



# Diseño de software espacial

Semestre: 2r semestre 2025/2026

## Índice

<b>Contexto.....</b>	2
<b>Ejercicio 1.....</b>	3
<b>Ejercicio 2 .....</b>	5
<b>Ejercicio 3 .....</b>	7

## Contexto

Actualmente trabajáis para la Agencia Espacial Europea (ESA) en el departamento de desarrollo de software espacial. Por lo general sueles hacer mantenimiento de programas informáticos para misiones no tripuladas y pequeñas actualizaciones de estos. Como ya sabes, la NASA ha publicado que para 2040 esperan colonizar Marte y estas declaraciones han generado mucho revuelo en la ESA. Debido a la presión que hay actualmente en la agencia, parece que va a venir una temporada de mucha carga de trabajo. Tu responsable te ha enviado varios correos urgentes para la creación de nuevos sistemas de software. Tu misión como responsable del departamento es diseñar estos nuevos sistemas.

Consejos:

- Crea software sencillo (si te piden coches, no hagas aviones). Muchos desarrolladores pensando un poco en la escalabilidad o en la evolución del software crean sistemas mucho más complejos de lo necesario. Se ha demostrado que no es una buena práctica. Lo mejor que puedes hacer es crear el software con las especificaciones que te piden y eso sí, utilizando las mejores prácticas posibles (principios SOLID) e intentar que sea lo más escalable posible.
- Piensa como desarrollador. A la vez que diseñas piensa como se llevaría a código. Mucha gente piensa que diseñar software es como diseñar bases de datos. ¡NO! Es un error muy común. En las bases de datos no hay interfaces, no hay clases abstractas, no hay polimorfismo, etc....
- Piensa en los casos de uso. Hazte las siguientes preguntas: ¿Cómo haría un usuario para dar de alta un libro?, ¿Cómo haría una biblioteca para dar de alta un autor?, ¿Cómo haría la agencia espacial para ejecutar el método anular\_mision()?. Imagínate la UI (interfaz de usuario), imagínate los campos que tiene que llenar e imagínate los botones que tiene. Esto siempre ayuda a entender como debes diseñarlo.

Por último, dejo unas frases que siempre se deberían de tener en cuenta:

"Una línea de código se escribe una vez, pero se lee 10 veces"

"En el mundo real, el coste de leer y entender código es 10 veces mayor que el coste de escribirlo. Si tu diseño es 'inteligente' pero difícil de seguir, estás creando una deuda técnica que alguien (probablemente tú mismo en seis meses) tendrá que pagar con horas de sueño"

## Ejercicio 1

### **Sistema Integrado de Gestión de Activos Interplanetarios (SIGAI)**

*De: Dirección General de la Agencia Espacial Europea (ESA)*

*Para: Departamento de Ingeniería de Software*

*Asunto: Especificaciones funcionales para la nueva flota de exploración*

*Estimados ingenieros,*

*La expansión de nuestras fronteras hacia Marte y las lunas de Júpiter requiere de un sistema de control unificado. No podemos permitirnos gestionar cada activo de forma aislada; necesitamos un software robusto que modele nuestra flota actual y futura bajo los siguientes lineamientos operativos:*

#### **1. Definición de Activos de Vuelo**

*Nuestra flota se basa en el concepto de Vehículo Espacial. Independientemente de su propósito, todo vehículo en nuestra base de datos debe estar identificado por su nombre y su autonomía operativa. Sin embargo, en la práctica, solo operamos con dos tipos de configuraciones reales:*

- *Sondas Exploradoras: Diseñadas para el espacio profundo. Deben incluir obligatoriamente su frecuencia de transmisión para el enlace de datos. Su protocolo operativo principal es la ejecución de escaneos de largo alcance.*
- *Naves Tripuladas: Diseñadas para el transporte de personal. Deben registrar la capacidad de tripulantes a bordo. Su protocolo operativo principal es la gestión del soporte vital.*

*Es fundamental que el sistema sea capaz de invocar el "Protocolo de Misión" de cualquier vehículo sin importar su tipo, asegurando que cada uno ejecute su tarea específica (escanear o gestionar vida) según corresponda. "Protocolo misión" enviará siempre un mensaje a tierra y acto seguido ejecutará la tarea específica de cada vehículo.*

#### **2. Integridad Estructural (Propulsión)**

*Un vehículo no es nada sin su capacidad de moverse. Por normativa de seguridad, cada vehículo debe ser fabricado e integrado con un Sistema de Propulsión único (con datos de tipo de combustible y empuje máximo).*

*Nota de seguridad: El sistema de propulsión es una parte física indivisible del vehículo. Si un vehículo es dado de baja o destruido en el registro, su sistema de propulsión debe desaparecer con él de forma automática; no son piezas intercambiables entre naves una vez ensambladas.*

#### **3. Carga Útil y Logística**

*Para que una misión tenga éxito, los vehículos suelen transportar Módulos Científicos. Cada módulo cuenta con un instrumento principal (espectrómetros, taladros, etc.). A diferencia de los motores, estos módulos pueden ser asignados a diferentes vehículos según las necesidades de la investigación.*

*Finalmente, todo lo anterior debe ser coordinado bajo una Misión, la cual define un destino y un identificador único, supervisando el despliegue de los vehículos implicados.*

