Justificaciones de Diseño (2.2)

La clase DispositivoDeCatalogo se modeló como clase concreta para que sea extensible en tiempo de ejecución. De esta manera, se logra el dinamismo pedido.

Marca se modeló de manera análoga.

Categoria se modeló como clase concreta dado que el enunciado menciona que deben ser creadas por el administrador, de esta manera es extensible en tiempo de ejecución (esto no podría lograrse con un enumerado) y soluciona el problema de consistencia de datos.

DispositivoConcreto se modeló de manera análoga.

Para la sección de “Proceso de Alquiler”, se tuvo especialmente en cuenta el principio de responsabilidad única. De esta manera, se tienen varias clases con una única responsabilidad tales como el Notificador, el VerificadorDeSolicitudDeAlquiler, el GestionadorOrdenDeAlquiler y el EnviadorSolicitud. De esta manera, se evita un alto acoplamiento y una clase Dios, además de una alta cohesión, si lo comparamos contra un diseño alternativo en donde todas estas responsaiblidades estuvieron dentro de una sola clase.

La clase EstadoOrden fue modelada como enum dado que en ninguna parte del enunciado se menciona que debe ser extensible en tiempo de ejecución. De esta manera, supongo una baja frecuencia al cambio.

CondicionAnteIva se modeló de manera análoga.

TipoMedioDeContacto se modeló de manera análoga.

FormaDeRetiro se modeló de manera análoga.

El método procesar(SolicitudDeAlquiler) del ProcesadorSolicitudDeAlquiler es asincrónico, se crea un hilo de ejecución por cada solicitud que quede a la espera de una respuesta por parte del proveedor externo. Se encarga de procesar las solicitudes que no han sido procesadas hasta el momento.

Cada instancia de ReceptorPosicion es un gemelo digital de un dispositivo concreto, el cual sólo se encarga de recibir la posición y enviársela al GestionadorDeAlerta.

El GestionadorDeAlerta verifica si el dispositivo se encuentra fuera de rango (para esto es necesario que utilice el repositorio de la OrdenDeAlquiler). En caso de estarlo, instancia la clase AlertaFueraDeRango llamando al método estático builder crearAlertaFueraDeRango().

Arquitectura (1.2)

El proceso de asignación de dispositivos a una orden se está realizando de manera sincrónica.

La alternativa sería realizarlo de manera asincrónica. Ejemplo: teniendo un cronJob que ejecute cada cierta cantidad de horas y realice el proceso de asignación.

Arquitectura (1.3)

La propuesta que me parece más adecuada es que integrarse mediante API REST utilizando un mecanismo asincrónico push based de manera tal que no se deba esperar una respuesta inmediata por parte del sistema externo y que simplemente avise cuando el envío se haya completado.

Le debería enviar la dirección del cliente, la dirección del local, la fecha y hora en que deben pasar a buscar los dispositivos concretos y una dirección de callback así me puede notificar cuando se haya completado.