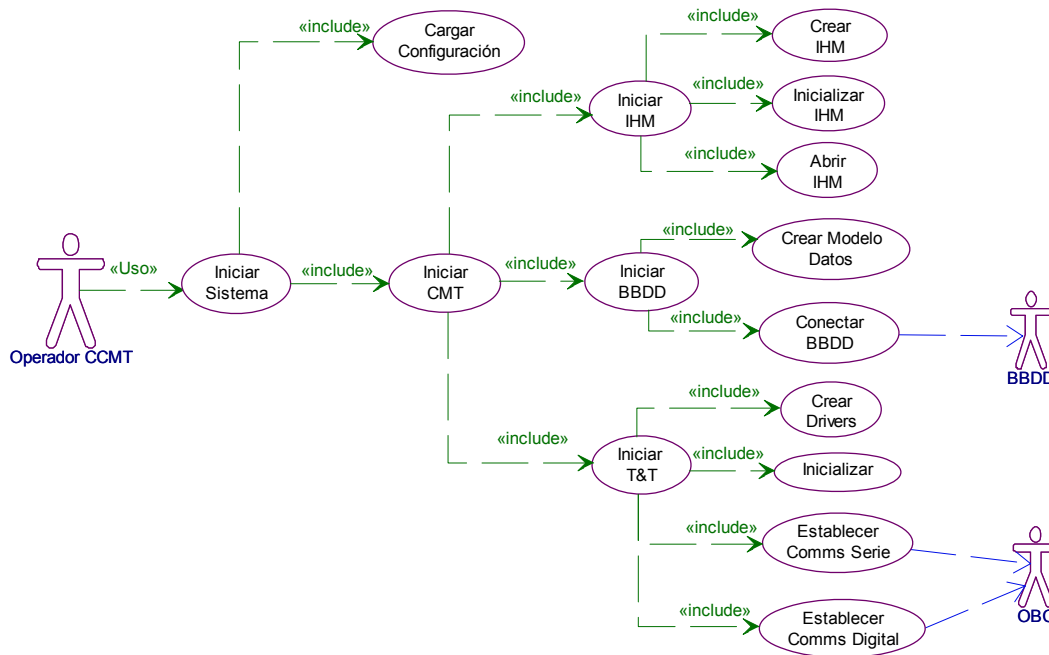


Figura 17: Caso de uso: [CU_001] Iniciar Sistema

4.1.2 CASO DE USO: [CU_010] EDITAR MISIÓN

El caso de uso Editar Misión es un caso de uso genérico que se especializa con los casos de uso Crear Misión, Modificar Misión y Eliminar Misión.

La edición de una misión implicará obtener la lista de misiones de la base de datos.

El caso de uso Crear Misión creará una misión en blanco. El operador deberá introducir todos los parámetros que configuran una misión y deberá guardar la misión creada.

El caso de uso Modificar Misión obtendrá la lista de misiones de la base de datos del sistema. El operador del CCMT deberá seleccionar la misión a editar, actualizará los parámetros de la misión objeto de modificación y guardará la misión modificada.

El caso de uso Eliminar Misión obtendrá la lista de misiones de la base de datos del sistema. El operador del CCMT deberá seleccionar la misión a eliminar y ordenará la eliminación de la misma. El sistema solicitará confirmación antes de borrar la misión de la base de datos del sistema.

Los actores que intervienen son: Operador CCMT y BBDD.

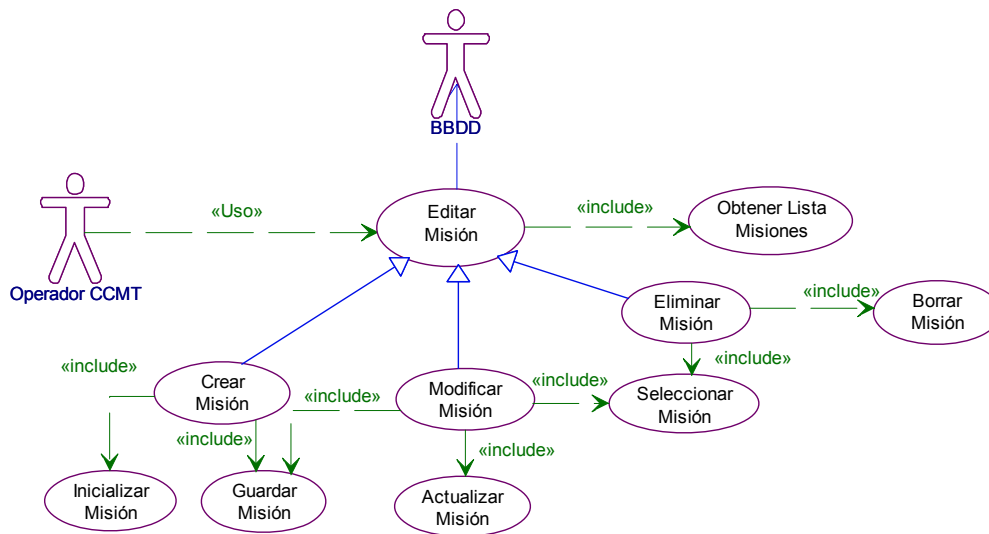


Figura 18: Caso de uso: [CU_010] Editar Misión

4.1.3 CASO DE USO: [CU_020] CARGAR MISIÓN

El caso de uso Cargar Misión carga la misión a ejecutar en el subsistema OBC.

La carga de una misión en el OBC implica varias operaciones. En primer lugar, se deberá obtener la lista de misiones de la base de datos del sistema. A continuación, el operador del CCMT deberá seleccionar la misión a cargar de la lista de misiones y deberá introducir las condiciones iniciales de la misión. Por último, el operador del CCMT ordenará cargar la misión en el OBC.

El caso de uso Cargar Misión OBC se descompone en dos funciones principales, en primer lugar, se leerá la trayectoria nominal almacenada en un fichero Excel y a continuación se enviará la misión al subsistema OBC.

La lectura de la trayectoria nominal cargará en memoria los eventos a ejecutar a lo largo de la misión, la trayectoria en rampa y la trayectoria fuera de rampa.

El envío de la misión al subsistema OBC consiste en enviar las condiciones iniciales de la misión, la parte de la trayectoria nominal del vehículo en la rampa de lanzamiento y la trayectoria nominal fuera de la rampa de lanzamiento al subsistema OBC.

Los actores que intervienen son: Operador CCMT, OBC, BBDD y MS Excel.

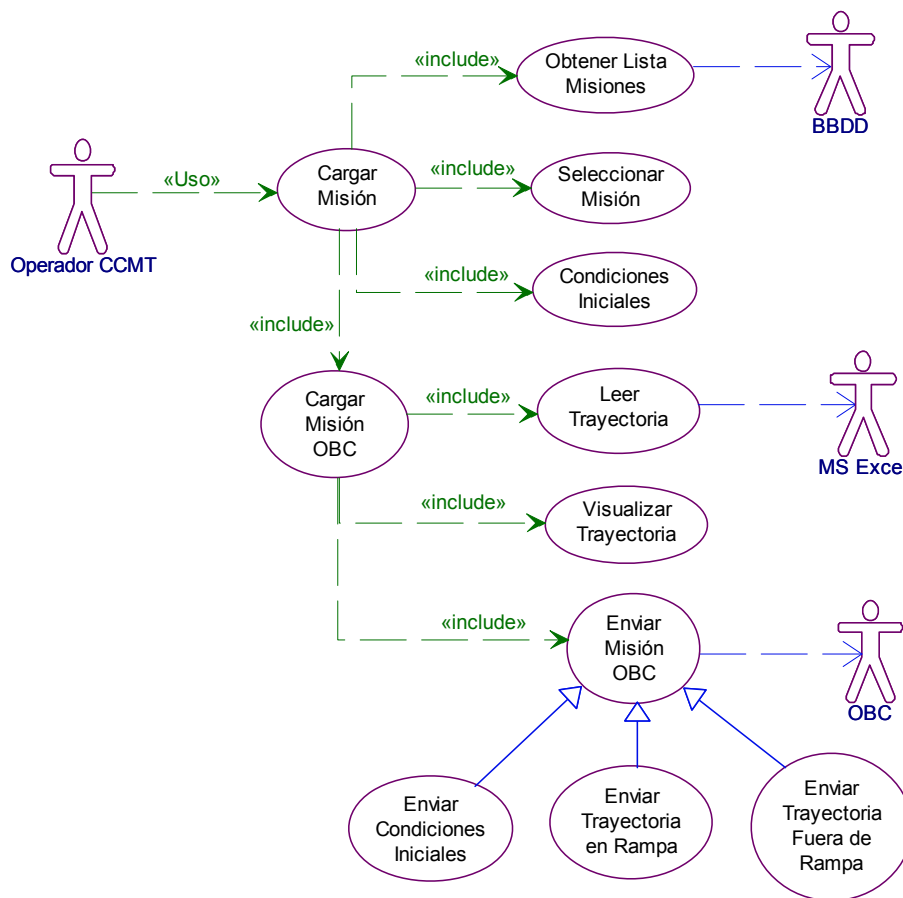


Figura 19: Caso de uso: [CU_020] Cargar Misión

4.1.4 CASO DE USO: [CU_030] EJECUTAR MISIÓN

El caso de uso Ejecutar Misión mostrado en la Figura 20 describe las acciones requeridas para ejecutar una misión.

En primer lugar, el operador del CCMT deberá cargar una misión. El caso de uso Cargar Misión se ha descrito en el apartado anterior.

Al finalizar la carga de la misión el operador del CCMT deberá ejecutar la secuencia test. El chequeo de los subsistemas incluye el test de comunicaciones entre los subsistemas OBC - SimRT-VA y entre los subsistemas CCMT - OBC, el test del navegador inercial y el test de los subsistemas del vehículo.

Una vez finalizada la secuencia de testeo, el operador del CCMT deberá ejecutar los comandos necesarios para configurar el navegador inercial en modo NAV (Navegación). El diagrama de estados presentado en la Figura 13 describe la secuencia de comandos a ejecutar para configurar la INS en modo navegación. Los comandos que deberá ejecutar el operador del CCMT son: Alinear INS y Modo Navegación.

Una vez configurada la INS en modo navegación el sistema estará preparado para ejecutar la misión.

Para iniciar la ejecución de la misión el operador del CCMT deberá ejecutar el comando Iniciar Misión.

El comando Iniciar Misión iniciará la cuenta atrás. Al finalizar la cuenta atrás se enviará la orden de ignición del primer motor cohete al subsistema OBC. El subsistema OBC ordenará al SimRT-VA la ejecución de la simulación a la vez que iniciará la monitorización del vehículo. El sistema de navegación, guiado y control integrado en el OBC entrará en funcionamiento. El subsistema OBC enviará el vector de estado al subsistema CCMT.

Una vez iniciada la misión, el subsistema CCMT activará la monitorización de la misión. La monitorización de la misión permitirá visualizar el vector de estado instantáneo y graficar la trayectoria real del vehículo.

Al finalizar la misión el subsistema CCMT permitirá guardar el histórico de la misión en un fichero.

Los actores que intervienen son: Operador CCMT, OBC, SimRT-VA y CCMT.

