

Patrones de Diseño

Modalidad

Actividad Grupal

Grupo

8

Integrantes

AKIYOSHI, Joel
ROISMAN ECHART, Gonzalo
NORIEGA, Patricio Ezequiel

Ejercicio 1

En una empresa existen dos tipos de empleados:

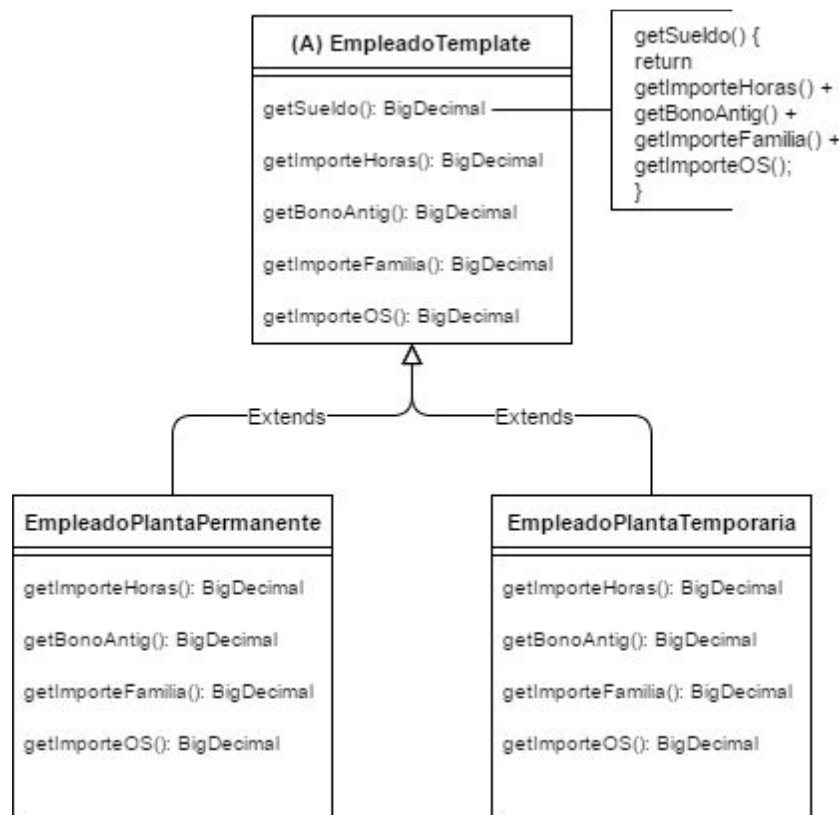
- Los empleados de planta permanente que cobran la cantidad de horas trabajadas por \$30, más un bono por antigüedad y un monto fijo por salario familiar. Asimismo se les aplica un descuento por obra social.
- Los empleados de planta temporaria que cobran la cantidad de horas trabajadas por \$20, más salario familiar (no cobran antigüedad ni se les descuenta obra social).

Empleando al menos 1 patrón de diseño, defina una jerarquía de empleados que:

- Sea polimórfica respecto al cálculo del sueldo.
- Defina una única manera de calcular el sueldo.
- Sea extensible para agregar futuros tipos de empleados.

Nota: La respuesta puede expresarla con un diagrama de clases o en código fuente Java.

Resolución con uso del patrón Template Method:



Ejercicio 2

Se debe diseñar un sistema que simula un sistema de archivos (filesystem) para un sistema operativo. Se trabajará con archivos y carpetas. Las carpetas pueden contener archivos u otras carpetas.

