

Procesos de Desarrollo de Software

Domain-Driven Design

Sesión Práctica 02



Contenidos

1. Domain Driven Design.
2. Caso práctico.
3. Resultados.

Domain Driven Design

Repasar conceptos y definir estrategias para resolver el caso práctico



Proceso de desarrollo de software 2025/26

1. DDD – Pasos

Resolveremos el ejercicio en los siguientes pasos:

1.- Desarrollaremos un modelado básico inicial con:

- Términos de **lenguaje ubicuo**
- **Modelo del dominio** “ligero” con cualquier diagrama útil
- Historias de usuario

2- Realizaremos diseño táctico según DDD:

- Identificar **entidades** / **objetos valor**
- Identificar **agregados**
- Asignar reglas de negocio / **servicios de dominio**
- Identificar **servicios de aplicación**



Caso práctico

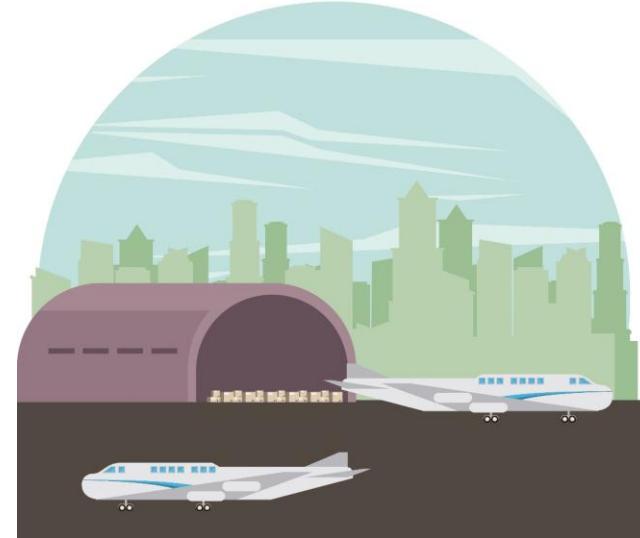
Aeródromo



Proceso de desarrollo de software 2025/26

2. Caso práctico – Enunciado

Un aeródromo quiere actualizar su sistema de gestión para la asignación de mecánicos a posibles averías reportadas en los aviones.



Fuente: <http://www.freepik.com>. Diseñado por jemastock



2. Caso práctico – Enunciado

Cada avión del aeródromo tiene información como número de registro, matrícula, antigüedad, fecha de registro y última revisión. Además, cada avión es de un tipo determinado, recogiéndose de cada tipo su marca y modelo, capacidad y peso.



Fuente: <http://www.freepik.com>. Diseñado por jemastock



2. Caso práctico – Enunciado

Cada avión está asignado a un propietario, que será el encargado de gestionar y actualizar su información y la de sus aviones.



Fuente: <http://www.freepik.com>. Diseñado por jemastock



2. Caso práctico – Enunciado

En el aeródromo hay dos tipos de empleados:
mecánicos y pilotos.

Mientras que los pilotos pueden pilotar cualquier tipo de avión, cada mecánico está cualificado para reparar sólo determinados tipos de aviones.



2. Caso práctico – Enunciado

Según el protocolo de aviación internacional, el piloto comprobará que todo funciona correctamente y en caso de detectar un fallo en el avión lo reportará para que un mecánico pueda repararlo.



2. Caso práctico – Enunciado

En el mismo momento en que se reporta la avería, el sistema seleccionará el mecánico libre más adecuado al tipo de avión y de avería.

- Este mecánico será el encargado de la reparación de dicho avión inmediatamente.
- Si no hay ningún mecánico libre, la avería se asigna a uno de los mecánicos cualificados para resolverla, quedando en cola hasta su resolución.
La avería se asigna al mecánico cualificado para resolverla que tenga menos averías pendientes de resolución, y en caso de empate, a uno cualquiera.



