

Los Atuendos

## ACTIVIDAD 6

Patrones de diseño de software

Los Atuendos

Profundización

Cristian Andres Pulido Cristancho

Ucompensar

Ingeniería de software

Bogotá, Colombia

2025-2

## Patrones de creación y patrones estructurales.

### INDICE

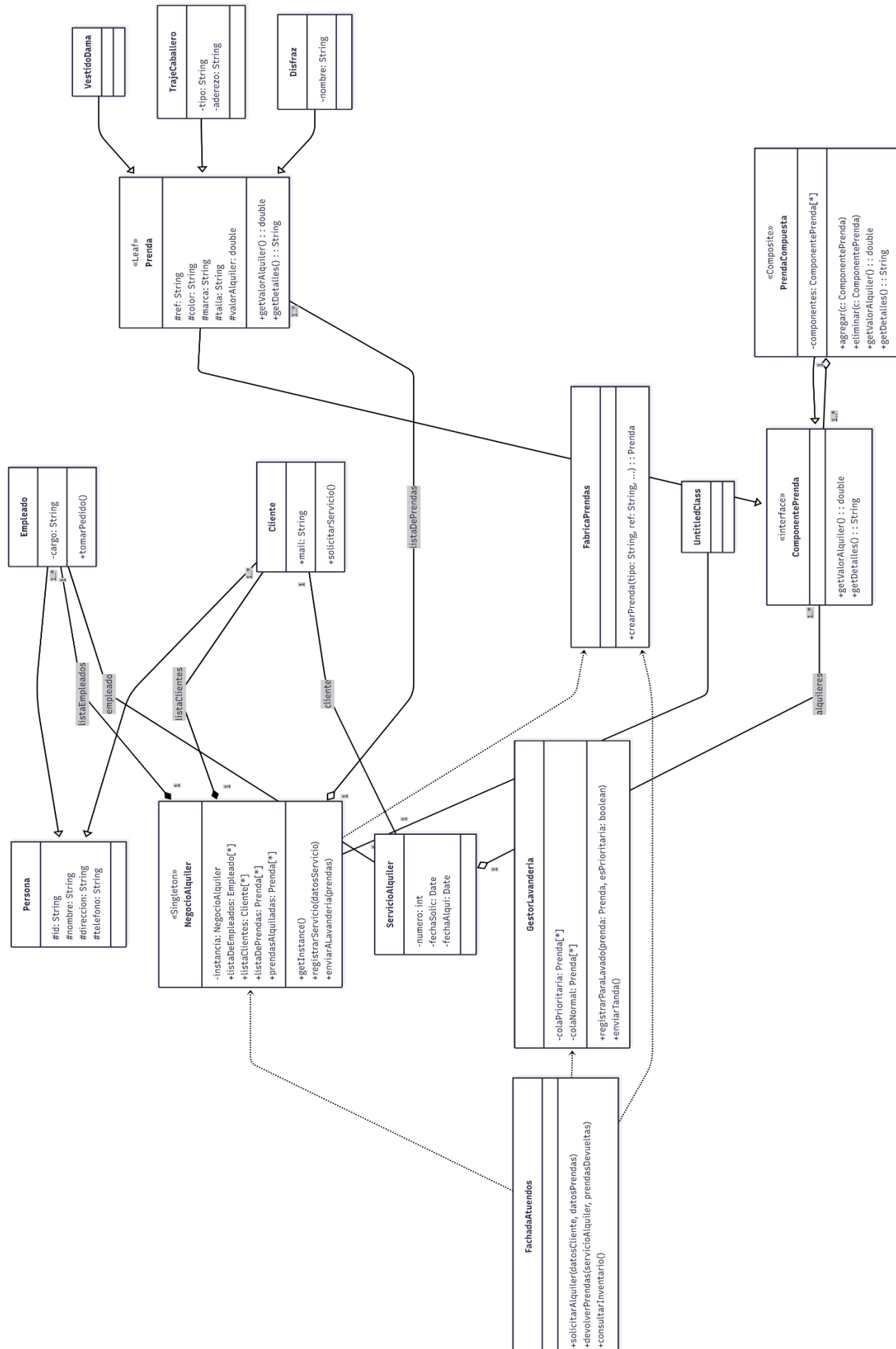
Patrones de creación y patrones estructurales. ....	2
Patrones de diseño identificados.....	3
Justificación general.....	3
Patrones de secuencia .....	5
Patrón Factory Method Creación de una Prenda .....	5
Patrón Singleton: Acceso a la Única Instancia del Negocio .....	6
Patrón Composite: Cálculo del Valor de Alquiler de un Traje Compuesto .....	7
Patrón Facade: Devolución de Prendas y Envío a Lavandería .....	8