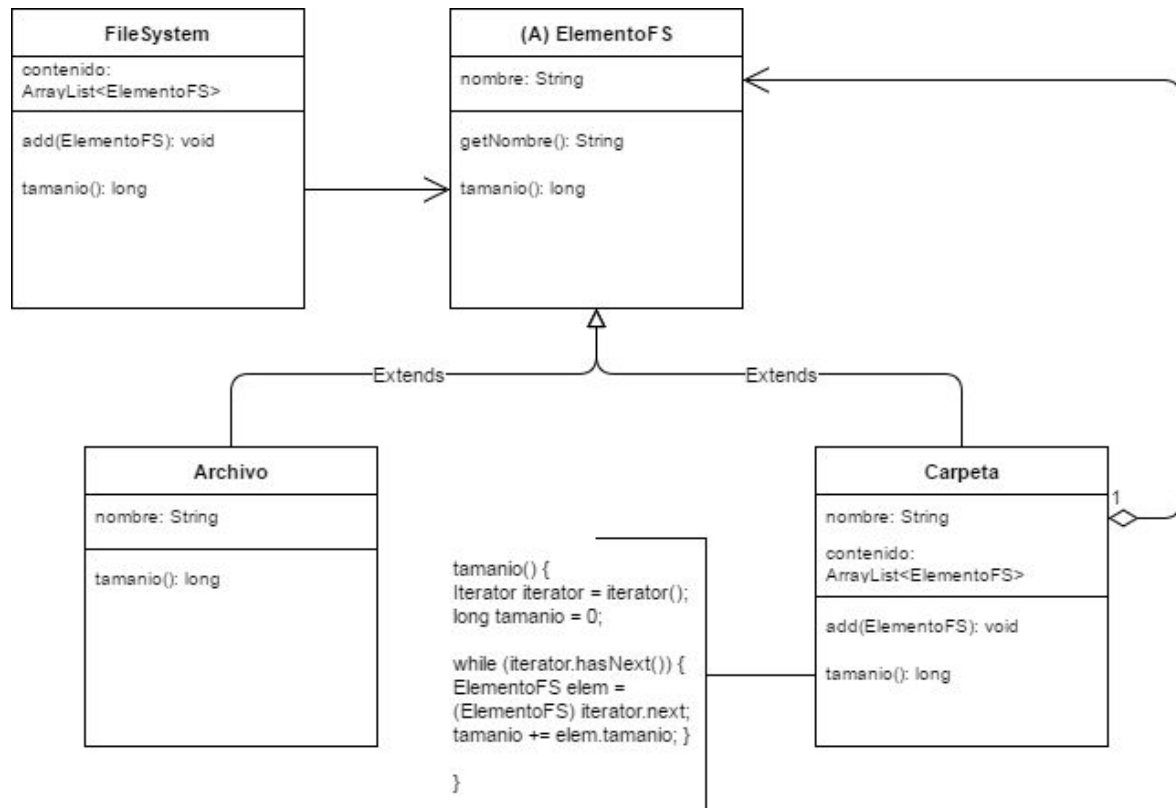
Se desea contar con el método `tamano()` donde:

- Para un archivo, es el tamaño del mismo.
- Para una carpeta, es el tamaño de los archivos que contiene.

¿Cómo lo diseñaría? ¿Es apropiado para el caso aplicar algún patrón de diseño?

Nota: La respuesta puede expresarla con un diagrama de clases o en código fuente Java.

Resolución con uso del patrón Composite:



Ejercicio 3

Se desea modelar el sistema operativo de un teléfono móvil. El equipo debe responder a los siguientes eventos:

- Llamada entrante.
- Corte llamada.
- Llamada rechazada.
- Corte llamada participante.

Mientras una llamada está establecida, puede recibir otras llamadas. En ese caso, las opciones son:

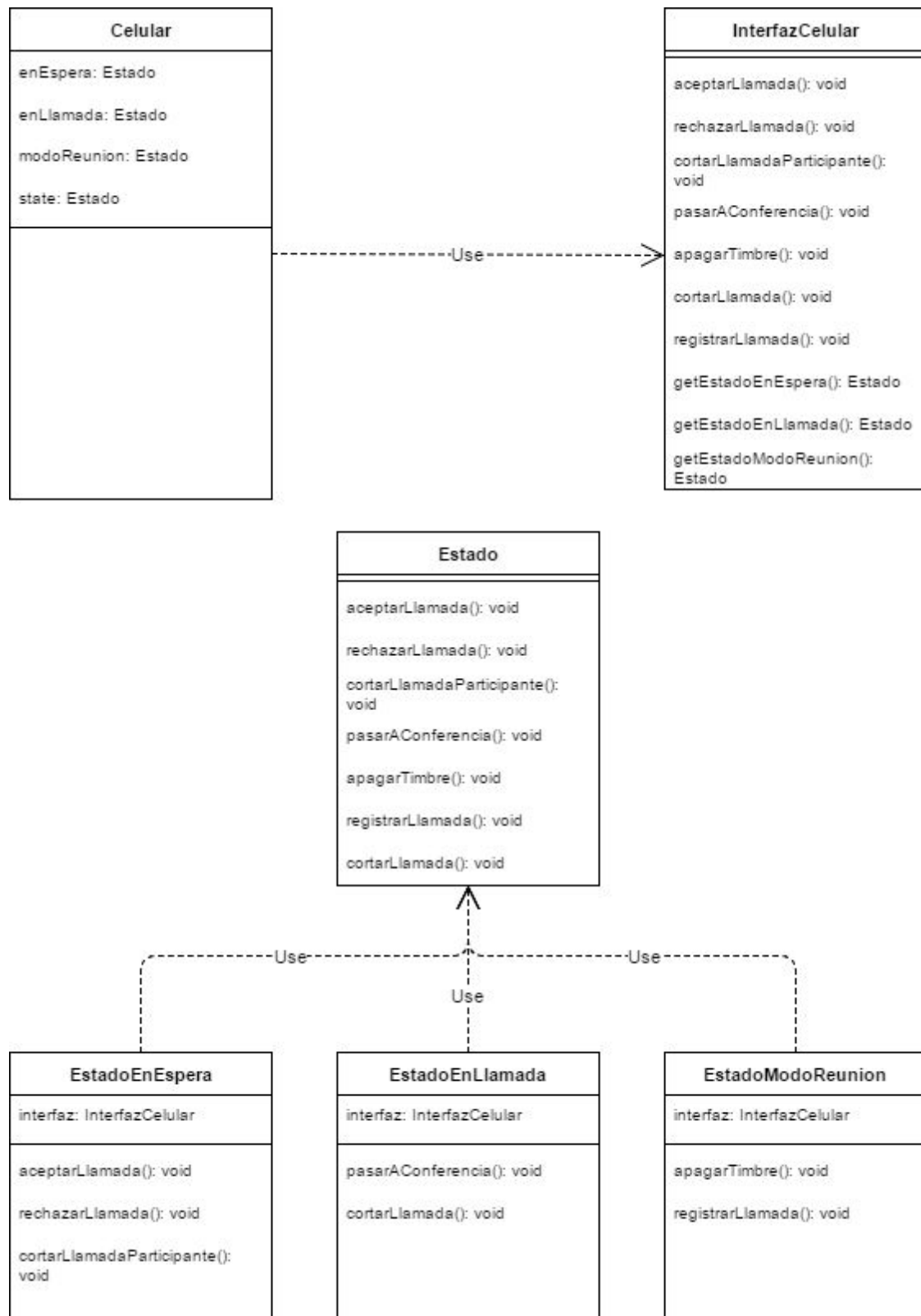
- Pasar a conferencia tripartita.
- Rechazar la llamada.
- Si se pasa a conferencia, las llamadas siguientes se rechazan.

Modo reunión:

- Se apaga el timbre.
- No se reciben llamadas; son registradas para que el usuario las vea más tarde.
- Si el usuario rechaza 3 llamadas consecutivas, el teléfono automáticamente se pone en modo reunión.
- Si el usuario realiza una llamada durante el modo reunión, el teléfono vuelve al modo normal.

¿Qué patrón de diseño emplearía para modelar este comportamiento? Explique mediante un diagrama de clases el diseño propuesto.

Resolución con uso del patrón State:



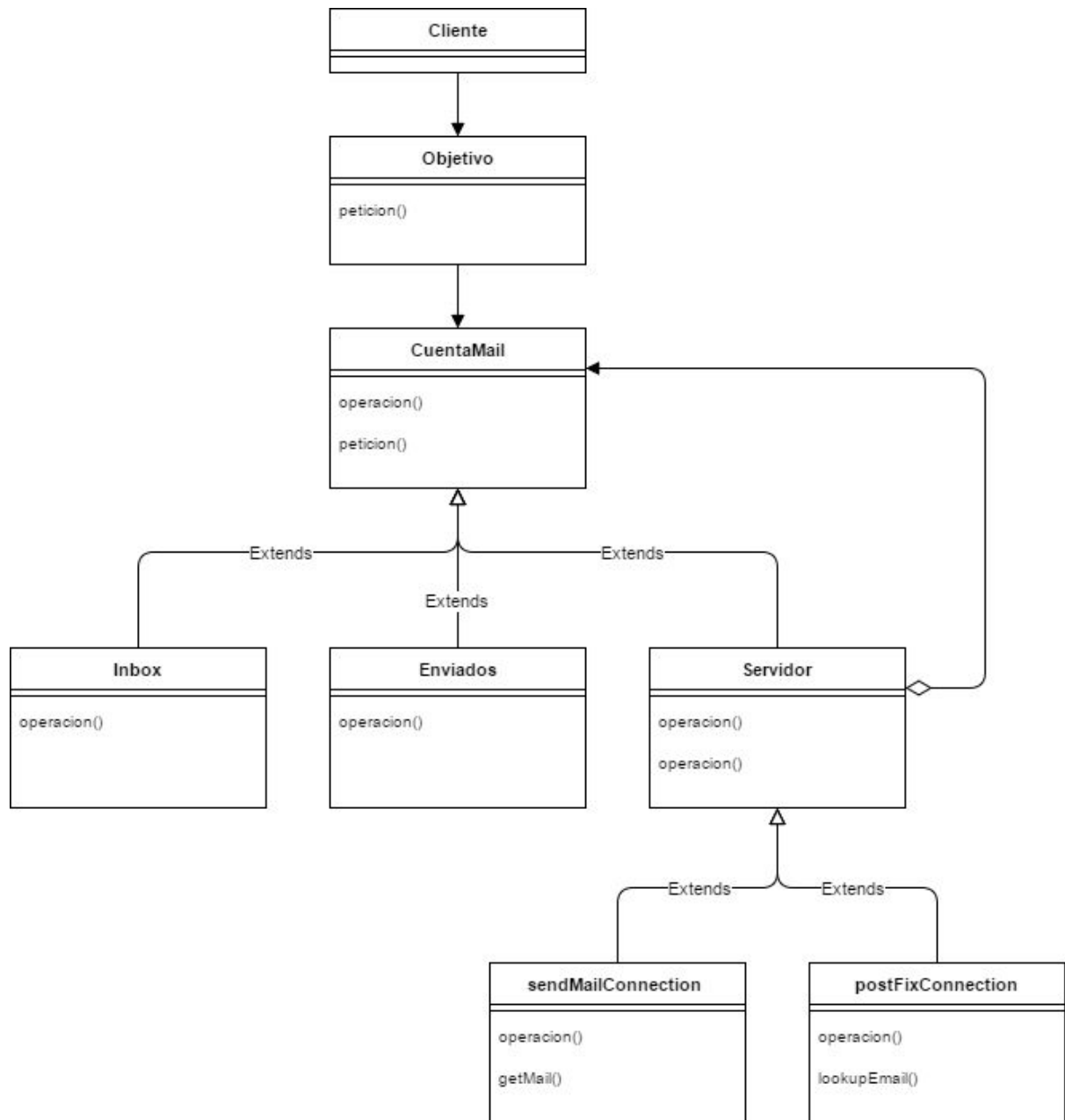
Ejercicio 4

El objetivo es diseñar un cliente de email (similar al Mozilla Thunderbird o MS Outlook). Este cliente manejará una única cuenta de mail. Debe permitir enviar, recibir, copiar y mover emails entre carpetas. Una cuenta tiene una carpeta de entrada predeterminada (Inbox) y una carpeta en la cual se almacenan los mensajes enviados (Sent Mails). Algunas de las características que deberá poseer son:

- El usuario puede exportar los mensajes y carpetas en diferentes formatos. Por ejemplo, exportar las carpetas a directorios y subdirectorios y los emails a archivos de texto dentro de los directorios, o exportar a un único archivo XML.
- Para descargar los mensajes, el cliente debe conectarse con el servidor de correo. Existen diferentes servidores de email o Mail Transfer Agents (MTA), por ejemplo, Sendmail y Postfix. Existen clases que se conectan directamente con estos servidores, la clase SendmailConnection y la clase PostfixConnection, pero ambas definen un método diferente para recibir email, por ejemplo: en SendmailConnection el método es getMail() y en PostfixConnection el método es lookupEmail(). Es deseable que el cliente de mail pueda usar cualquiera de estas clases de manera indistinta.

Proponer un diseño que pueda incluir uno o más Patrones de Diseño. Comunicar la solución mediante un diagrama de clases.

Resolución con uso del patrón Decorator:



Ejercicio 5

Se debe diseñar una aplicación móvil (app) para la venta de artículos de computación. La app deberá mostrar en un listado los artículos junto con una imagen, una descripción resumida y el precio. También debe permitir seleccionar uno o más artículos, especificar la forma de pago y envío una vez confirmada la operación de compra. Algunas de las funcionalidades que deberá ofrecer son:

- Listado de artículos: Los datos de los artículos a mostrar en un listado (por categoría o como resultado de una búsqueda) pueden provenir de diferentes fuentes de datos (servicios web de proveedores). El objetivo es poder recorrer de manera uniforme la colección de artículos recuperada.
- La app deberá permitir a los usuarios suscribirse en anuncios de ofertas. Para evitar el excesivo consumo de datos, se debe diseñar una solución donde sea el servidor quien notifique a los clientes (app) las promociones especiales y no los clientes quienes lo consulten si hay nuevas ofertas.

Modelar el sistema mediante el paradigma OO y expresar la solución con un diagrama de clases. Si lo considera apropiado, puede emplear Patrones de Diseño.

Resolución con uso de los patrones Iterator y Observer:

