SOFTWARE PARA ADMINISTRACIÓN DE ALQUILER DE VESTUARIO

Etapa de Profundización

Nataly Quintana Bernal

Ingeniería de Software

Mat. Patrones de Diseño

Fundación Universitaria Compensar

Introducción

Tomando como base los parámetros brindados para la ejecución de la etapa de profundización y dando solución a la actividad propuesta, en este informe se revisará el diagrama de clases y los requerimientos del caso para identificar los patrones de diseño de creación y estructurales adecuados, y se realizará la implementación de la aplicación en Java utilizando NetBeans, verificando su correcto funcionamiento mediante pruebas y generando la documentación de las mismas.

Caso de estudio:

"Los Atuendos" es un negocio de alquiler de vestidos para dama, caballero y disfraces. De cada prenda que el negocio tiene para alquiler se lleva un registro con esta información: referencia, color, marca, talla, valor del alquiler. Cuando se trata de un vestido para dama, se tiene además la indicación de si tiene pedrería, si es largo o corto y la cantidad de piezas que lo componen. Si es un traje para caballero se discrimina el tipo: convencional, frac, sacoleva, otro; además se indica si el traje lleva corbata, corbatín o plastrón. Por el lado de los disfraces, cada uno tiene un nombre. Cuando una persona acude al negocio para solicitar un servicio, se toma su número de identificación, nombre, dirección, teléfono y correo electrónico. Una persona puede alquilar la cantidad de trajes o vestidos que necesite. Cada servicio de alquiler queda registrado con número, fecha de solicitud, fecha de alquiler, empleado que tomó el pedido, cliente que toma el servicio. Nombre, número de identificación, dirección, teléfono y cargo registrado de cada empleado del negocio. Cuando se devuelven las prendas, se registran en un listado para enviar a lavado, según el orden en que van llegando al negocio; pero cuando la prenda llega manchada, es delicada o el administrador considera necesario, se le da prelación para enviarlo a lavado. Por restricciones logísticas, las prendas se envían a lavado en tandas, según la disponibilidad del día.

Análisis de Calidad del Caso de Estudio

- Requisitos Funcionales:

Registro de entidades	Clientes, Empleados, Prendas *(se debe validar que todos los campos necesarios estén presentes)*
Registro de servicios de alquiler	*Se debe impedir asociar un cliente, empleado o prenda inexistentes. Se debe validar la disponibilidad de la prenda*
Consultas	ID, Cliente, fecha, talla *(consistencia de datos mostrados)*

- Requisitos no funcionales:

  - Base de datos relacional: validar integridad, claves foráneas entre prendas, clientes, empleados y servicios.
  - Manejo de excepciones: pruebas para garantizar que el sistema no se caiga ante entradas inválidas.

1. Análisis de patrones de diseño

Patrones de creación: Teniendo en cuenta que estos ayudan a mejorar el manejo de instancias de objetos, se contempló:

- Factory Method: En la creación de entidades (*clientes, empleados, prendas*) se puede generar duplicación de códigos.

  Al utilizar este patrón, se puede centralizar la creación de objetos, lo que facilitaría al añadir nuevos tipos de entidades posteriormente

Patrones estructurales: Teniendo en cuenta que estos ayudar a mejorar la organización de las clases y sus relaciones, se contempló:

- Adapter: Si posteriormente el sistema debe integrar servicios externos, la interfaz actual puede no coincidir con la del sistema externo

  Al utilizar este patrón, permite crear una clase intermedia que sirva como traductora con una interfaz externa, lo que genera flexibilidad para integrar

- Composite: Las prendas pueden manejarse en conjuntos (varias piezas) o individualmente.

  Al utilizar este patrón, permite tratar objetos individuales y compuestos de la misma forma.

- Facade: Los procesos principales requieren interacción con varias clases. Esto complica el uso desde la capa de presentación.

  Al utilizar este patrón, permite que se unifique el acceso a las operaciones principales.