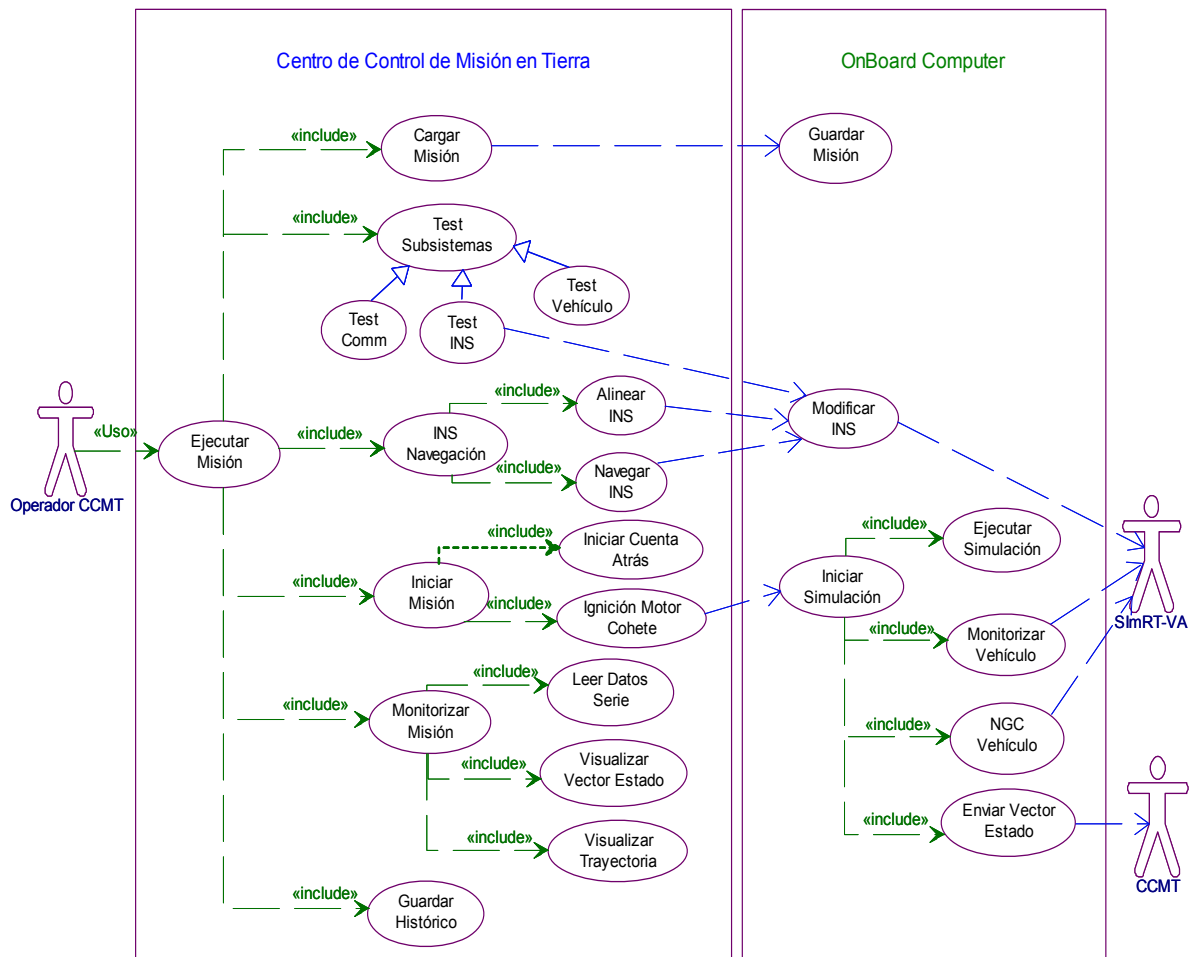


Figura 20: Caso de uso: [CU_030] Ejecutar Misión

4.1.5 CASO DE USO: [CU_040] CANCELAR MISIÓN

El caso de uso Cancelar Misión permitirá cancelar una misión cargada. Para cancelar una misión el operador del CCMT deberá haber cargado una misión en el sistema. La ejecución del comando Cancelar Misión enviará la orden al subsistema OBC para reseteará el *buffer* que almacena la misión.

Los actores que intervienen son: Operador CCMT y OBC.

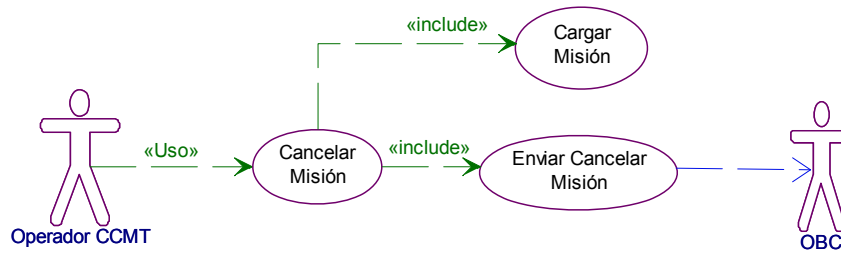


Figura 21: Caso de uso: [CU_040] Cancelar Misión

4.1.6 CASO DE USO: [CU_050] ABORTAR MISIÓN

El caso de uso Abortar Misión permitirá cancelar una misión una vez ejecutada.

La ejecución de este caso de uso requiere que el operador del CCMT haya cargado y ejecutado una misión previamente.

La ejecución del comando Abortar Misión enviará el comando correspondiente al subsistema OBC. El subsistema OBC ordenará la destrucción del vehículo al recibir la orden de abortar misión.

Los actores que intervienen son: Operador CCMT, OBC y SimRT-VA.

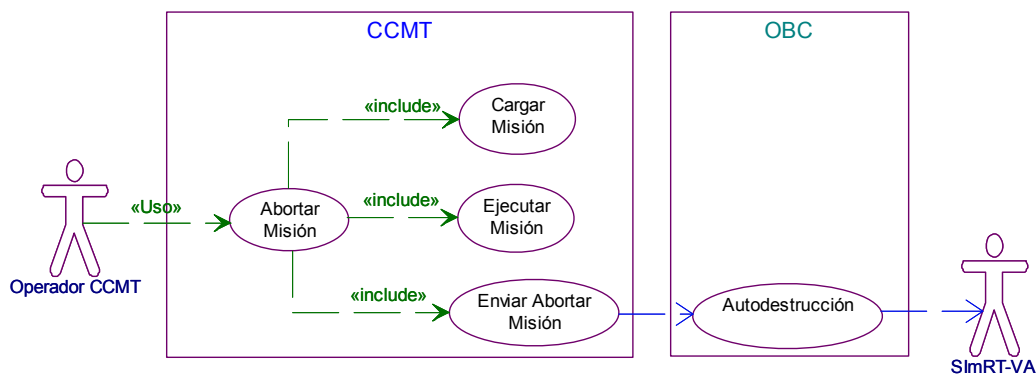


Figura 22: Caso de uso: [CU_050] Abortar Misión

4.1.7 CASO DE USO: [CU_060] GUARDAR HISTÓRICO

El caso de uso Guardar Histórico permitirá guardar el histórico de una misión en un fichero.

El operador del CCMT deberá ejecutar el comando guardar histórico de misión. El subsistema CCMT mostrará la lista de misiones ejecutadas almacenadas en la base de datos del sistema.

El operador del CCMT deberá seleccionar una misión de la lista de misiones ejecutadas y ordenar al sistema que guarde la misión seleccionada.

El subsistema CCMT solicitará al usuario el directorio del sistema de ficheros para guardar el fichero con la misión.

Una vez introducido el directorio, el subsistema CCMT generará automáticamente el nombre del fichero a partir de la información contenida en la base de datos, leerá el histórico de la misión de la base de datos y lo escribirá en el fichero.

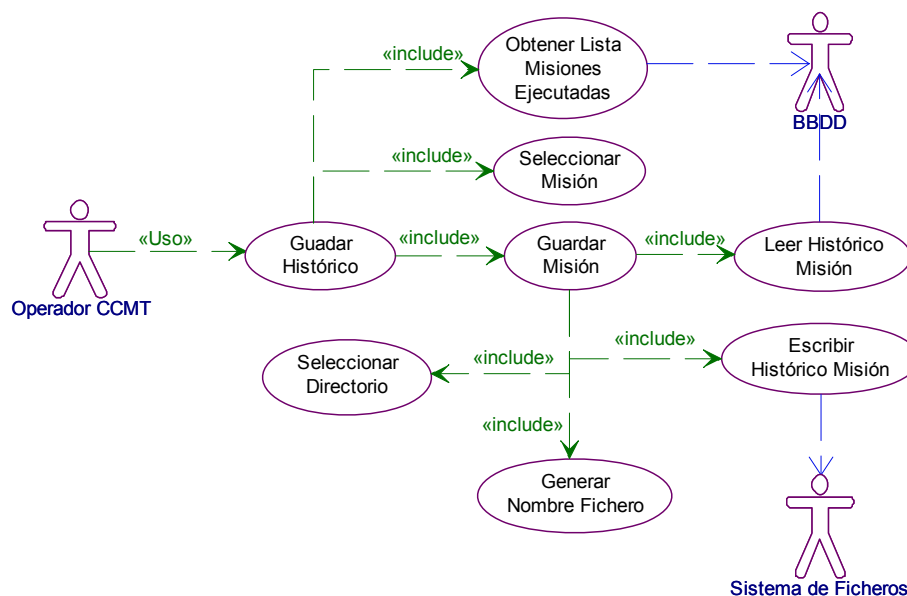


Figura 23: Caso de uso: [CU_060] Guardar Histórico

4.2 ESPECIFICACIÓN DE REQUISITOS

La definición de requisitos presentada en este capítulo utilizará las palabras **debe** o **deberá** y **debería** para definir los requisitos que aplican al sistema objeto de desarrollo. **Debe** o **deberá** especificará un requisito mandatorio. La aplicación deberá satisfacer los requisitos especificados con las palabras **debe** o **deberá**. **Debería** expresará un requisito no mandatorio pero recomendable, si aparece un requisito descrito con la palabra **debería** indicará que el requisito podría ser mandatorio en una futura versión del documento.

La especificación de cada requisito incluirá un atributo para identificar el criterio de validación. Los métodos de validación disponibles son:

- **Test:** Verificación del requisito mediante la ejecución de casos formales de Test. Un caso de test queda definido por un conjunto determinado de entradas, un entorno de test, un procedimiento de test y un conjunto de salidas ó resultados esperados. El test es totalmente determinista y reproducible.
- **Análisis:** Verificación del requisito mediante procesado y análisis de datos obtenidos de otros métodos de verificación. Ejemplos de análisis son reducción, interpretación y extrapolación de resultados de test.
- **Inspección:** Verificación del requisito mediante un examen detallado, por ejemplo, del diseño, de la implementación de algún algoritmo, etc. También se incluye inspección de código (e.j. requisitos de estándares de programación) ó de la documentación (e.j. requisitos de documentación).
- **Demostración:** Demostración por ejecución en un entorno real o simulado del cumplimiento del requisito. Aplicable principalmente al test/demostración de interfaces gráficos y escenarios de utilización.

Cada requisito se identificará por un código con el siguiente formato **XX-YYY-ZZZZ**, donde **XX** es un identificador de tipo numérico que identifica el tipo de requisito, **YYY** es un código utilizado para identificar el subsistema al que aplica el requisito. Por último **ZZZZ** es un identificador numérico utilizado para enumerar los requisitos dentro del tipo y subsistema al que pertenecen:

- **Tipo:** Los requisitos se englobarán en una de las siguientes categorías:

<p>01 Requisito funcional</p> <p>02 Requisito de rendimiento</p> <p>03 Requisito de interfaz</p> <p>04 Requisito operacional</p> <p>05 Requisito de recursos</p> <p>06 Requisito de verificación</p> <p>07 Requisito de documentación</p> <p>08 Requisito de protección</p> <p>09 Requisito de portabilidad</p>	<p>10 Requisito de calidad</p> <p>11 Requisito de fiabilidad</p> <p>12 Requisito de mantenibilidad</p> <p>13 Requisito de seguridad</p> <p>14 Requisito de disponibilidad</p> <p>15 Requisito de diseño</p> <p>16 Requisito de desarrollo</p> <p>17 Requisito de formación</p> <p>18 Otro tipo de requisitos</p>
--	---
- **Subsistema:** Los subsistema para los que se especificarán requisitos son:
 - GEN** Sistema General
 - CMT** Centro de Control de Misión en Tierra
 - OBC** OnBoard Computer
 - SIM** Simulador Tiempo Real Vehículo Aeroespacial

La tabla siguiente presenta la plantilla seleccionada para la especificación de un requisito:

<Identificador> <Título>

Descripción:

Criterio de Validación: Test/Demostración/Análisis/Inspección

Tabla 34: Especificación de un requisito

4.2.1 REQUISITOS FUNCIONALES

01-GEN-0001 Crear misión

Descripción: El sistema debe permitir crear una misión nueva.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0010 Guardar misión

Descripción: El sistema debe permitir guardar una misión.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0020 Parámetros de misión

Descripción: Los parámetros que definen una misión deben ser:

- Identificador.
- Descripción.
- Posición de lanzamiento: Norte, Este, Altitud, Huso y Esferoide.
- Fecha y hora.
- Carga (masa en kg).
- Trayectoria nominal. Este campo contiene el nombre del fichero Excel que contiene la trayectoria de referencia.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0030 Modificar misión

Descripción: El sistema debe permitir modificar los parámetros de una misión guardada previamente.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0040 Eliminar misión

Descripción: El sistema debe permitir eliminar una misión guardada previamente.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0050 Lista de misiones

Descripción: El sistema debe permitir consultar la lista de misiones guardadas en el sistema.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0060 Cargar misión

Descripción: El sistema debe permitir cargar una misión para su posterior ejecución.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0070 Ejecutar misión

Descripción: El sistema debe permitir ejecutar una misión cargada previamente en el sistema.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0090 Test subsistemas

Descripción: El sistema debe permitir chequear el estado de los subsistemas.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0100 Modos navegador inercial

Descripción: El sistema debe permitir modificar los modos de operación del navegador inercial integrado en el vehículo aeroespacial.

Los modos de operación del navegador inercial deben ser:

- LPO: Estado inicial.
- TEST: Modo test.
- ALIN: Modo alineación.
- NAV: Modo navegación.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0110 Cuenta atrás

Descripción: El sistema debe ejecutar la cuenta atrás al ejecutar una misión.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0120 Guarda histórico de misión

Descripción: El sistema debe guardar el histórico de una misión.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0130 Exportar histórico de misión

Descripción: El sistema debe permitir exportar el histórico de una misión ejecutada.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0140 Generación nombre fichero histórico

Descripción: El sistema debe generar de forma automática el nombre de los ficheros que contienen una misión exportada.

El formato del nombre del fichero será el siguiente:

Nombre Misión + "_" + AA_MM_DD + "_" + HH_MM_SS.analog

Nombre Misión + "_" + AA_MM_DD + "_" + HH_MM_SS.event

AA_MM_DD: Fecha local del PC

HH_MM_SS: Hora local del PC

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0150 Cancelar misión

Descripción: El sistema debe permitir cancelar una misión cargada pre-

viamente.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0160 Abortar misión

Descripción: El sistema debe permitir abortar una misión en ejecución.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0170 Estado comunicaciones

Descripción: El sistema debe notificar el estado de los enlaces de comunicación siguientes:

- CCMT – OBC (Serie)
- CCMT – OBC (Digital)
- OBC – SimRT-VA (Serie)

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0180 Fase de vuelo

Descripción: El sistema debe notificar la fase de vuelo de la misión.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-GEN-0190 Eventos de Misión

Descripción: El sistema debe notificar los eventos generados durante una misión.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-OBC-0001 Navegación, guiado y control

Descripción: El subsistema OBC debe implementar un controlador digital capaz de guiar el vehículo aeroespacial según una trayectoria predefinida.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0001 Modelo atmosférico

Descripción: El subsistema SimRT-VA debe integrar el modelo atmosférico estándar ISA.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0010 Simulador INS

Descripción: El subsistema SimRT-VA debe simular el funcionamiento de un navegador inercial.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0020 Simulador perturbaciones

Descripción: El subsistema SimRT-VA debe simular las perturbaciones producidas por la atmósfera sobre el vehículo.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0030 Dinámica del vehículo

Descripción: El subsistema SimRT-VA debe modelar la dinámica tran-

sacional y rotacional del vehículo aeroespacial.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0040 Aerodinámica del vehículo

Descripción: El subsistema SimRT-VA debe modelar la aerodinámica del vehículo aeroespacial.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0050 Propulsión del vehículo

Descripción: El subsistema SimRT-VA debe modelar el sistema propulsor del vehículo aeroespacial generando el vector de empuje.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0060 Actuadores del vehículo

Descripción: El subsistema SimRT-VA debe modelar el sistema de actuadores del vehículo aeroespacial.

Criterio de Validación: Test/Demostración/Análisis/Inspección

01-SIM-0070 Notificación estado del vehículo

Descripción: El subsistema SimRT-VA debe notificar al subsistema OBC el estado instantáneo del vehículo.

Criterio de Validación: Test/Demostración/Análisis/Inspección

4.2.2 REQUISITOS NO FUNCIONALES

4.2.2.1 Requisitos de rendimiento

02-GEN-0001 Uso de CPU

Descripción: El uso medio de CPU no debe superar el 70 % del total en cada máquina de la plataforma.

Criterio de Validación: Test/Demostración/Análisis/Inspección

02-GEN-0010 Ancho de banda

Descripción: El ancho de banda de las tarjetas de comunicaciones serie debe ser superior a 300 Kbps.

Criterio de Validación: Test/Demostración/Análisis/Inspección

4.2.2.2 Requisitos de Interfaz

03-GEN-0001 Interfaz de usuario

Descripción: El sistema debe disponer de una interfaz gráfica de usuario.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-GEN-0010 Barra de título

Descripción: La interfaz gráfica de usuario debe disponer de una barra de título para identificar el sistema.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0020 Trayectoria visión cenital

Descripción: La interfaz gráfica de usuario debe mostrar la vista cenital de la trayectoria nominal y simulada.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0030 Mensajes de eventos de misión

Descripción: La interfaz gráfica de usuario debe mostrar la lista de eventos producidos durante la ejecución de una misión.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0040 Estado sistema

Descripción: La interfaz gráfica de usuario debe mostrar el estado de los diferentes subsistemas que forman el sistema.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0050 Estado comunicaciones

Descripción: La interfaz gráfica de usuario debe mostrar el estado de las comunicaciones entre los diferentes subsistemas.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0060 Cuenta atrás

Descripción: La interfaz gráfica de usuario debe mostrar la fecha y hora actual y el valor de la cuenta atrás una vez iniciada la misión.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0070 Misión

Descripción: La interfaz gráfica de usuario debe mostrar los parámetros que definen una misión.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0080 Comandos

Descripción: La interfaz gráfica de usuario debe disponer de los elementos gráficos necesarios para comandar el sistema.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-GEN-0090 Vector de estado

Descripción: La interfaz gráfica de usuario debe mostrar el vector de estado instantáneo.

Criterio de Validación: Test/**Demostración**/Análisis/Inspección

03-CMT-0001 Configuración CCMT

Descripción: El subsistema CCMT debe leer de un fichero de texto la

configuración de la aplicación.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-CMT-0010 Persistencia datos

Descripción: El subsistema CCMT debe integrar una base de datos para mantener la información gestionada por la aplicación.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-CMT-0020 Interfaz de comunicación serie CCMT-OBC

Descripción: El subsistema CCMT debe disponer de una interfaz de comunicaciones serie para comunicarse con el subsistema OBC.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-CMT-0030 Interfaz de comunicación digital CCMT-OBC

Descripción: El subsistema CCMT debe disponer de una interfaz de comunicaciones digital para comunicarse con el subsistema OBC.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-OBC-0001 Configuración OBC

Descripción: El subsistema OBC debe leer de un fichero de texto la configuración de la aplicación.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-OBC-0010 Interfaz de comunicación serie OBC-CCMT

Descripción: El subsistema OBC debe disponer de una interfaz de comunicaciones serie para comunicarse con el subsistema CCMT.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-OBC-0020 Interfaz de comunicación digital OBC-CCMT

Descripción: El subsistema OBC debe disponer de una interfaz de comunicaciones digital para comunicarse con el subsistema CCMT.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-OBC-0030 Interfaz de comunicación serie OBC-SimRT-VA

Descripción: El subsistema OBC debe disponer de una interfaz de comunicaciones serie para comunicarse con el subsistema SimRT-VA.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-SIM-0001 Configuración SimRT-VA

Descripción: El subsistema SimRT-VA debe leer de un fichero de texto la configuración de la aplicación.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-SIM-0010 Interfaz de comunicación serie SimRT-VA- OBC

Descripción: El subsistema SimRT-VA debe disponer de una interfaz de comunicaciones serie para comunicarse con el subsistema OBC.

Criterio de Validación: Test/Demostración/Análisis/Inspección

03-GEN-0100 Comunicaciones serie HDLC

Descripción: El enlace de comunicaciones serie definido en los requisitos 03-CMT-0020, 03-OBC-0010, 03-OBC-0030 y 03-SIM-0010 debe implementar el protocolo HDLC.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

4.2.2.3 Requisitos de recursos

05-CMT-0001 Plataforma CCMT

Descripción: La plataforma sobre la que debe instalarse el subsistema CCMT debe ser un PC Linux.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

05-OBC-0001 Plataforma OBC

Descripción: La plataforma sobre la que debe instalarse el subsistema OBC debe ser un PC con un sistema operativo tiempo real.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

05-SIM-0001 Plataforma SimRT-VA

Descripción: La plataforma sobre la que debe instalarse el subsistema SimRT-VA debe ser un PC con un sistema operativo tiempo real.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

4.2.2.4 Requisitos de documentación

07-GEN-0001 Idioma documentación del proyecto

Descripción: La documentación del proyecto debe redactarse en español.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

07-GEN-0010 Edición documentación del proyecto

Descripción: La documentación del proyecto debe redactarse con la herramienta Microsoft Word 2003.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

07-GEN-0020 Formato documentación del proyecto

Descripción: La documentación del proyecto debe presentarse en formato electrónico en PDF.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

4.2.2.5 Requisitos de diseño

15-GEN-0001 Diseño UML

Descripción: El sistema debe diseñarse utilizando el lenguaje de modelado unificado UML.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

15-OBC-0001 Diseño OBC

Descripción: El diseño del subsistema OBC debe permitir la integración de diferentes sistemas de navegación, guiado y control acordes al vehículo aeroespacial simulado.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

15-OBC-0001 Diseño SimRT-VA

Descripción: El diseño del subsistema SimRT-VA debe permitir la integración de diferentes modelos de vehículos aeroespaciales.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

4.2.2.6 Requisitos de desarrollo

16-CMT-0001 Desarrollo IHM

Descripción: El desarrollo de la interfaz gráfica de usuario del sistema deberá implementarse en el lenguaje de programación *Java*[™].

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

16-OBC-0001 Desarrollo OBC

Descripción: El desarrollo del subsistema OBC deberá realizarse en el lenguaje de programación *Java*[™] en su versión RT.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

16-SIM-0001 Desarrollo SimRT-VA

Descripción: El desarrollo del subsistema SimRT-VA deberá realizarse en el lenguaje de programación C/C++.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

16-GEN-0001 Desarrollo drivers de comunicación

Descripción: El desarrollo de los *drivers* de comunicación del sistema deberán realizarse en el lenguaje de programación C/C++.

Criterio de Validación: Test/Demostración/Análisis/**Inspección**

5 Diseño

Este capítulo presenta el diseño software de la plataforma de simulación de vuelos aeroespaciales.

El diseño se estructura entorno a los siguientes niveles de detalle:

- Arquitectura software.
- Diseño preliminar.
- Diseño detallado.

La descripción del diseño se realiza desde los puntos de vista siguientes:

- Estático.
- Funcional.
- Dinámico.

La visión estática de la arquitectura software describe que componentes software constituyen el sistema. El diseño preliminar desde un punto de vista estático detalla las clases ó módulos que implementan los componentes presentados en la arquitectura del software. Por último, el diseño detallado describe los atributos y métodos de las clases introducidas en el diseño preliminar.

Desde el punto de vista funcional detallaremos la función para la que ha sido concebido cada componente, clase, atributo y función.

La visión dinámica expone la interacción entre los diferentes elementos de cada nivel y la evolución en el tiempo.

5.1 ARQUITECTURA SOFTWARE

La arquitectura software del sistema se presenta en el diagrama de componentes de la Figura 24.

En el diagrama de componentes de la figura representa el sistema completo. En la figura podemos identificar los tres subsistemas que componen la plataforma de simulación, los componentes software que integra cada subsistema, las interfaces de los componentes y las dependencias entre componentes.

Además, en la figura se han identificado aquellos componentes específicos ó dependientes del vehículo aeroespacial simulado. Por ejemplo, los componentes identificados con las etiquetas NGC y Controlador del subsistema OBC deberán desarrollarse para cada vehículo ensayado. De igual forma, los componentes Aerodinámica, Dinámica, Actuadores, Transform y Propulsión del subsistema SimRT-VA vendrán determinados por el vehículo aeroespacial. El resto de componentes del sistema determinan la plataforma de ensayos.