**Patrones de diseño identificados**

<b>Patrón a Utilizar</b>	<b>¿Por qué contribuye a mejorar el diseño?</b>
<b>Factory Method</b>	Permite crear objetos de tipo Prenda (VestidoDama, TrajeCaballero, Disfraz) sin acoplar la lógica del negocio a las clases concretas. Facilita la extensión si en el futuro se agregan nuevos tipos de prendas sin modificar el código existente.
<b>Singleton</b>	Garantiza que exista una única instancia de NegocioAlquiler que gestione los registros de prendas, clientes y empleados. Evita inconsistencias de datos y facilita el control centralizado de la información.
<b>Composite</b>	Permite tratar de forma uniforme las prendas que están compuestas por varias piezas (como los vestidos de dama con varias partes). Mejora la flexibilidad del sistema al manejar colecciones jerárquicas de objetos.
<b>Facade</b>	Proporciona una interfaz unificada para acceder a las operaciones del sistema (registro, consultas, envío a lavandería). Reduce la complejidad del sistema para el usuario y mejora la mantenibilidad.

**Justificación general**

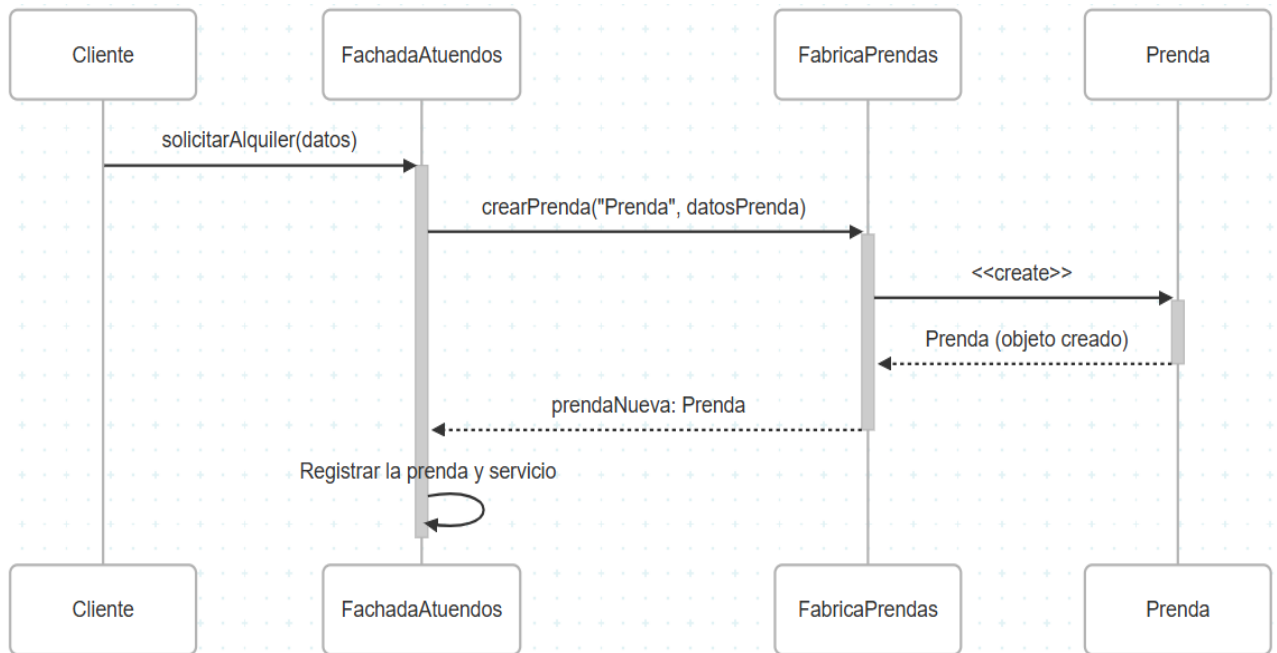
El sistema “Los Atuendos” está conformado por una variedad de entidades que se encuentran estrechamente vinculadas entre sí, entre las cuales se destacan los Clientes, los Empleados, las Prendas y los Servicios de Alquiler. Debido a la diversidad y complejidad de estos componentes, resulta fundamental aplicar patrones de diseño creacionales. Estos patrones permiten separar de manera adecuada la lógica encargada de crear o instanciar objetos de la lógica de negocio propiamente dicha. Gracias a esta separación, el sistema puede mantenerse con mayor facilidad y también escalar de forma más ordenada a medida que aumentan sus funcionalidades. De igual manera, la integración de patrones de diseño estructurales aporta una arquitectura más modular, organizada y flexible. Este tipo de patrones facilita que el sistema pueda ampliarse incorporando nuevas características, clases o módulos sin generar impactos negativos en las partes que ya se encuentran implementadas. Esta capacidad de adaptación es especialmente valiosa en un sistema orientado a la gestión de alquileres, donde los procesos pueden cambiar constantemente y los requerimientos evolucionan con el tiempo. Por ello, contar con una estructura robusta y adaptable garantiza que la plataforma pueda sostener su crecimiento y responder a nuevas necesidades sin comprometer su estabilidad.