*Esperamos un diseño limpio que refleje estas jerarquías y dependencias.*

*Atentamente, director general de la ESA*

## Ejercicio 2

### **Sistema de Optimización de Recursos Orbitales (SORO)**

*De: Director del Departamento de Operaciones en Tierra (DOCC)*

*Para: Equipo de Desarrollo de Software*

*Asunto: Licitación interna: Sistema de reserva y telemetría unificada*

Buenos días,

*Debido al incremento de misiones comerciales en la ESA, el departamento de Operaciones en Tierra necesita un software que gestione el uso de nuestras Estaciones de Seguimiento (antenas en tierra) por parte de diferentes Agencias Cliente (NASA, SpaceX, JAXA, etc.).*

*El sistema debe cumplir con los siguientes pilares de lógica de negocio:*

#### **1. Estandarización de Telemetría**

*Para que cualquier componente sea compatible con nuestra red, debe cumplir con un estándar de comunicación. Todo objeto que desee enviar datos a tierra debe ser capaz de conectar(), transmitirPaquete() y desconectar(). No nos importa cómo lo haga internamente una antena o un satélite, pero el sistema debe garantizar que todos sigan este estándar.*

#### **2. Gestión de Estaciones y Reservas**

*Necesitamos registrar qué Agencia Cliente está utilizando qué Estación de Seguimiento.*

*Una estación no se asigna permanentemente; se generan Registros de Uso (Booking).*

*Cada Registro de Uso debe almacenar la fecha de inicio, fecha de fin y el coste calculado.*

*Este modelo debe permitirnos consultar el historial: qué agencias usaron una antena específica y qué antenas ha alquilado una agencia a lo largo del tiempo.*

#### **3. Infraestructura de Estaciones**

*Existen dos tipos de estaciones en nuestra red:*

- *Estaciones de Radiofrecuencia: Utilizan una frecuencia específica y tienen un diámetro de parábola.*
- *Estaciones Ópticas (Láser): Utilizan una longitud de onda específica y requieren condiciones de visibilidad atmosférica. Ambas comparten características comunes como su ubicación geográfica y su estado operacional, pero su método de mantenimiento es totalmente distinto.*

#### 4. Componentes Críticos

*Cada Estación de Seguimiento depende intrínsecamente de una Unidad de Procesamiento de Datos de alto rendimiento. Si la estación se desmantela, esta unidad (específica para el hardware de esa antena) se retira del servicio inmediatamente. Además, las estaciones pueden tener asociados varios Sensores Meteorológicos para decidir si el enlace es viable.*

## Ejercicio 3

### **Proyecto "Fénix" - Red de Respuesta a Amenazas Near-Earth (NEO)**

*De: Directora de Seguridad Planetaria y Defensa Espacial (D-SPD)*

*Para: Arquitectos de Sistemas Senior*

*Asunto: Especificación del Sistema de Intercepción y Desvío de Asteroides*

*Estimados colegas,*

*La detección del asteroide 2025-EX4 nos ha obligado a acelerar el Proyecto Fénix. Necesitamos un software que gestione nuestra Red de Defensa Planetaria. Este sistema es de una complejidad crítica y debe integrar hardware heterogéneo bajo condiciones extremas.*

#### **1. El Protocolo de Actuación**

*Cualquier activo en el espacio (sea un satélite, un misil cinético o un láser orbital) debe ejecutar los métodos evaluarAmenaza(), fijarObjetivo(Coordenadas c) y activarContramedida().*

#### **2. Jerarquía de Activos Espaciales**

*Operamos con una base común de ActivoDefensa (con atributos como id, nivelEnergia y posicionActual). Sin embargo, existen dos ramas tecnológicas muy distintas:*

- *Interceptores Cinéticos: Su objetivo es colisionar. Tienen un atributo de masa y velocidadRelativa. Su mantenimiento requiere recalibrar los MotoresDeManiobra.*
- *Plataformas de Energía: Utilizan proyectores de iones o láseres. Tienen un atributo de longitudDeOnda y tiempoCarga.*

#### **3. Sistema de Redundancia Crítica**

*Cada ActivoDefensa no es una entidad única, sino un clúster. Debe estar compuesto por:*

- *Un NúcleoIA (procesador de decisiones). Si el núcleo falla, el activo queda inútil.*

- Al menos dos Matrices de Sensores. La relación es tan fuerte que el software debe instanciar los sensores en el momento que se crea el activo; no pueden existir sensores "sueltos" en el espacio.

## 4. Lógica de "Alquiler" y Despliegue

La ESA no siempre es dueña de los activos; a veces los "alquila" o solicita a Naciones Aliadas. Necesitamos una clase de asociación llamada DespliegueOperativo.

Esta clase vincula a una NaciónAliada con un ActivoDefensa.

Debe registrar el sectorEspacial asignado, el costeOperativoPorHora y el estadoDeAlerta (Verde, Amarillo, Rojo).

Un activo puede haber tenido múltiples despliegues en diferentes sectores a lo largo de su vida útil, y necesitamos el historial completo para auditorías de defensa.

## 5. El Factor Externo: Objetos NEO

El sistema debe gestionar una clase Asteroide (con atributos como diametro, composicion y trayectoria). Un DespliegueOperativo se activa específicamente para mitigar un Asteroide concreto.