Patrón a utilizar	Tipo (Creación / Estructural)	¿Por qué contribuye a mejorar el diseño?
Factory Method	Creación	Centraliza la creación de objetos como Cliente, Empleado, Prenda, evitando duplicación de código y permitiendo extender fácilmente con nuevos tipos.
Adapter	Estructural	Facilita la integración con sistemas externos (ej. lavandería, facturación), adaptando interfaces sin modificar la lógica interna del sistema.
Composite	Estructural	Permite manejar de forma uniforme prendas individuales y conjuntos de prendas (ej. un traje con varias piezas), reduciendo complejidad en el manejo de objetos.
Facade	Estructural	Simplifica el acceso a las funcionalidades principales (registrar alquiler, consultar disponibilidad, enviar a lavandería), ocultando la complejidad interna y mejorando la usabilidad del sistema.

Tabla1.

2. Diagrama de Clases (con patrones de diseño)

Con respecto al diagrama original, se tuvieron en cuenta los siguientes aspectos:

- **Factory Method**
  Para la creación de objetos como Prenda con subtipos: *VestidoDama*, *TrajeCaballero*, *Disfraz* y centralizado en una clase *PrendaFactor*

- **Singleton**:
  Clase *ConexionBD* como instancia única para acceso a datos.

- **Adapter**: Clase *LavanderiaAdapter* que permite conectar el sistema con un servicio externo de lavandería.

- **Composite**: Relación *PrendaCompuesta* para modelar conjuntos de prendas

- Facade: Clase *AlquilerFacade* que unifica las operaciones principales: registrar cliente, consultar alquiler, registrar prenda en lavandería.

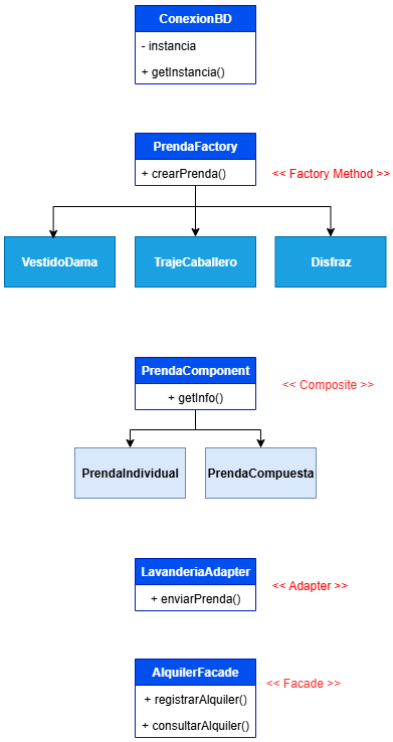
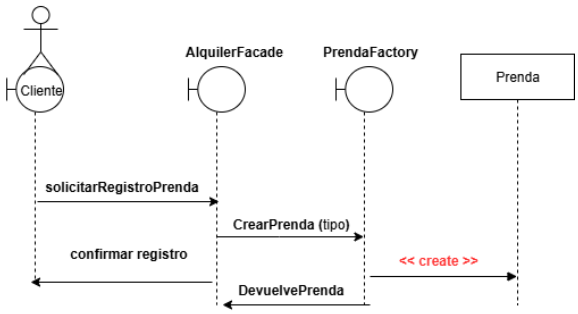