## Patrones de secuencia

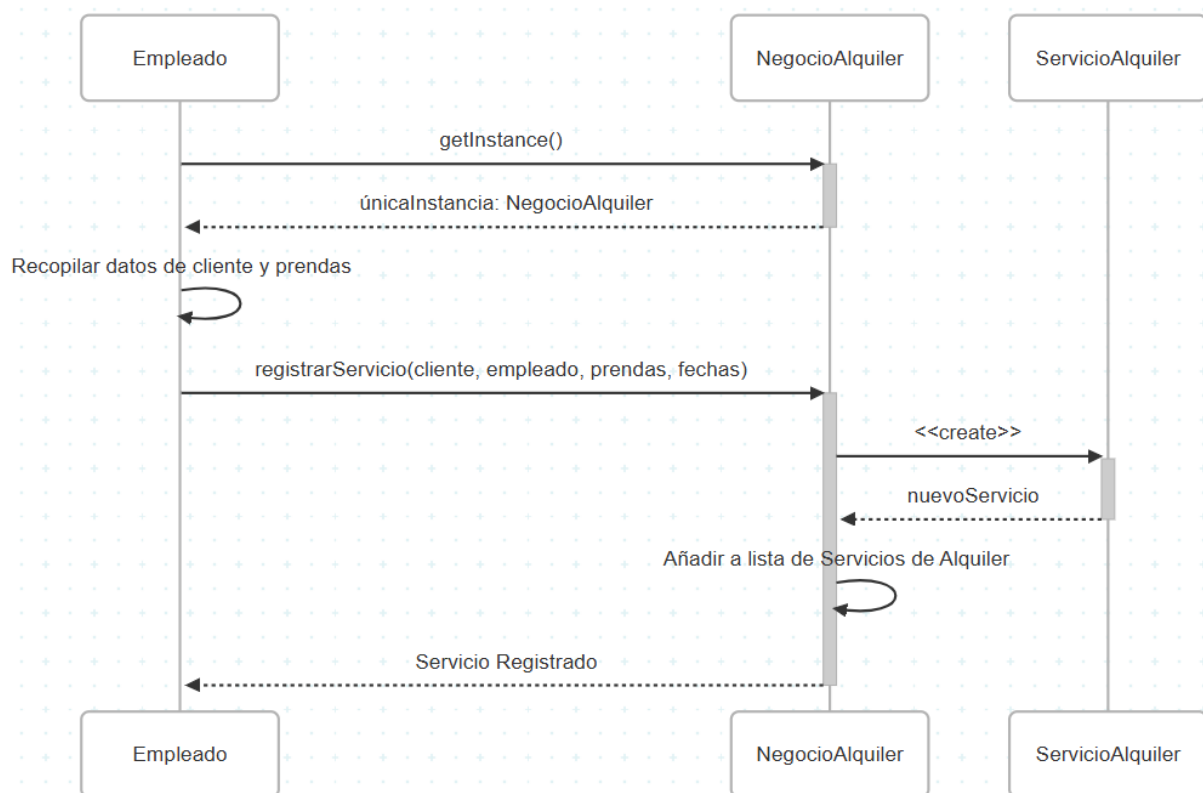
### Patrón Factory Method: Creación de una Prenda

Cuando el cliente solicita un nuevo tipo de prenda, la **FabricaPrendas** es la responsable de crear la instancia correspondiente a la clase concreta, en este caso **VestidoDama**. Gracias a este mecanismo, la lógica de negocio —ya sea en **NegocioAlquiler** o en la **FachadaAtuendos**— no necesita conocer ni manejar directamente los detalles internos de esa clase específica. Esto permite que el sistema se mantenga desacoplado y flexible ante la incorporación de nuevos tipos de prendas.



