

## Ejercicios de Aplicación Modificabilidad, Patron Pipes and Filters

### Ejercicio 1.

Una entidad Device está empleando una librería de terceros denominada Port en el diagrama de abajo. Como se ve en el modelo y código adjunto.

Al emplear el método *open()* expuesto por la librería Port, se observa que el código de Device (se adjunta una porción del código) queda gobernado por el manejo de errores, que además, contendrá duplicaciones, ya que es probable que muchas excepciones tengan el mismo tratamiento (por ejemplo *loggear el error*, etc).

Debido a que el manejo de errores en la comunicación entre subsistemas suele ser de relevancia arquitectónica, se pide:

- Proponga una solución para “limpiar” el código de manejo de errores de la entidad Device de forma tal que no tenga que conocer todas y cada una de las excepciones de la librería de terceros. Diagrame una solución que muestre todas las entidades de la nueva solución.

**Tenga especial cuidado en el uso de la notación**

- Mencione qué atributo de calidad consideró y que tácticas aplicó al implementar la solución requerida en a).

**En el diagrama que realizó para la parte a) indique qué táctcias está utilizando.**

*Nota: Puede asumir que todas las excepciones son “checked”.*

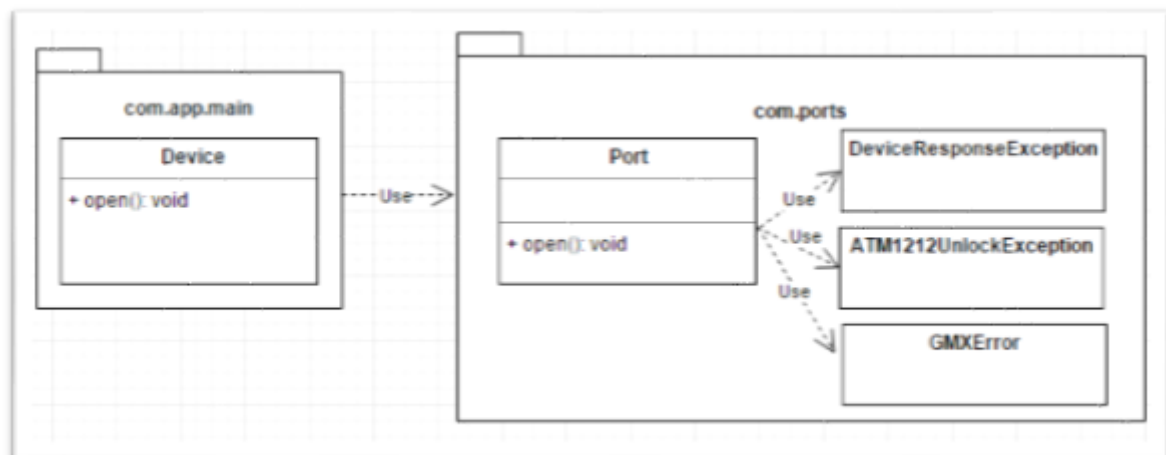


Figura 1 - Modelo de entidades



```
Device.java - C_CODE - Visual Studio Code
Archivo Editar Ver Ir a Ayuda

Device.java
1
2 public class Device{
3     private Port port;
4     public Device(){
5         port = new Port(23); // Port es una Libreria de terceros
6     }
7
8     public void open(){
9         try{
10
11             port.open();
12
13         } catch (DeviceResponseException e) {
14         } catch (ATM1212UnlockedException e) {
15         } catch (GMXError e) {
16         } finally {
17             //...
18         }
19     }
20 }
```

