

Apellido y Nombre: Legajo:

Depósito automatizado

En esta ocasión nos contrataron para diseñar un sistema de gestión de depósitos. En particular nos encargaremos de despachar los pedidos desde su lugar de almacenamiento al transporte mediante vehículos robotizados.



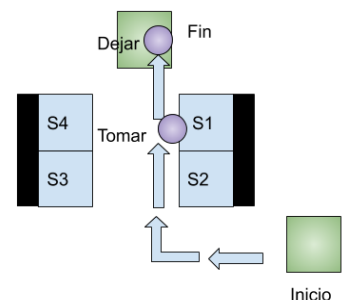
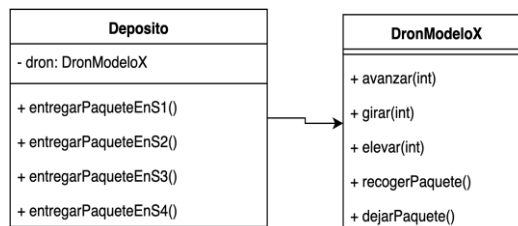
1. Un depósito es un edificio que almacena mercaderías que están alocadas en compartimientos estándares sobre los estantes/gondolas. Una sola mercadería se almacena dentro de un único compartimiento. Cada depósito tiene una disposición de compartimientos distinta.
2. Los depósitos reciben pedidos de los clientes por mercaderías que solicitaron, que terminan conformando un paquete que se despacha a los transportes para que luego los estos los entreguen.
3. Los pedidos se atienden por orden de llegada y la atención implica:
 - Asignar a un robot la tarea de buscar una mercadería;
 - Asegurarse que toda la mercaderías listadas en el pedido estén en el lugar de consolidación;
 - Avisar al transportista solo cuando el paquete está completo.
4. Un robot sabe ejecutar acciones básicas como desplazarse en distintas direcciones, agarrar y soltar la mercadería. Por ahora tenemos 2 tipos de robots en nuestros depósitos que son: clark y drone. Tener en cuenta que los controladores de cada robot no manejan una interfaz común para su comunicación.
5. Un robot para buscar la mercadería recibe una secuencia de órdenes preestablecidas, porque en sí mismo no es capaz de resolver el recorrido.
6. El recorrido consiste salir de un punto de inicio designado, llegar hasta el compartimiento, recoger la mercadería, ir al punto de consolidación, dejar la mercadería y volver al punto de inicio.
7. Los recorridos dentro de un depósito están preestablecidos al momento de iniciar el sistema. Hay que proveer facilidades para su configuración.
8. Hay N robots por deposito y solo pueden estar afectados a un recorrido por vez.
9. Un robot ante la imposibilidad de avanzar por haber detectado un obstáculo debe poder realizar una de las siguientes acciones: (1) esperar N unidades de tiempo y reintentar. (2) quedarse quieto y llamar a mantenimiento; (3) esquivar obstáculo (función nativa de cada robot); (4) volver a punto de inicio.

Estas acciones son de *failover* y se configura de antemano para el armado de paquete (recolección de todas las mercaderías).

- Se realizó una prueba de concepto **exitosa** en un escenario reducido (figura de la derecha y diagrama de clase Deposito-DronModeloX), con posiciones fijas y solo un tipo de robot. Ahora hay que adaptarla para cumplir con los nuevos requerimientos que surgen del dominio planteado.

#Depósito>>entregarPaqueteEnS1 (prototipo)

```
dron.avanzar(10)
dron.girar(-90)
dron.avanzar(10)
dron.recojerPaquete(10)
dron.avanzar(10)
dron.dejarPaquete(10)
// [...] instrucciones con recorrido inverso para volver
```