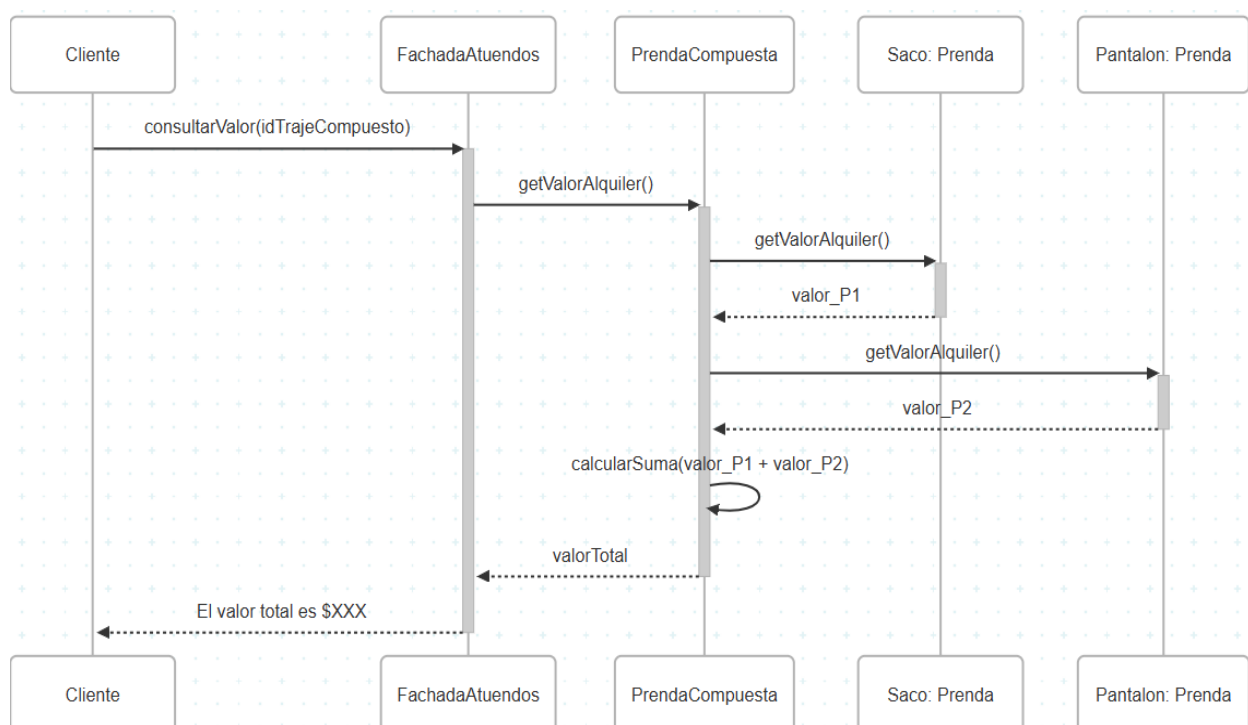
### Patrón Singleton: Acceso a la Única Instancia del Negocio

Este patrón se utiliza cuando es necesario acceder a la gestión centralizada del sistema; por ejemplo, al momento de registrar un servicio de alquiler. Su función es garantizar que todas las operaciones se realicen siempre sobre la **única instancia existente de NegocioAlquiler**, evitando duplicidades y asegurando que toda la información y los procesos se manejen de forma coherente y unificada dentro del sistema.



### Patrón Composite: Cálculo del Valor de Alquiler de un Traje Compuesto

Un traje de caballero puede representarse como una **PrendaCompuesta**, integrada por varias prendas individuales, como el saco, el pantalón y el aderezo. Cada una de estas partes se modela como un objeto **Prenda** independiente. Cuando el cliente desea conocer el valor total del alquiler de todo el conjunto, no es necesario calcular cada elemento por separado; en su lugar, la **PrendaCompuesta** se encarga de gestionar esta solicitud. El patrón *Composite* permite que la prenda compuesta delegue el cálculo del costo a cada una de sus prendas internas, sumando sus valores y proporcionando así el precio final del alquiler del traje completo. Esto facilita el manejo de conjuntos de prendas y mantiene el sistema organizado y extensible.



### Patrón Facade: Devolución de Prendas y Envío a Lavandería

El cliente realiza todo el proceso de devolución a través de la **FachadaAtuendos**, la cual simplifica y centraliza las operaciones necesarias. Aunque internamente la devolución implica varias acciones —como actualizar el estado del servicio de alquiler, registrar la prenda para su proceso de lavado y remitirla al **GestorLavanderia**—, la fachada se encarga de coordinar estas tareas de manera transparente. De este modo, el cliente interactúa con un único punto de acceso sin necesidad de conocer la complejidad del flujo interno.