Resultados

Possible solución al problema

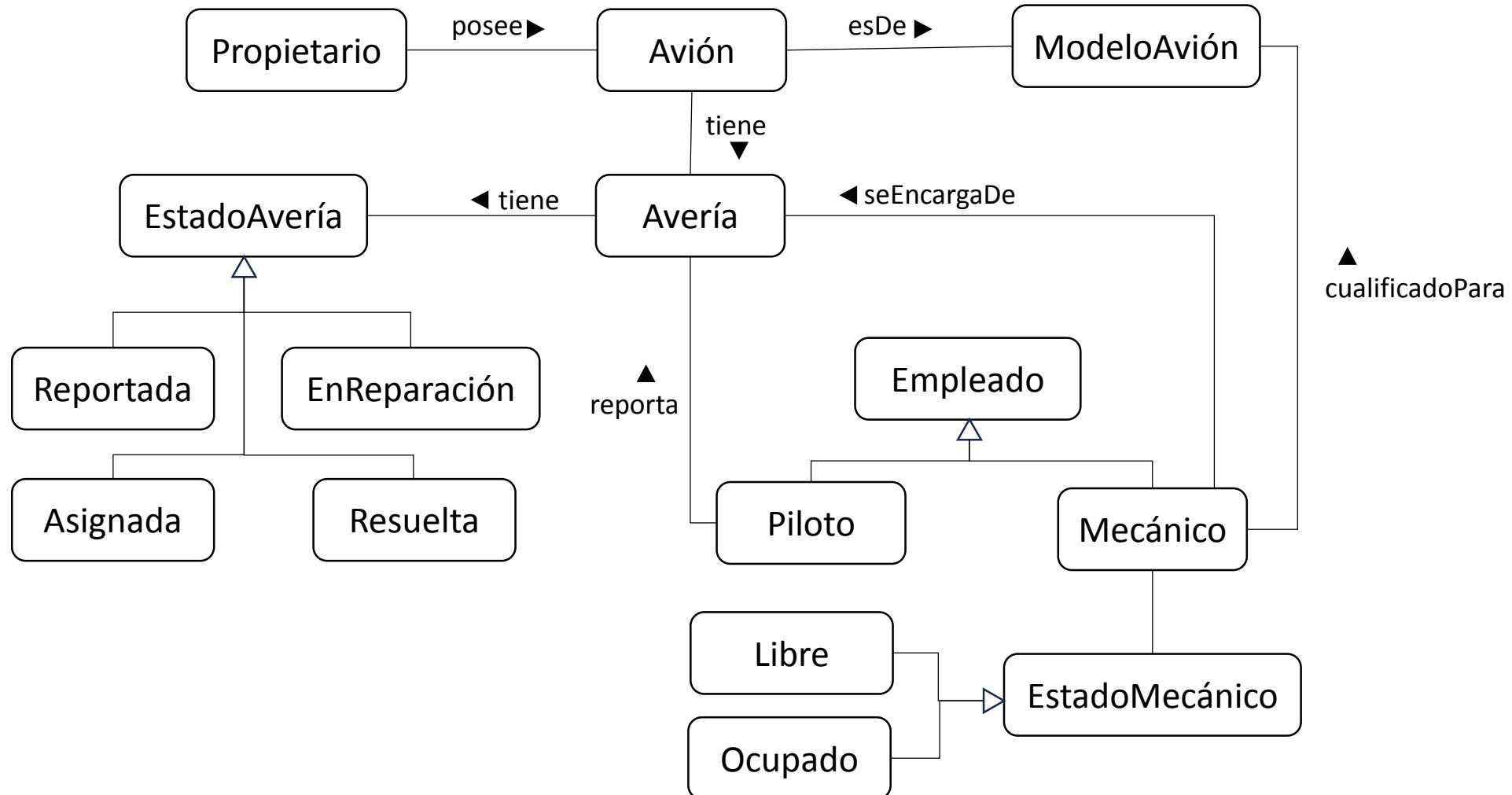


Proceso de desarrollo de software 2025/26

3. Resultados – Lenguaje ubicuo

Término	Descripción
Aeródromo	Instalación donde se gestionan aviones, empleados y reparaciones.
Avería	Problema detectado en un avión que requiere una reparación.
Avión	Aeronave registrada en el aeródromo que puede sufrir averías.
Mecánico	Empleado cualificado para reparar determinados tipos de avión.
Tipo de avión	Clasificación de un avión según marca, modelo y características.
Propietario	Persona responsable de gestionar la información de sus aviones.
Piloto	Empleado que puede pilotar cualquier avión y reportar averías.
Avería reportada	Avería informada por un piloto.
Avería en reparación	Avería a la que ya se ha enviado un mecánico.
Avería asignada	Avería asignada a un mecánico pero que aún no ha comenzado a repararla.
Avería resuelta	Avería que ya se ha resuelto por un mecánico.
Mecánico libre	Mecánico que no está reparando ninguna avería.

3. Resultados – Modelo conceptual



3. Resultados – Historias de usuario

- **Como** propietario **quiero** gestionar la información de mi avión **para** que el aeródromo tenga los datos actualizados sobre el mismo.
- **Como** piloto **quiero** reportar una avería en un avión **para** que se asigne un mecánico que la repare.
- **Como** mecánico **quiero** indicar que he reparado una avería **para** que quede constancia de mi trabajo e indicar que quedo libre.

Nota:

Faltarían los criterios de aceptación.
¡Recordad que toda H.U. debe tenerlos!