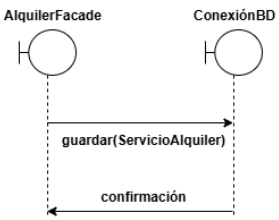
Figura 1. *Diagrama de clases*

3. Diagramas de Secuencia

ESC1: Registrar una prenda nueva / Factory Method



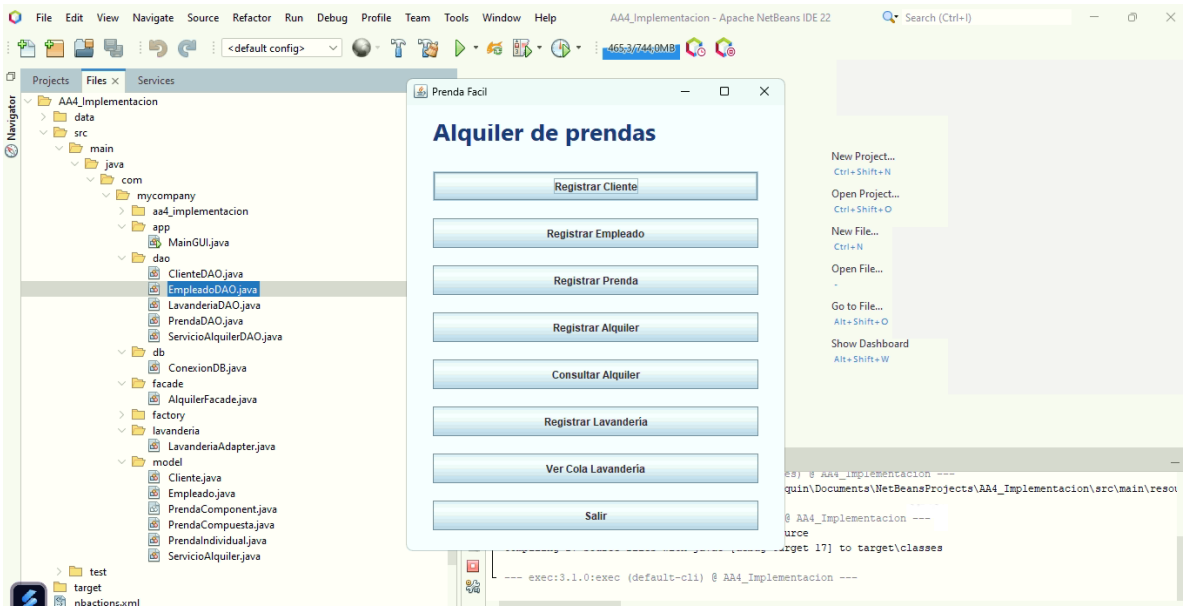
ESC2: Guardar una alquiler / Singleton – *Acceso a Base de Datos*



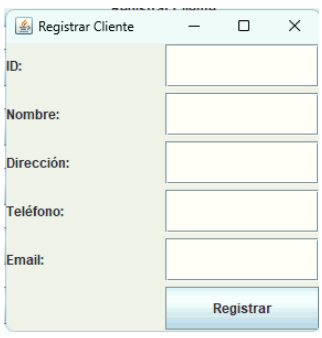
Segundo momento de actividad:

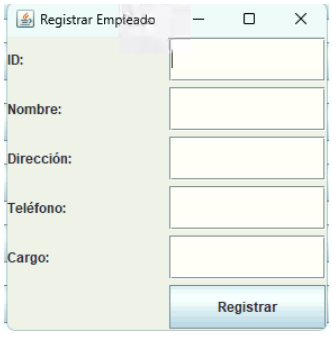
El desarrollo de la implementación del software fue subido a github y se puede acceder a la URL: NatalyQB/Patrones_Dise-o . Adicionalmente fue importado a la carpeta compartida en drive de Ucompensar Patrones de Diseño Software

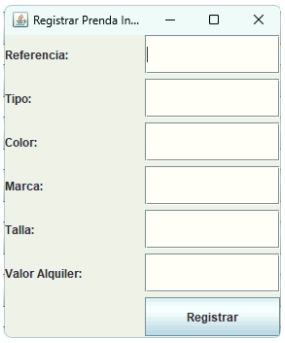
- En la implementación del proyecto se generó el menú principal



- Se confirma acceso a cada una de las opciones disponibles







- Se genero validación de cada una de las opciones confirmando que el flujo funciona de manera correcta, el proceso fue anexado como un gif a la carpeta del proyecto y adicionalmente puede ser visualizado sobre el link AA4_Implementacion_Quintana_Bernal_Nataly .gif