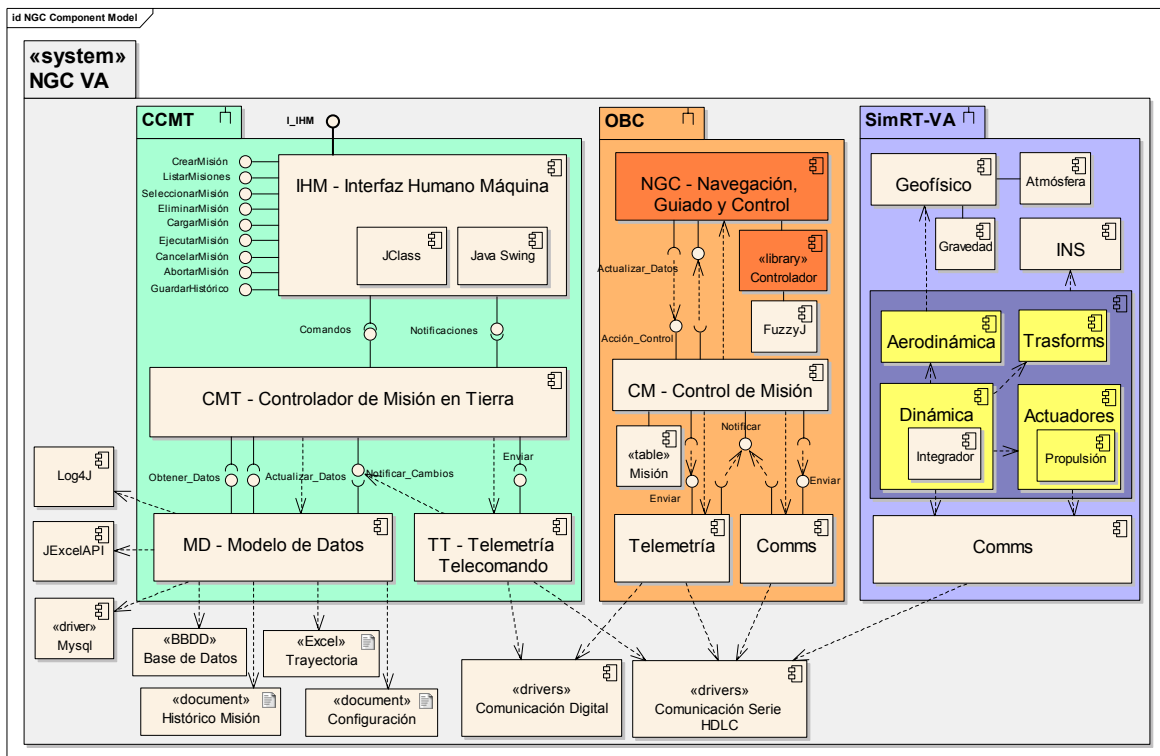


Figura 24: Diagrama de componentes

A continuación describiremos la función que desempeña cada componente en el sistema.

El subsistema CCMT está compuesto por los componentes software:

- **IHM: Interfaz Humano-Máquina.** Implementa la interfaz de usuario del sistema. La interfaz de usuario se implementará utilizando la librería *Swing* de *Java*TM e integrará los componentes gráficos de la librería *JClass*, desarrollados por la empresa *Quest Software*, para graficar las trayectorias.
- **CMT: Controlador de Misión en Tierra.** Implementa el controlador de la aplicación, según el modelo MVC. Es el núcleo del sistema.
- **MD: Modelo de Datos.** Implementa el modelo de datos de la aplicación. Es el componente encargado de mantener la persistencia de los datos, cargar la configuración del sistema, escribir el histórico de una misión, leer la trayectoria nominal y generar un *log*. El componente integrará los siguientes componentes comerciales: *Log4J*, *JExcelAPI* y *MySQL Connector*.
- **TT: Telemetría y Telecomando.** Implementa el sistema de comunicaciones serie y digital para intercambiar información con el subsistema OBC.

El subsistema OBC integra los componentes software siguientes:

- **NGC: Navegación, Guiado y Control.** Implementa el sistema de navegación, guiado y control del vehículo aeroespacial. Integra el controlador digital. La implementación del controlador basado en lógica difusa utilizará la librería *FuzzyJ*.

- **CM: Control de Misión.** Es el núcleo del subsistema OBC. Implementa la lógica de la aplicación.
- **Telemetría.** Implementa el sistema de comunicaciones serie y digital para intercambiar información con el subsistema CCMT.
- **Comms.** Implementa el sistema de comunicación serie para intercambiar información con el subsistema SimRT-VA.

El subsistema SimRT-VA esta compuesto por los componentes software:

- **Geofísico.** Implementa el modelo atmosférico y gravitacional. Calculará las condiciones meteorológicas, temperatura, presión, densidad, velocidad y dirección del aire, la velocidad del sonido, la fuerza de gravedad instantánea.
- **Aerodinámica.** Implementa el cálculo de los coeficientes aerodinámicos del vehículo, así como la fuerza y momento aerodinámicos instantáneos.
- **Dinámica.** Es el componente principal del subsistema. Implementa el reloj del simulador. Integra el algoritmo de integración de las ecuaciones de movimiento del vehículo. Calcula los vectores fuerza y momento a partir de la información proporcionada por los módulos de gravedad, propulsión y aerodinámica.
- **INS.** Simula el funcionamiento del navegador inercial y el GPS. La información de posición y los ángulos de orientación del vehículo los proporciona el modulo de dinámica.
- **Trasformaciones.** Componente especializado en el cálculo de transformación de coordenadas entre los diferentes sistemas de coordenadas definidos.
- **Actuadores.** Simula el funcionamiento de los actuadores del rehílo. Proporcionará el vector de empuje del motor en cada fase de vuelo.

Los componentes comunes utilizados por los tres subsistemas son:

- **Comunicación Serie HDLC.** *Driver* de comunicación serie HDLC. Integra la API proporcionada por el fabricante de las tarjetas de comunicación.
- **Comunicación Digital.** *Driver* de comunicación digital. Integra la API proporcionada por el fabricante de las tarjetas E/S digitales.

El sistema integra los siguientes componentes comerciales:

- **Log4J.** Librería *Open Source* desarrollada por *Apache Software Foundation* utilizada para generar el *log* del sistema.
- **JExcelAPI.** Librería *Open Source* utilizada para leer, escribir y modificar ficheros Excel.
- **MySQL Connector.** *Driver* de conexión con la base de datos *MySQL*.
- **JClass.** Librería de componentes gráficos desarrollada en *Java™* por *Quest Software*.
- **Swing.** Librería de componentes para el desarrollo de interfaces gráficas de usuario en *Java™* desarrollada por *Sun Microsystems*.
- **FuzzyJ.** Motor de inferencia de lógica difusa desarrollada en *Java™* por el *National Research Council of Canada's Institute for Information Technology*.

Por último, el diagrama de componentes de la Figura 24 muestra varios objetos necesarios para el funcionamiento del sistema, a saber:

- **BBDD.** Base de datos del sistema.
- **Excel.** Trayectoria nominal.
- **Ficheros:**
 - Histórico Misión.
 - Configuración.

La distribución física de componentes se presenta en el diagrama de despliegue de la Figura 25. El diagrama de despliegue modela la arquitectura del sistema en tiempo de ejecución, representa la configuración de los elementos hardware (nodos) y muestra la distribución de los componentes software en los nodos.

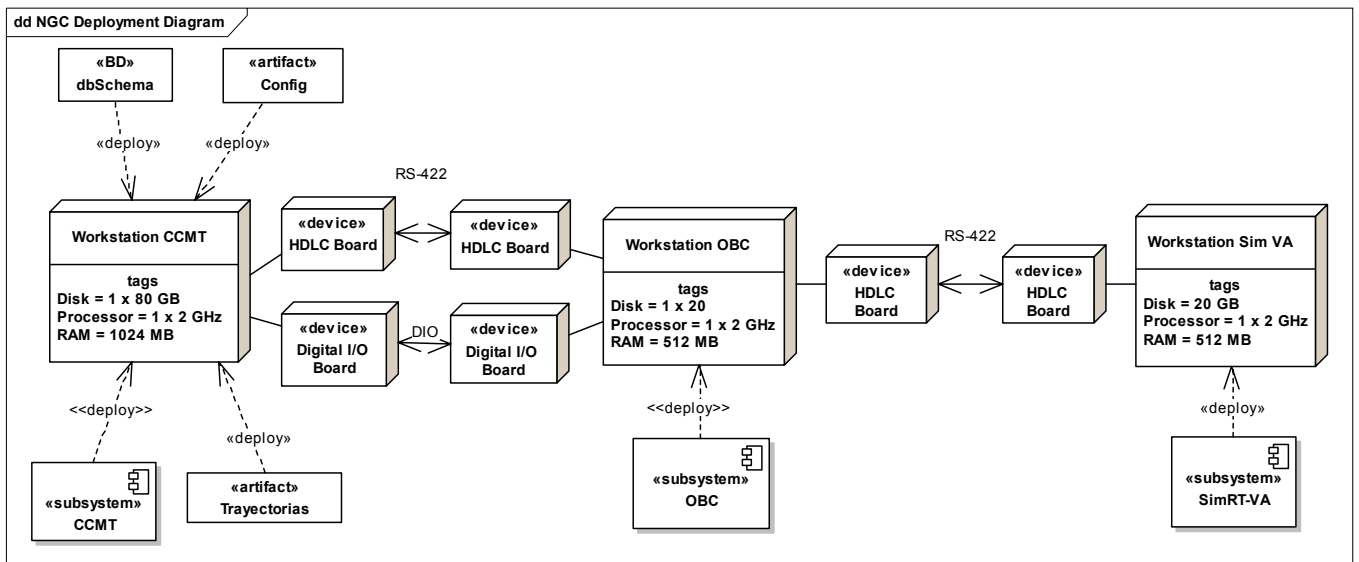


Figura 25: Diagrama de despliegue

5.2 DISEÑO PRELIMINAR

En este apartado presentamos el diseño preliminar de los componentes software del sistema. El objetivo del diseño preliminar es descomponer cada componente software en módulos ó clases.

Para cada componente software identificado en la arquitectura software, presentada en el capítulo 5.1, se describe el diagrama de clases y la función que desarrolla cada clase.

Durante la fase de diseño se intentará reaprovechar el mayor número de patrones de diseño.

El diseño del CCMT incluye una maqueta de la interfaz gráfica de usuario del sistema.

El diseño del simulador del vehículo aeroespacial se desarrollará con la ayuda de la herramienta *Simulink* integrada en *MATLAB*.

5.2.1 CENTRO DE CONTROL DE MISIÓN EN TIERRA

En este apartado presentamos el diseño preliminar de los componentes que forman el Centro de Control de Misión en Tierra.

Para el componente IHM se presentará una maqueta de la ventana principal de la aplicación.

5.2.1.1 Interfaz Humano-Máquina

La Figura 26 presenta la maqueta de la interfaz gráfica de usuario propuesta para el sistema.

En la maqueta se identifican todas las áreas visibles de la ventana principal de la aplicación CCMT.

Describiremos brevemente la función de los elementos gráficos dpresentados en la maqueta:

- **Barra de título.** Muestra el título de la aplicación: NGC – Centro de Control de Misión en Tierra.
- **Panel Trayectoria.** El panel titulado Trayectoria contiene dos vistas de la trayectoria del vehículo, la vista cenital y la frontal. En cada vista se superpondrán las trayectorias nominal y real. Además, se mostrarán los eventos producidos durante el transcurso de la misión.
- **Panel Eventos.** El panel titulado Eventos mostrará la lista de eventos generados. En la lista se mostrarán los eventos de la misión y todos los eventos de la aplicación. Se identificarán con un código de color los diferentes tipos de eventos. Por ejemplo, los errores se mostrarán en color rojo, etc.
- **Panel Estado.** El panel titulado Estado contiene la información relativa a:
 - Estado de las comunicaciones. Enlace CCMT – OBC y OBC – SimRT-VA.
 - Estado de ocupación de los *buffers* de recepción de datos.
 - Estado del navegador inercial.
 - Fase de vuelo del vehículo.
 - Motor activo.
 - Interetapa activa.
 - Estado de la misión.
- **Panel Cuenta Atrás:** El panel titulado Cuenta Atrás contiene la hora (UTC) del sistema, la fecha actual y la cuenta atrás restante para el lanzamiento del vehículo.
- **Panel Misión:** El panel titulado Misión contiene información relativa a la misión cargada en el OBC.
 - Identificar de misión.
 - Fecha y Hora de lanzamiento.