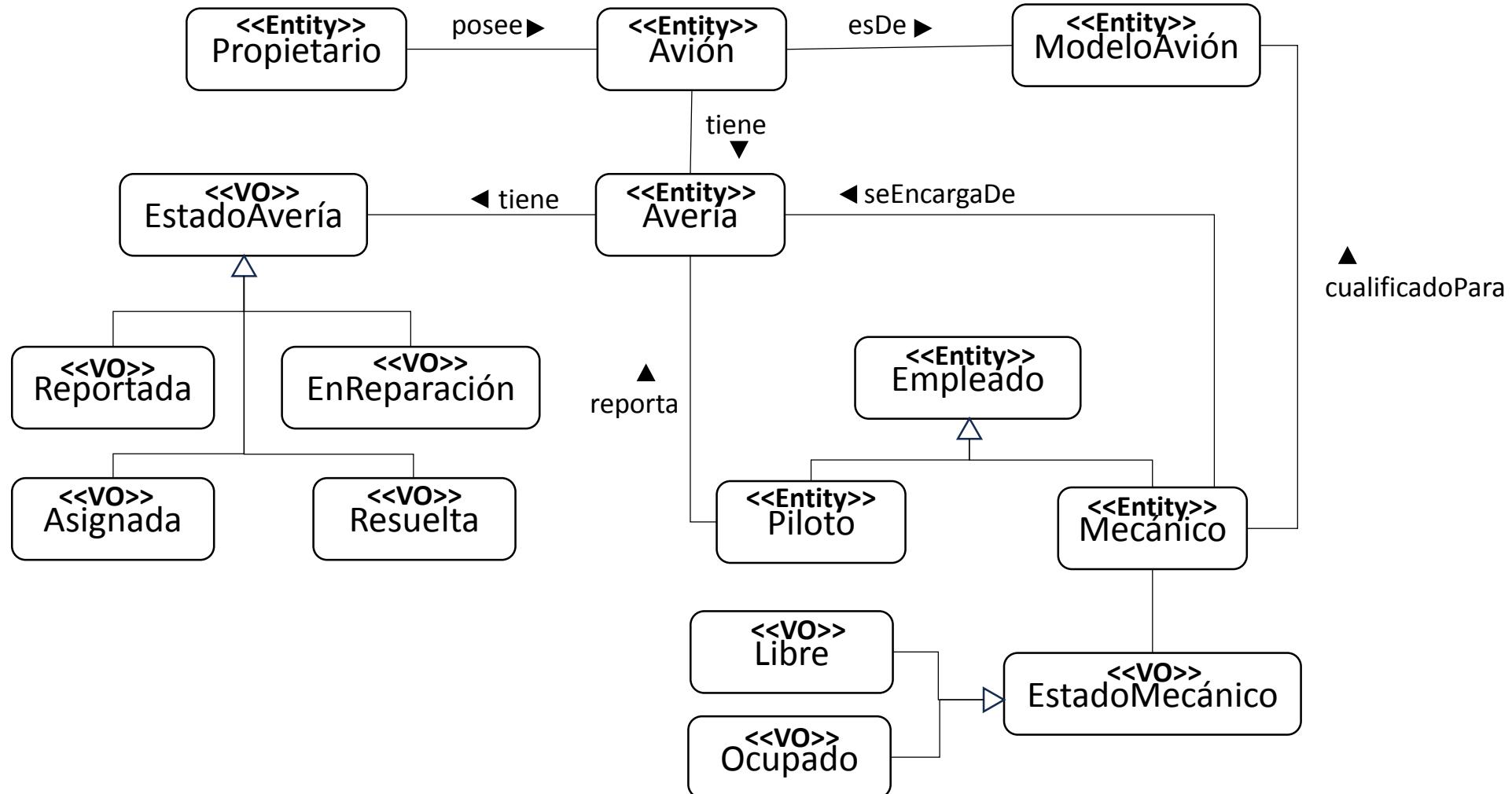
3. Resultados – Reglas de negocio

- **R1** – Un mecánico solo puede reparar tipos de avión para los que esté cualificado.
- **R2** – Cuando un piloto reporta una avería, el sistema debe asignar un mecánico libre inmediatamente.
- **R3** – Un mecánico no puede reparar más de una avería al mismo tiempo.
- **R4** – Si no hay ningún mecánico libre adecuado, la avería queda en cola.