Figura 2 - Bosquejo del código de la entidad Device

## Ejercicio 2.

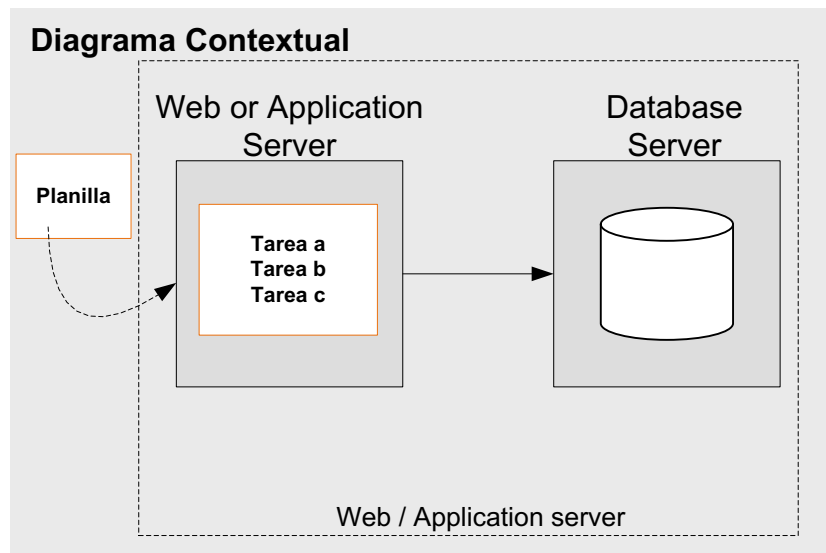
Una aplicación web requiere procesar información de variada naturaleza y complejidad sobre la información que recibe.

En particular, se trata de una planilla de excel con varios GB de información que es cargada a una aplicación web. La planilla tiene como propósito proporcionar la información para mover empleados de un edificio a otro.

Un enfoque simplista de la solución podría ser generar un componente que encapsule todas las tareas concretas que se realizan sobre la planilla. Sin embargo, se ha notado que es un enfoque poco flexible en términos de reutilización de código y potencial problemas de performance de la solución.

Las tareas en concreto son las siguientes:

- Leer la planilla
- Validar la información recibida con los datos en una base de datos y generar un archivo de errores
- Transformar la información válida en formato XML
- Agregar información de otra base de datos
- Generar una nueva planilla de excel



**Se pide:**

- a) Encuentre un patron arquitectonico que le permita dar soporte a la reutilizacion de las tareas descritas y discuta las mejores alternativas para favorecer la performance de la solucion. **Diagrame la estructura del patron aplicado en funcion de las tareas. Utilice la vista que sea más apropiada y utilice diagramas UML correspondientes a la vista.**

**Ejercicio 3.**

Una empresa dedicada al procesamiento de datos para terceros desea implementar un nuevo sistema para la impresión de los recibos de sus clientes.

Los pasos que se piensan seguir para el procesamiento de los recibos es el siguiente:

1. El cliente enviará los datos de los recibos a imprimir. Los datos podrán ser enviados en varios formatos (txt, Excel, tablas de base de datos, etc.)
2. Los datos recibidos se transformarán a un formato XML que cuenta con tags específicos para el procesamiento e impresión de los recibos.
3. El contenido de los XMLs se procesará para la impresión agregando información necesaria para la impresión del mismo.
4. El XML se procesará para imprimir los recibos en determinado formato. Actualmente se manejan tres tipos de formato de recibos.

Otras consideraciones que se desean contemplar son:

1. En el futuro se aceptarán nuevos tipos de formatos de archivo para que envíen los clientes
2. En el futuro se impriman recibos en otros formatos y posiblemente se incorpore un servicio de publicación de recibos impresos en formato digital para que los clientes bajen las imágenes y las impriman.
3. Se desea que se contemple la eficiencia y confiabilidad sobre otros atributos de calidad, en particular:
  - a. Que los componentes que comprendan el procesamiento de los recibos se puedan distribuir entre distintos procesadores.
  - b. Se desea que la información que se maneja entre los distintos componentes no se pierda ante el fallo de uno de los componentes.

- A. ¿Qué estilo o patrón de arquitectura recomendaría para estructurar este sistema?
- B. Describa la topología del estilo aplicándola al problema planteado
- C. Describa como puede asegurar que la información que se maneja entre los distintos componentes no se pierda ante un fallo