- **Panel Vector Estado.** El panel titulado Vector de Estado contiene la información relativa al vector de estado instantáneo del vehículo.
- **Panel Configuración.** El panel titulado configuración contiene dos botones:
 - MSN: Misión. Abre el diálogo para la visualización y edición de misiones.
 - BBDD: Base de datos. Abre el diálogo para editar los parámetros de conexión con la base de datos.
- **Panel Misión:** El panel titulado misión de la botonera de comandos contiene los botones siguientes:
 - INICIAR: Inicia la misión cargada.
 - PARAR: Detiene la cuenta atrás.
 - ABORTAR: Aborta una misión en curso.
 - CARGAR: Abre el diálogo que permite seleccionar la misión a cargar e introducir los parámetros iniciales de la misión.
- **Panel Modos.** El panel titulado Modos contiene los botones necesarios para comandar el navegador inercial.

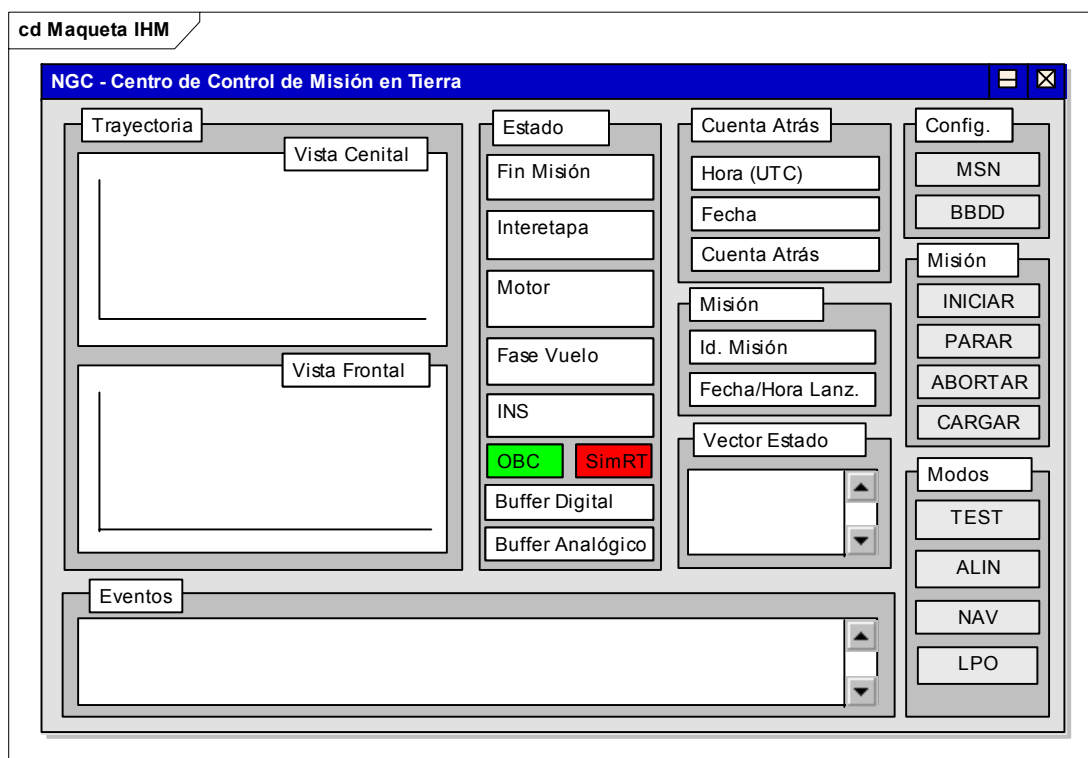


Figura 26: Maqueta IHM

La Figura 27 muestra el diagrama de clases de la interfaz gráfica de usuario y la conexión con el paquete *javax.swing* de Java™.

La Tabla 35 enumera y describe brevemente las clases presentadas en el diagrama de clases de la Figura 27.

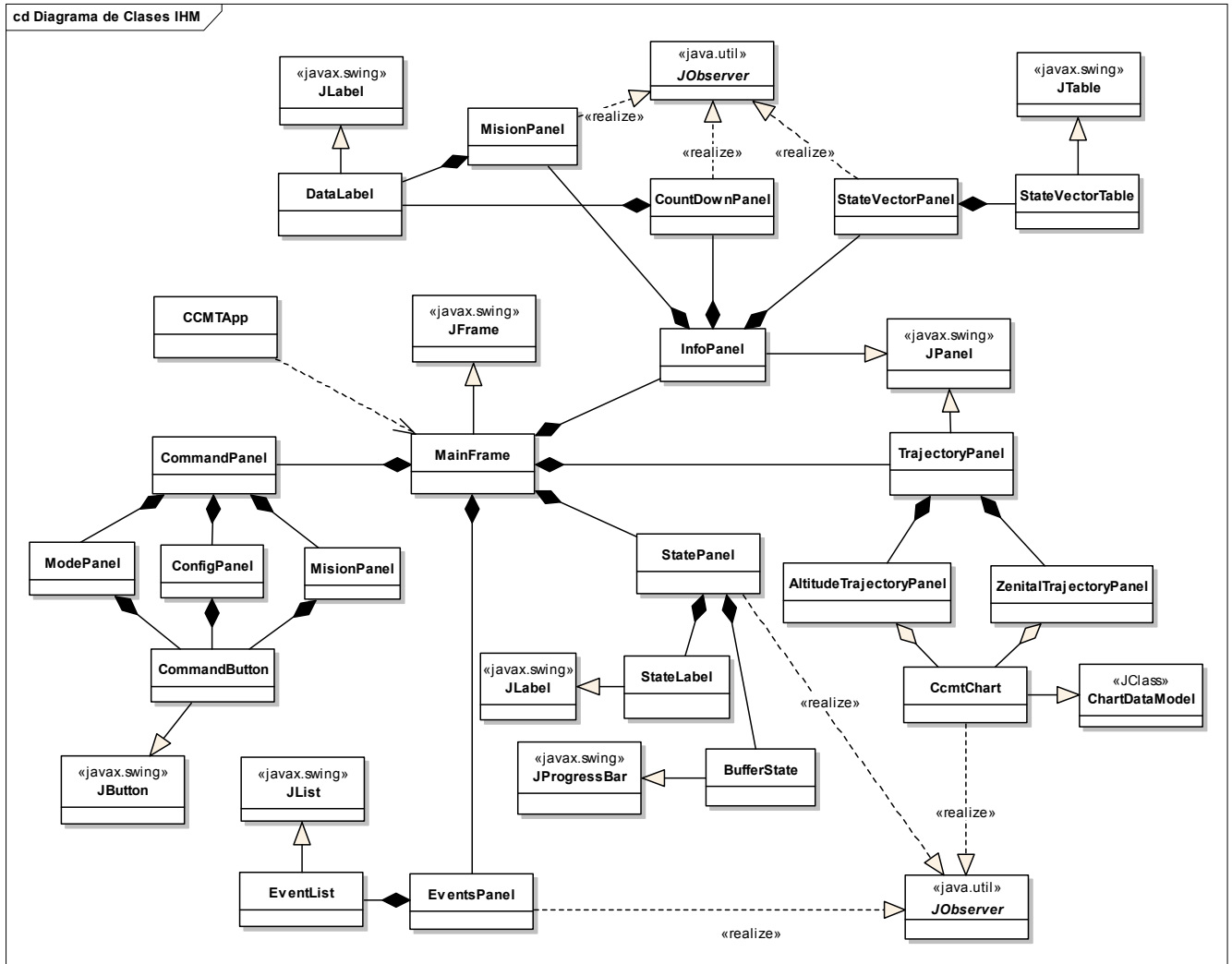


Figura 27: Diagrama de clases IHM

Clase	Descripción
CCMTApp	Contiene la función principal de la aplicación. Es la clase encargada de construir la interfaz gráfica y enlazarla con el controlador de la aplicación.
MainFrame	Marco principal de la aplicación.
TrajectoryPanel	Contiene los paneles con los componentes gráficos utilizados para graficar las trayectorias.
AltitudeTrajectoryPanel	Presenta la vista frontal de la trayectoria.
ZenitalTrajectoryPanel	Presenta la vista cenital de la trayectoria.
CcmtChart	Implementa la lógica para presentar las trayectorias. Hereda de la clase <i>ChartDataModel</i> .
StatePanel	Contiene los elementos gráficos necesarios para mostrar el estado de los dispositivos instalados en el vehículo. Este panel será dinámico y vendrá de-

	terminado por el vehículo simulado. La configuración del panel se incluirá en el fichero de configuración.
StateLabel	Muestra el estado de un dispositivo. La clase modificará el color de fondo para visualizar los diferentes estados.
BufferState	Barra de progreso utilizada para indicar el estado de ocupación de los <i>buffers</i> de recepción de datos.
EventsPanel	Contiene los elementos gráficos necesarios para mostrar la lista de eventos de la aplicación.
EventList	Presenta la lista de eventos del sistema.
InfoPanel	Contiene los elementos gráficos necesarios para mostrar la información de la misión en curso, cuenta atrás y vector de estado instantáneo.
MisionPanel	Muestra la información de la misión en curso.
CountDownPanel	Muestra la cuenta atrás.
StateVectorPanel	Contiene los elementos gráficos necesarios para mostrar el vector de estado instantáneo.
StateVectorTable	Muestra el vector de estado instantáneo en forma tabular.
DataLabel	Componente gráfico genérico utilizado para presentar la información de los datos.

Tabla 35: Descripción clases IHM

5.2.1.2 Controlador de Misión en Tierra

El diagrama de clases del componente CMT se presenta en la Figura 28. La descripción de las clases se detalla en la Tabla 36.

Clase	Descripción
CmtManager	Clase principal del componente. Es la clase encargada de construir el resto de clases del componente.
CmdMsn	Clase genérica de los comandos relacionados con el control de la misión.
CmdMsnStop	Comando utilizado para detener la cuenta atrás.
CmdMsnInit	Comando utilizado para iniciar una misión.
CmdMsnCancel	Comando utilizado para cancelar una misión cargada en el OBC.
CmdMsnAbort	Comando utilizado para abortar una misión.
CmdMode	Clase genérica de los comandos relacionados con la gestión del navegador inercial.
CmdNav	Comando utilizado para ordenar el modo NAV de la INS.
CmdAlin	Comando utilizado para ordenar el modo ALIN de la INS.
CmdTest	Comando utilizado para ordenar el modo TEST de la INS.
Cmdlpo	Comando utilizado para ordenar el modo LPO de la INS.
CmdConfig	Clase genérica de los comandos relacionados con la

	configuración del sistema.
CmdConfigBD	Comando utilizado para abrir el diálogo de configuración de los parámetros de la base de datos del sistema.
CmdConfigMsn	Comando utilizado para abrir el diálogo de configuración de una misión.
CmdCloseApp	Comando ejecutado al ordenar el cierre de la aplicación.
AnalogDataReader	Clase encargada de leer periódicamente el <i>buffer</i> de datos analógicos.
BinaryDataReader	Clase encargada de leer periódicamente el <i>buffer</i> de datos binarios.
ReaderTask	Hilo de ejecución utilizado para la lectura de <i>buffers</i> .
DbWriter	Clase encargada de gestionar la escritura de datos en la base de datos.
WriterTask	Hilo de ejecución encargado de enviar un paquete de datos a la base de datos.

Tabla 36: Descripción clases CMT

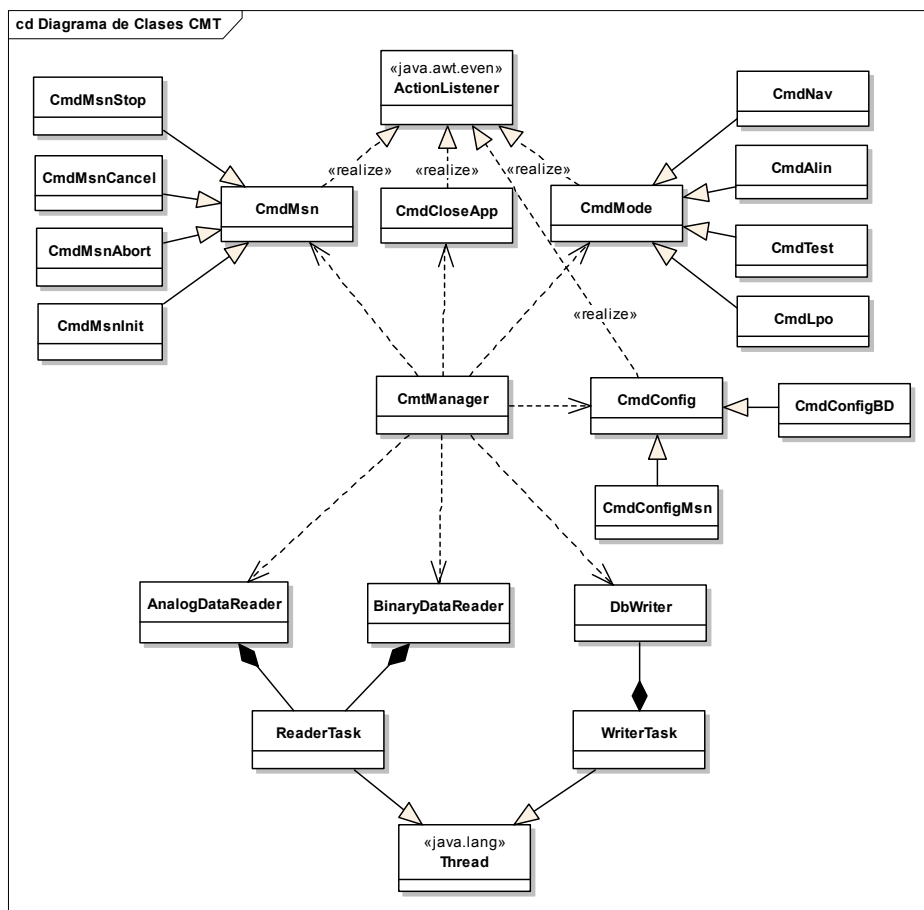


Figura 28: Diagrama de clases CMT

5.2.1.3 Modelo de Datos

El diagrama de clases del componente identificado con el nombre Modelo de Datos se presenta en la Figura 29.

La Tabla 37 detalla las clases que componen el modelo de datos y describe su funcionalidad.

Clase	Descripción
DBInterface	Intefaz del componente.
DBImpl	Implementación de la interfaz.
DBException	Excepción generada por el componente para notificar un error.
ConcurrentInsertion	Clase encargada de enviar un paquete de datos a la base de datos.
ExcelManager	Clase encargada de gestionar la lectura de datos desde un fichero <i>Excel</i> .
PropertiesManager	Clase encargada de la gestión de los parámetros de configuración de la aplicación.
Logger	Log del sistema.
DBManager	Clase encargada de la gestión con la base de datos.
ModuleState	Clase encargada de mantener el estado de los diferentes equipos del vehículo.
Module	Mantiene el estado e información de un equipo concreto del vehículo.
BinaryHistoricDataQueue	Lista de datos binarios.
BinaryData	Dato binario.
AnalogHistoricDataQueue	Lista de datos analógicos.
AnalogData	Dato analógico.
CommState	Mantiene el estado de los diferentes enlaces de comunicación.
Phase	Mantiene la fase de vuelo.
EventQueue	Lista de eventos del vehículo.
Event	Evento.
Misión	Mantiene la información relativa a una misión.
LaunchPoint	Mantiene la información relativa al punto de lanzamiento.
SystemErrorTable	Lista de errores.
SystemError	Error.
ByteList	Lista de <i>bytes</i> . Es la estructura utilizada por el componente de telemetría para componer una trama de datos.
WayPoint	Punto de trayectoria.

Tabla 37: Descripción clases MD

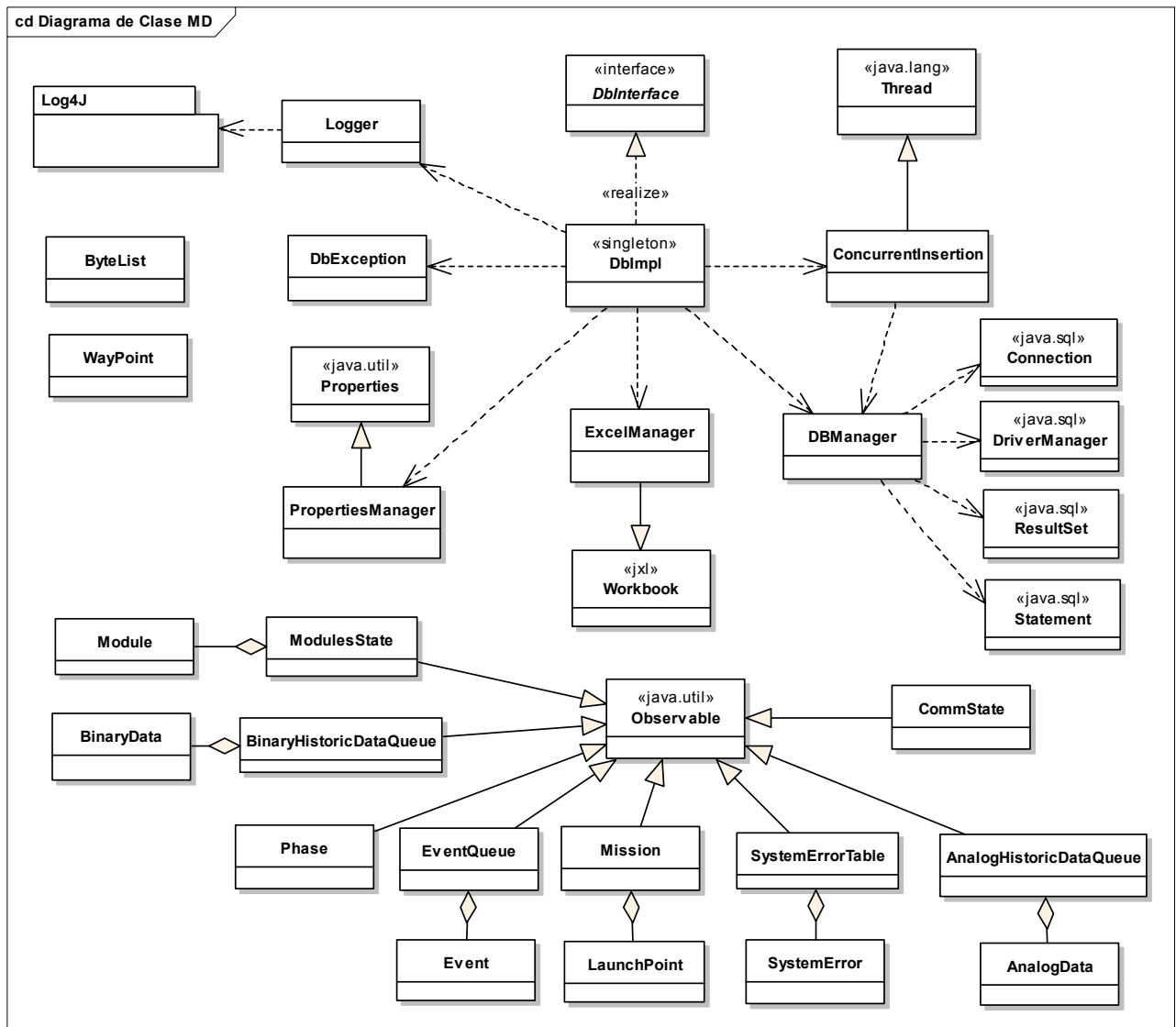


Figura 29: Diagrama de clases MD

La Figura 30 muestra el modelo de la base de datos del sistema. En el modelo se representan las tablas que componen la base de datos y la relación entre ellas. En el diagrama se han identificado únicamente las columnas relativas a las claves primarias y foráneas, el resto de información se añadirá durante el desarrollo del diseño preliminar.

Clase	Descripción
T_MISSIONS	Mantiene la lista de misiones.
T_MISSION_EVENTS_DATA	Mantiene la lista de eventos de las misiones ejecutadas.
T_MISSION_STATE_VECTOR_DATA	Mantiene la lista de puntos (Vector de estado) de las misiones ejecutadas.

Tabla 38: Descripción modelo BD

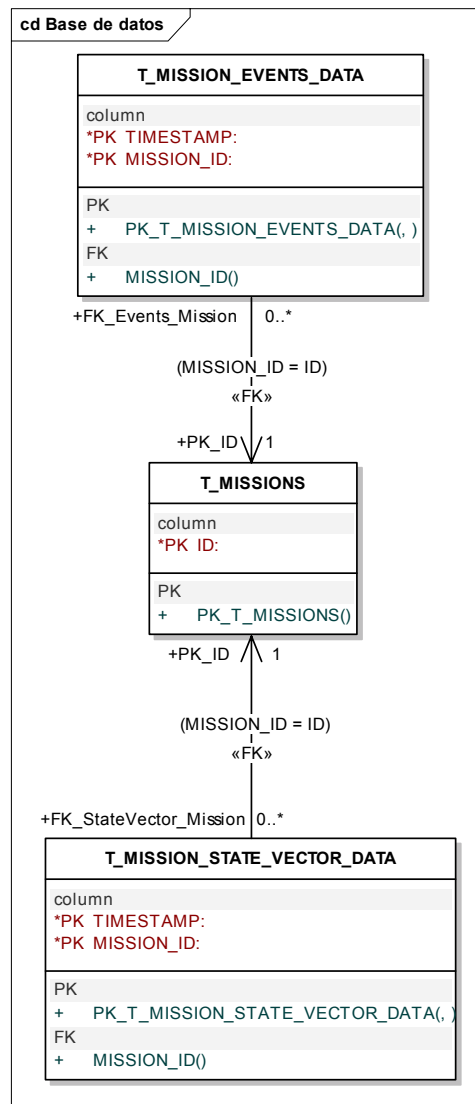


Figura 30: Modelo BD

5.2.1.4 Telemetría y Telecomando

El diagrama de clases del componente encargado de las comunicaciones entre los subsistemas CCMT y OBC se presenta en la Figura 31.

La Tabla 39 detalla las clases del diagrama presentado en la Figura 31 y describe brevemente la funcionalidad que desarrollan.

Clase	Descripción
CommInterface	Interfaz pública del componente.
CommManager	Implementa la interfaz del componente.
JNI_DigitalDriver	Interfaz JNI utilizada para enlazar el driver de comunicación digital desarrollado en C/C++.
DigitalDriver	Implementación del driver de comunicaciones digitales.

Pci7248_Board_Driver	Implementación del driver de la tarjeta de E/S digitales. Proporcionado por el fabricante de la tarjeta.
JNI_SerialDriver	Interfaz JNI utilizada para enlazar el driver de comunicación serie HDLC desarrollado en C/C++.
SerialDriver	Implementación del driver de comunicación serie.
Synclink_Board_Driver	Implementación del driver de la tarjeta de comunicaciones serie HDLC. Proporcionado por el fabricante de la tarjeta.
Queue	Clase abstracta utilizada para la gestión de los <i>buffers</i> de datos.
BinaryBuffer	<i>Buffer</i> de datos binarios.
AnalogBuffer	<i>Buffer</i> de datos analógicos.
Frame	Clase utilizada para la codificación / decodificación de las tramas del protocolo de comunicación.
ByteList	Trama de datos a enviar.