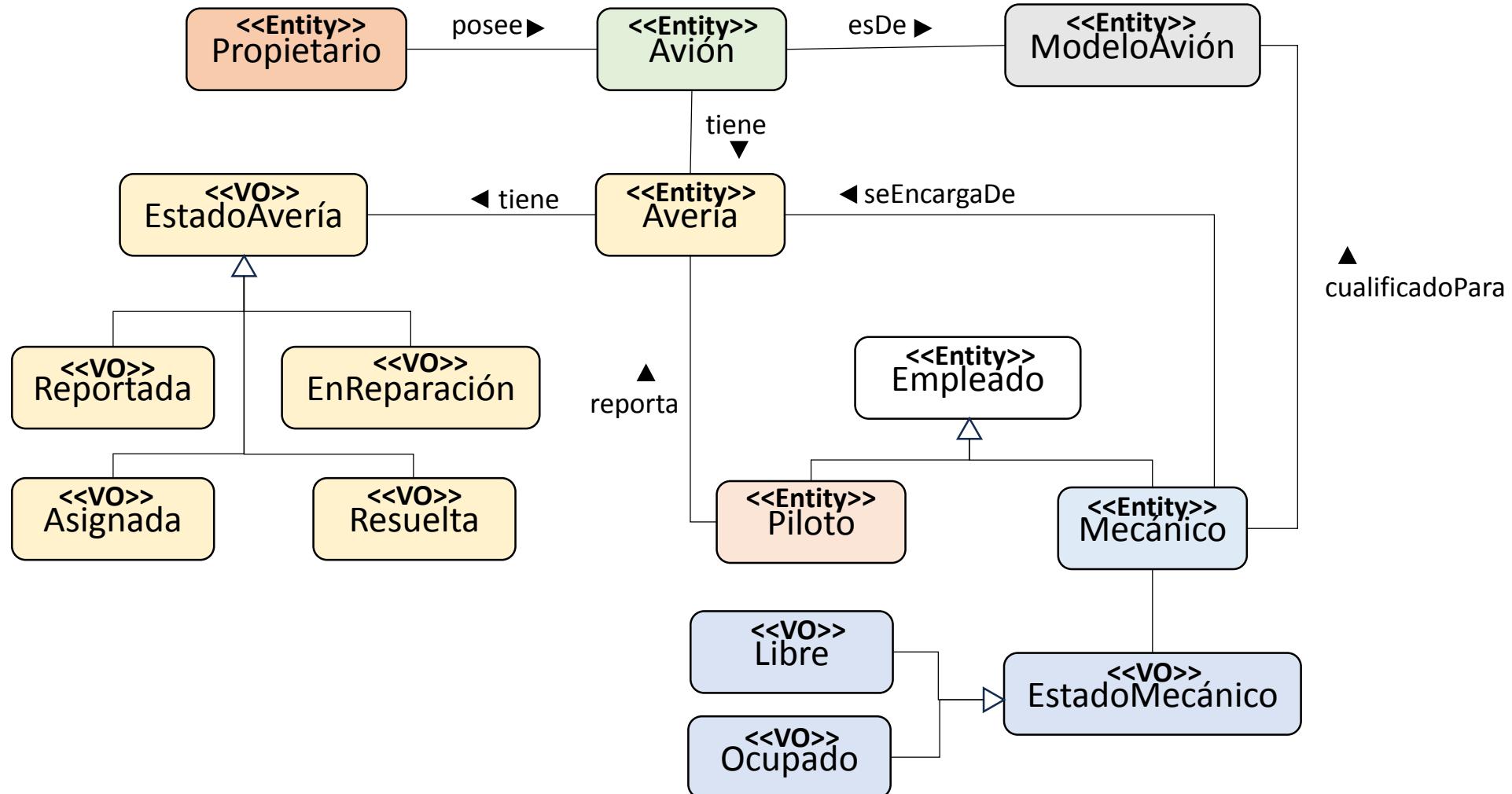
3. Resultados – Reglas de negocio

- **R5** – La avería se asigna al mecánico cualificado con menos averías pendientes de resolución.
- **R6** – En caso de empate entre mecánicos, el sistema selecciona uno de forma aleatoria.
- **R7** – Cuando un mecánico inicia una reparación, su estado pasa a ocupado.
- **R8** – Cuando una reparación finaliza, la avería pasa a estado resuelta y el mecánico queda libre.

3. Resultados – Entidades y objetos valor



3. Resultados – Agregados



3. Resultados – Servicios de dominio

```
public interface SeleccionMecanicos{  
    public Mecanico seleccionarMecanico(  
        ModeloAvionId modeloAvionId,  
        RepositorioMecanicos repositorioMecanicos  
    );  
}
```

3. Resultados – Servicios de dominio

```
public class SeleccionMecanicosPorMenosAveriasPendientes implements SeleccionMecanicos {  
    public Mecanico asignarMecanico(  
        ModeloAvionId modeloAvionId,  
        RepoMecanicos repositorioMecanicos  
    ) {  
        // @todo implementación  
        return MecanicoId;  
    }  
}
```

Preguntas para la implementación

- ¿Cómo sabemos si un mecánico está libre o no?
- ¿Cómo sabemos si un mecánico puede arreglar este tipo de avería?

3. Resultados – Servicios de dominio

```
public class SeleccionMecanicosPorMenosAveriasPendientes implements SeleccionMecanicos {  
    public Mecanico asignarMecanico(  
        ModeloAvionId modeloAvionId,  
        RepoMecanicos repositorioMecanicos  
    ) {  
        // Buscamos mecánicos cualificados  
        List<Mecanico> candidatos = repositorioMecanicos.buscarLibresCualificadosPara(  
            modeloAvionId  
        );  
  
        // Ordenamos los mecánicos  
        candidatos.ordenarPorMenosAveriasPendientes()  
  
        return candidatos[0].primero().id  
    }  
}
```