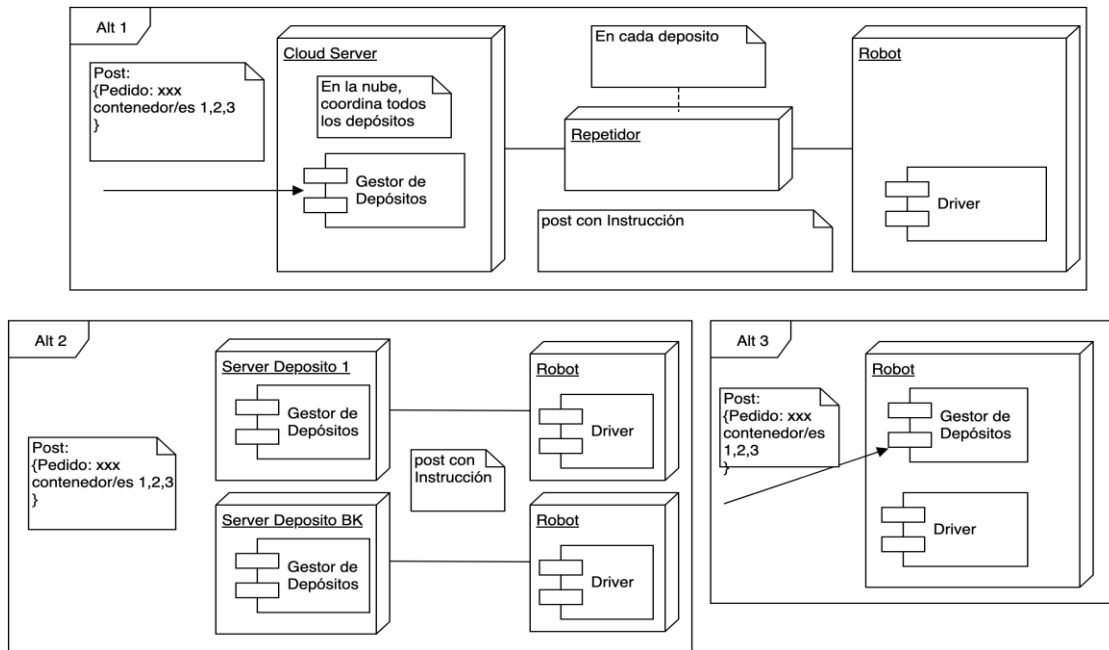
- Queda fuera del alcance el control del stock y entonces damos por hecho que los compartimientos siempre tienen la mercadería solicitada.

- Como sabemos que los recorridos para cada compartimiento es complejo, repetitivo y se arman manualmente queremos facilitar la creación de los mismos.



- Los robots son marca ACME, esto quiere decir que no fallan, se mueven precisamente donde se les indica, su coeficiente de rozamiento con el aire es 0, etc...

Punto 1 - Arquitectura (30 Puntos)



Compare los siguientes despliegues utilizando los atributos de calidad.

Escoja cual utilizaría y agregue cualquier detalle que considere. Asuma que los robots tienen todo el hardware, sensores y SO necesario para hacer todo lo que se les pide.

Punto 2 - Modelado en Objetos (40 Puntos)

- Modelar el dominio presentado en el contexto, utilizando el paradigma orientado a objetos. Comunique su solución mediante un diagrama de clases (obligatorio), pseudocódigo, otros diagramas y/o prosa (complementario).
- Realice diagrama de secuencia para comunicar cómo se resuelve el despacho de un paquete correctamente y otro de un escenario donde se aplique la estrategia de failover de tipo "volver a punto de inicio". Anotar todas las suposiciones que considere necesarias.

Punto 3 - Persistencia (30 Puntos)

Utilizando un DER (obligatorio), código anotado y/o prosa (complementario), se pide explicar cómo persistir el modelo de objetos del punto anterior en una base de datos relacional, indicando claramente:

- Qué elementos del modelo es necesario persistir.
- Las claves primarias, las foráneas y las restricciones según corresponda.
- Si fueran necesarias, estrategias de mapeo de herencia utilizadas. Explicar por qué fueron elegidas y compararlas con alguna otra. También las estructuras de datos que deban ser desnormalizadas, si corresponde.
- Justificaciones sobre las decisiones de diseño tomadas anteriormente.

Por otra parte, nos han solicitado que se debe armar un reporte para medir la eficacia del despacho de pedidos. Esto implica que hay que conocer la cantidad de mercadería despachada por cada tipo de robot por semana/mes/año y también los totales de pedidos completos en los que participó. Se estima que este requerimiento puede tener problemas de rendimiento, aún optimizando los índices de la tablas en la base relacional. De alternativas de solución y justifique.