Tabla 39: Descripción clases T&T

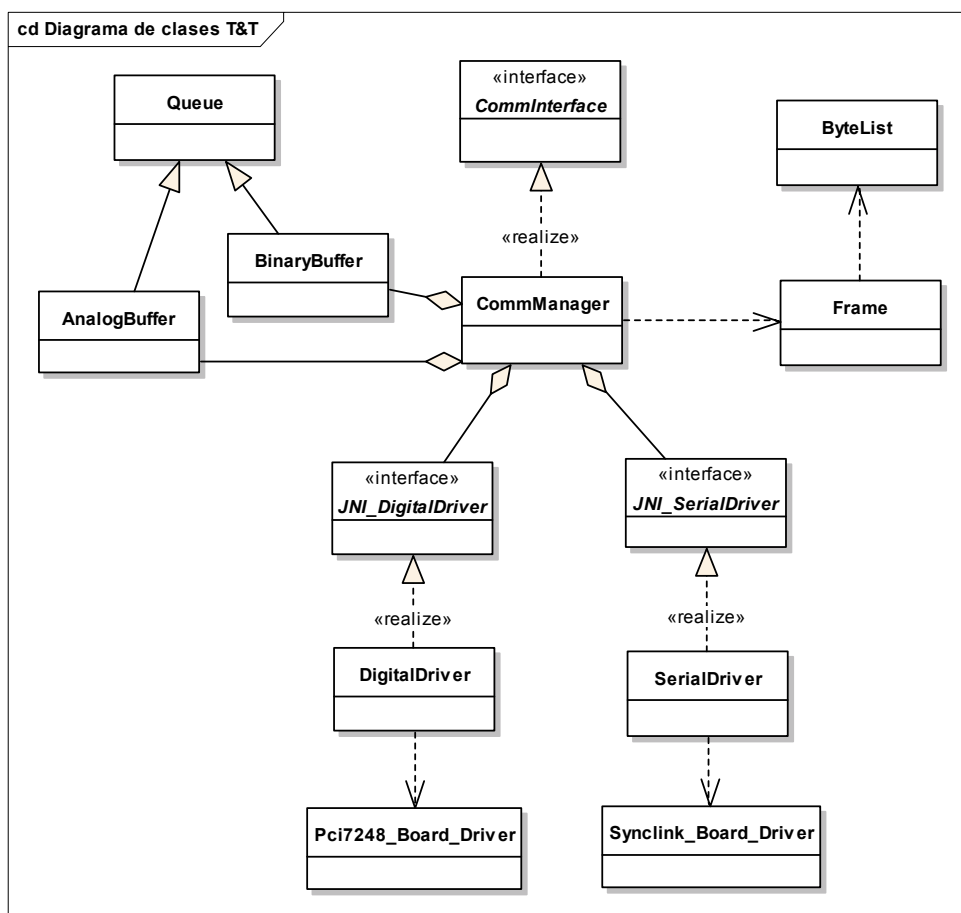


Figura 31: Diagrama de clases T&T

5.2.2 ONBOARD COMPUTER

En este apartado presentamos el diseño preliminar de los componentes que forman el OnBoard Computer.

5.2.2.1 Control de Misión

El diagrama de clases del componente identificado con nombre Control de Misión se presenta en la Figura 32. La Tabla 40 detalla las clases del diagrama y describe brevemente su funcionalidad.

Clase	Descripción
OBCApp	Clase principal de la aplicación.
OnboardSystem	Núcleo del componente. Coordina el resto de componentes del sistema y mantiene la información relativa a la misión en ejecución.
CommInterface	Interfaz del componente encargado de las comunicaciones.
NGC_Interface	Interfaz del componente Navegación, Guiado y Control.
Misión	Mantiene la información de la misión en curso.
State	Mantiene el estado de la misión.
LaunchPosition	Mantiene la posición de lanzamiento.
NominalTrajectory	Mantiene la trayectoria nominal de la misión.
WayPoint	Punto de la trayectoria.

Tabla 40: Descripción clases CM

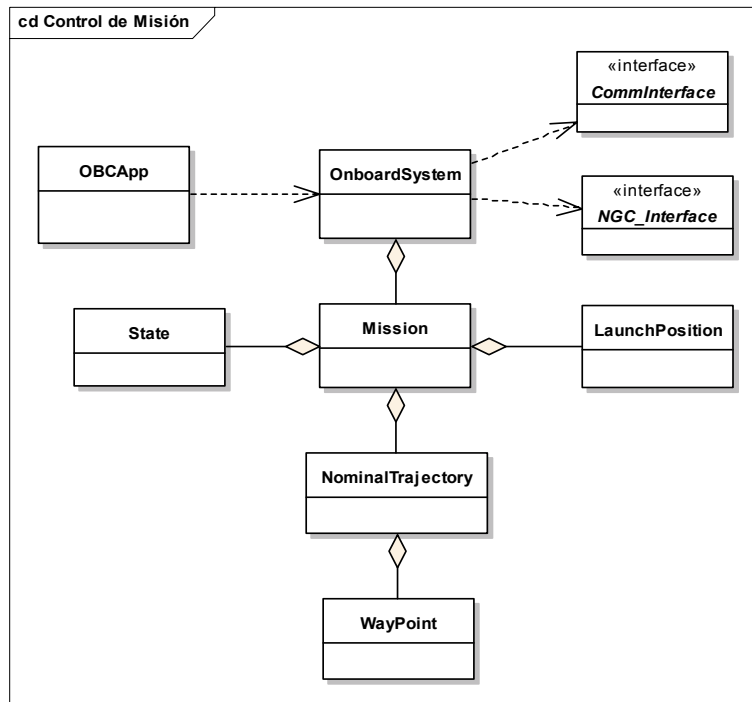


Figura 32: Diagrama de clase CM

5.2.2.2 Navegación, Guiado y Control

El diagrama de clases del componente Navegación, Guiado y Control se presenta en la Figura 33. La Tabla 41 detalla las clases del diagrama y describe brevemente su funcionalidad.

Clase	Descripción
NGC_Interface	Interfaz del componente.
NGC_Impl	Implementación de la interfaz.
DigitalControllerInterface	Interfaz del controlador digital.
DigitalControllerImpl	Implementación de la interfaz del controlador digital.
FuzzyLogicController	Controlador digital específico del vehículo aeroespacial basado en lógica difusa.
FLC_1st_Phase	Implementación del controlador de la primera fase de vuelo.
FLC_2nd_Phase	Implementación del controlador de la segunda fase de vuelo.
FLC_3rd_Phase	Implementación del controlador de la tercera fase de vuelo.
ControlSignal	Señal de control a enviar al vehículo aeroespacial.
Action	Acciones a enviar al vehículo aeroespacial. P.j. ignición motor, activación interetapa, etc.

Tabla 41: Descripción clases NGC

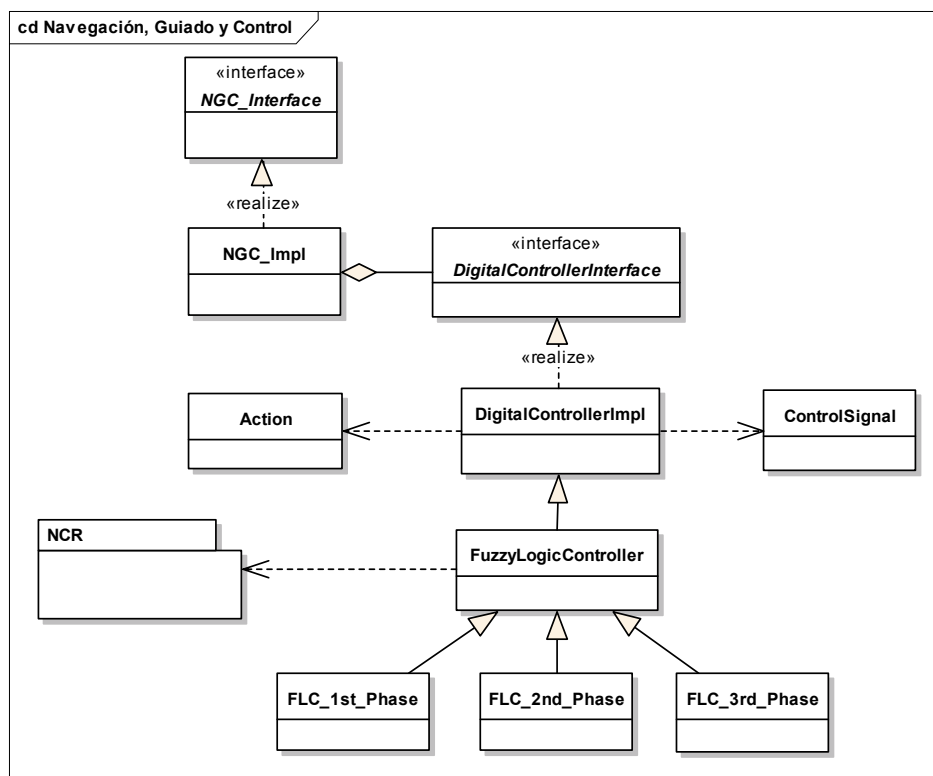


Figura 33: Diagrama de clase NGC

5.2.2.3 Telemetría y Comunicaciones

El diagrama de clases de los componentes de telemetría y comunicaciones es idéntico al presentado en el apartado 5.2.1.4 Telemetría y Telecomando.

5.2.3 SIMULADOR VEHICULO AEROESPACIAL

El diseño preliminar del Simulador del Vehículo Aeroespacial SimRT-VA se ha desarrollado con la ayuda del entorno de simulación *Simulink* integrado en la herramienta *MATLAB*.

El objetivo para la fase de codificación es generar el código del simulador del vehículo a partir del modelo desarrollado en *Simulink*.

El diagrama de módulos del vehículo se presenta en la Figura 34.

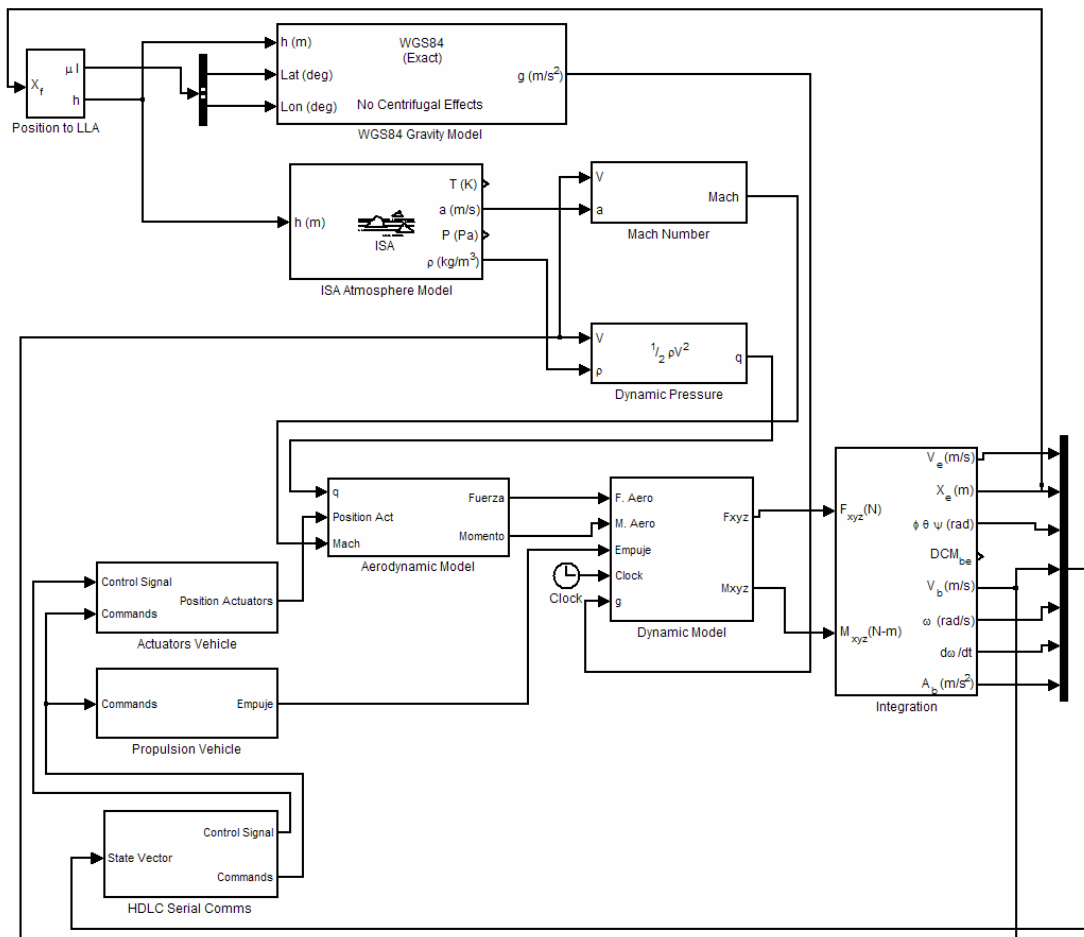


Figura 34: Diagrama de módulos SimRT-VA

5.3 DISEÑO DETALLADO

El diseño detallado no se describirá debido a la extensión del proyecto.

6 Conclusiones

La experiencia adquirida durante años de trabajo en el entorno empresarial, centrado en el desarrollo de sistemas críticos para el departamento de defensa español, motivó la idea de presentar un sistema de navegación, guiado y control de vehículos aeroespaciales como proyecto final de carrera.

El alcance del proyecto final de carrera, dado el volumen de trabajo, no ha permitido implementar la propuesta presentada en este documento. No obstante, la memoria describe todas las fases de un proyecto de ingeniería del software, desde la planificación del proyecto hasta la etapa de codificación del producto.

El objetivo de este capítulo es analizar el trabajo realizado durante el desarrollo del proyecto, discutir aquellos aspectos que desde el punto de vista del autor son mejorables y comentar las posibles líneas de trabajo abiertas para culminar el trabajo iniciado en este proyecto final de carrera.

Para presentar los puntos citados anteriormente se seguirá la misma estructura adoptada en el documento a fin de analizar todas las fases del proyecto.

6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO

6.1.1 PLAN DE PROYECTO

La planificación del proyecto presentada en el plan de proyecto podría desviarse si el equipo de proyecto no acumula la experiencia requerida.

La organización general del equipo es una estructura típica de un proyecto de ingeniería. Probablemente, en un proyecto de I+D+I deberíamos motivar la participación de ingenieros *juniors*.

En cuanto al ciclo de vida seleccionado, considero que es un modelo idóneo para proyectos de tamaño medio. En proyectos de mayor envergadura, un ciclo de vida iterativo siguiendo el modelo del ciclo de vida en V para cada iteración podría ser interesante.

6.1.2 ANÁLISIS

La estructura del capítulo, presentando en una primera instancia el análisis científico a fin de introducir el contexto de trabajo, continuando con el análisis del sistema informático que propone una solución técnica desde el punto de vista de la ingeniería del software y finalizando con el análisis económico que evalúa el coste de implementar la solución propuesta, sigue la línea lógica de eventos en la realización del proyecto.

El análisis científico no aporta la información necesaria para implementar el simulador de un vehículo aeroespacial. El trabajo restante en este campo es más pro-

pio de estudiantes de ingeniería aeronáutica ó física. No obstante, se puede desarrollar un modelo simple para validar la plataforma de simulación.

En cuanto al análisis del sistema informático, considero que profundizar en el diseño de los protocolos de comunicación es excesivo nivel de detalle para la etapa de análisis. Este desarrollo es más propio de la etapa de diseño. A pesar de esto, ha sido la base ideal para estimar el ancho de banda requerido por los enlaces de comunicación.

Desde el punto de vista del autor es preferible anticiparse en lo posible y analizar en profundidad aquellos aspectos críticos del proyecto para no poner en riesgo la viabilidad del mismo.

El análisis económico recoge los tres aspectos a tener en cuenta a la hora de evaluar el coste del proyecto, es decir, recursos humanos implicados, licencias de software requeridas y material hardware.

Respecto a las licencias del software a adquirir para el desarrollo del proyecto, la compra de licencias flotantes puede suponer una ventaja para el equipo de proyecto. Éste sería un punto a evaluar a la hora de adquirir las licencias.

6.1.3 ESPECIFICACIÓN

La especificación técnica presentada en el capítulo 4 sólo incluye requisitos de alto nivel (HLR). La especificación de requisitos de bajo nivel (LLR) no se ha incluido.

En un sistema compuesto por diferentes subsistemas es interesante definir requisitos de bajo nivel derivados de los requisitos de alto nivel.

La especificación de requisitos de bajo nivel delimitará el diseño de los subsistemas y facilitará la etapa de validación.

La mantenibilidad de los requisitos redactados directamente en documentos resulta compleja cuando el número de requisitos aumenta. El uso de herramientas para este propósito es recomendable. Algunas herramientas recomendables podrían ser: *RequisitePro* ó *Doors*.

Un punto importante a incluir en futuros trabajos derivados del proyecto presentado en esta memoria sería incluir la trazabilidad entre requisitos y componentes software y entre requisitos y procedimientos de test.

Las matrices de trazabilidad es un trabajo tedioso y difícil de mantener. Una herramienta ideal para realizar este trabajo es *Reqtify*.

En cuanto a la elección de los lenguajes de programación y COTS seleccionados para el desarrollo del proyecto, opino que el lenguaje de programación *Java*[™] propuesto para el desarrollo del CCMT es acertada. *Java*[™] es un lenguaje mucho más productivo que otros lenguajes de programación como por ejemplo *C/C++* ó *Delphi*.

La implementación del subsistema OBC utilizando la versión tiempo real de *Java*[™] nos permitirá testear las capacidades de una máquina virtual RT. Una vez más, la productividad del lenguaje es un punto a tener en cuenta, además, el entorno de desarrollo *JamaivaVM* permite compilar el *bytecode* para generar código en el lenguaje de programación C.

La limitación de *JamaivaVM* radica en el conjunto de librerías de la API de *Java*[™] disponibles. Esta limitación no representará un problema para el desarrollo del OBC ya que el software a desarrollar y los COTS seleccionados para la implemen-

tación del sistema de control heredarán de librerías disponibles en la API de *JavaRT* proporcionada con *JamaivaVM*.

Respecto al desarrollo de la interfaz gráfica de usuario basada en el *Swing* de *Java™* es una opción, otra opción a tener en cuenta sería utilizar el *Workbench* de *Eclipse*. Posiblemente, esta segunda propuesta sea más interesante ya que ofrece un mejor rendimiento.

6.1.4 DISEÑO

El capítulo 5, dedicado al diseño del sistema, presenta la arquitectura software y el diseño preliminar del sistema. El diseño detallado no se ha realizado debido al volumen de trabajo que representa.

El capítulo dedicado a la arquitectura software presenta el diagrama de componentes y el diagrama de despliegue del sistema. Dos aspectos a incluir en este apartado, desde el punto de vista del autor, serían:

- La definición de las interfaces de los componentes. La definición de interfaces debe detallar los métodos y eventos de cada componente.
- La estructura de carpetas del repositorio.

Respecto al diseño preliminar, se han desarrollado los diagramas de clases de los componentes. La inclusión de los diagramas de secuencia principales permitirá despejar las dudas sobre las relaciones entre clases.

Existen numerosas herramientas para el diseño de software, p.j. *Rational Rose*, *Enterprise Architect*, *Rhapsody*, etc. El diseño presentado en este documento ha sido realizado con la ayuda de la herramienta *Enterprise Architect*. *Enterprise Architect* no es la herramienta recomendada para el desarrollo de sistemas críticos. Para este tipo de sistemas es más común utilizar *Rhapsody*.

Las herramientas citadas anteriormente permiten generar la documentación de diseño automáticamente. Una recomendación, antes de continuar con el diseño, sería preparar los *templates* necesarios para generar la documentación del diseño preliminar automáticamente. No tiene sentido generar manualmente un documento con la información detallada de cada clase.

6.2 CONCLUSIÓN

El desarrollo de una plataforma de simulación de estas características debe permitir profundizar en el diseño de sistemas de control digital avanzados.

Las posibilidades que se vislumbran con el lanzamiento de Galileo son múltiples y la necesidad de sistemas de control automático para todo tipo de vehículos es un área a tener en cuenta en un futuro cercano, por ejemplo, vuelos no tripulados (UAV), control de las fases críticas de vuelo, navegación automática de embarcaciones, etc.

El trabajo realizado a lo largo del proyecto me ha permitido adquirir conocimientos relacionados con el sector aeroespacial y la ingeniería de control, así como ampliar los conocimientos adquiridos durante el transcurso de los estudios de ingeniería informática.

En cuanto a la investigación realizada en relación al uso de la versión tiempo real de lenguaje de programación *Java™* y de la especificación RTJS podemos decir

que la máquina virtual de *JavaRT* mejora notablemente las capacidades respecto de la versión estándar.

Así pues, podemos decir que el estudio de la versión *JavaRT* me ha permitido profundizar en el conocimiento sobre un lenguaje de programación del que se espera, en un futuro, pueda competir con los lenguajes utilizados tradicionalmente en el desarrollo de sistemas críticos.

La propuesta realizada para desarrollar el sistema de navegación, guiado y control basado en lógica difusa me ha permitido ampliar los conocimientos adquiridos en la asignatura de inteligencia artificial relacionados con el tema.

La redacción de los capítulos dedicados a la especificación técnica de requisito y diseño del sistema me han aportado la experiencia necesaria para comprender la importancia del uso de herramientas de gestión de requisitos y de diseño de software.

Por último, comentar que el estudio económico me ha generado la necesidad de contactar con diferentes empresas, por lo general extranjeras, para solicitar presupuesto sobre sus productos.

Finalmente, podemos concluir que el desarrollo del proyecto presentado en este documento demuestra la viabilidad del mismo y establece las bases para implementar la plataforma de simulación de vuelos para vehículos aeroespaciales.

6.3 LÍNEAS ABIERTAS

La implementación de una plataforma de simulación de estas características puede proporcionar un marco de trabajo útil para los siguientes propósitos:

- Plataforma de test de sistemas de control para plantas modeladas con cualquier herramienta de prototipado.
- Implementación de sistemas de control para plantas difíciles de modelar (HWIL).
- Plataforma de prácticas para los estudiantes de teoría de control.

Bibliografía

- Elices, T. (1991). *Introducción a la dinámica espacial*. Publicación INTA.
- Driankov, D., Hellendoorn, H. & Reinfrank, M. (1993). *An Introduction to Fuzzy Control*. (2nd Ed.) Alemania. Springer.
- Fortescue, P., Stark, J. & Swinerd, G. (2003). *Spacecraft Systems Engineering*. (3rd Ed.). Chichester. Wiley.
- Noton, M. (1998). *Spacecraft Navigation and Guidance*. London. Springer.
- Ogata, K. (1995). *Discrete-time control systems*. (2nd Ed.). Englewood Cliffs. Prentice-Hall.

A handwritten signature in dark ink, appearing to read 'Alberto Pérez Marcos', is written over a faint, light-colored watermark of the same name. The signature is fluid and cursive, with a long horizontal stroke extending to the left.

Alberto Pérez Marcos

Este documento presenta el *Proyecto Final de Carrera* titulado **Navegación, guiado y control de vehículos aeroespaciales**.

El objetivo del proyecto es diseñar una plataforma de ensayos para la simulación de vuelos de vehículos aeroespaciales. La plataforma permitirá diseñar y evaluar los algoritmos de navegación, guiado y control de los vehículos aeroespaciales modelados en la plataforma de simulación, focalizando el trabajo de los ingenieros en el modelado de vehículos y en el desarrollo de sistemas de control digital.

La memoria recoge las fases de un proyecto de ingeniería del software, describiendo el plan de proyecto, el análisis del sistema, la especificación de requisitos y el diseño del mismo.

Aquest document presenta el *Projecte Fi de Carrera* titulat **Navegació, guiat i control de vehicles aeroespacials**.

L'objectiu del projecte és dissenyar una plataforma d'assajos per a la simulació de vols de vehicles aeroespacials. La plataforma permetrà dissenyar i avaluar els algorismes de navegació, guiat i control dels vehicles aeroespacials modelats a la plataforma de simulació, focalitzant el treball dels enginyers en el modelat de vehicles i en el desenvolupament de sistemes de control digital.

La memòria recull les fases d'un projecte d'enginyeria del software, descrivint el pla de projecte, l'anàlisi del sistema, l'especificació de requisits i el disseny del mateix.

This document shows the *Final Degree Project* which is called **Navigation, guidance and control of spacecrafts**.

The target of the project is to design a trial platform to simulate spacecraft's flights. The platform will permit design and evaluate the navigation, guidance and control algorithms of every modelled spacecraft in the simulation platform, focusing the engineers' work on modelling spacecrafts and developing digital control systems.

The memory includes the phases of a software engineering project, describing the project plan, the system analysis, the requirement specification and the software design.