

Apar [KZ]

Memoria Final - Ingeniería del Software

Arroyo Aguilera, Carlos
Casado Domínguez, Rubén
Díaz Yagüe, Víctor
Fraila Pascual, José Luis
María Paredes, Iván
Merino Martínez, Jorge
Prieto Escolar, Juan José

Control de cambios

Número de versión	Fecha	Autores	Descripción
V. 1.0	16/05/16	Víctor Díaz Yagüe	Creación borrador memoria final
V. 1.1	16/05/16	Víctor Díaz Yagüe	Apartado 1, 2.2, 2.3
V. 1.2	26/05/16	Víctor Díaz Yagüe	Diagramas, patrones, gestión del proyecto, realizar formato e índice.
V. 1.3	27/05/16	Víctor Díaz Yagüe	Diagramas, casos de prueba, descripción sesiones.

Índice

1.	Introducción	5
2.	Análisis y Diseño	6
2.1	Breve descripción de las Sesiones implementadas	6
2.2	Diagramas de Casos de Uso	6
3.	Casos de Uso	7
3.1	Subsistema Taller:.....	7
3.1.1	Descripción del Caso de Uso: Registrar ingreso del vehículo	7
3.1.2	Descripción del caso de Uso: Modificar ficha de vehículo.....	8
3.1.3	Descripción del caso de Uso: Facturar el servicio prestado	9
3.2	Subsistema Lavadero:.....	10
3.2.1	Descripción del caso de Uso: Interrumpir el programa de lavado y pago.....	10
3.2.2	Descripción del caso de Uso: Finalizar el programa de lavado y pago	11
3.2.3	Descripción del caso de Uso: Gestionar tipo Lavado	11
3.2.4	Descripción del Caso de uso: Iniciar el programa de lavado	13
4.	Arquitectura de tres capas en tu diseño	14
4.1	Diagramas de Clases y de Secuencia	14
4.1.1	Diagrama de clases.....	14
4.1.2	Subsistema Taller:	14
4.1.3	Subsistema Lavadero:	16
4.2	Capa de Presentación.....	18
4.2.1	Clases Participantes.....	18
4.2.2	Patrones Arquitectónicos Implementados.....	19
4.2.2.1	Fachada	19
4.2.3	Patrones de Diseño Implementados	19
4.3	Capa de Negocio	19
4.3.1	Clases Participantes.....	19
4.3.2	Patrones Arquitectónicos Implementados.....	20
4.3.2.1	Servicio de Aplicación	20
4.3.2.2	Transfer	20
4.3.2.3	MVC.....	21
4.3.3	Patrones de Diseño Implementados.....	22

4.3.3.1	Singleton	22
4.4	Capa de Integración	23
4.4.1	Clases Participantes.....	23
4.4.2	Patrones Arquitectónicos Implementados.....	23
4.4.2.1	DAO	23
4.4.3	Patrones de Diseño Implementados.....	23
5.	Otros Patrones que se pueden aplicar al Proyecto	24
5.1	Factoría.....	24
5.2	Adapter.....	25
5.3	Controlador Frontal.....	26
5.4	Plantilla	26
6.	Casos de Prueba	28
6.1	Visión general	28
6.2	Subsistema Taller:.....	28
6.2.1	Caso de prueba: Registrar ingreso del vehículo.....	28
6.2.2	Casos de prueba: Modificar ficha de vehículo	30
6.2.1	Caso de prueba: Facturar el servicio prestado.....	31
6.3	Subsistema Lavadero:.....	35
6.3.1	Caso de Prueba: Iniciar el programa de lavado	35
6.3.2	Caso de Prueba: Interrumpir el programa de lavado y pago.....	37
6.3.3	Caso de prueba: Finalizar el programa de lavado y pago.....	38
6.3.4	Caso de prueba: Gestionar tipo Lavado	39
6.4	Mensajes de error: parsers	39
	Para los campos vacíos o erróneos saltará un aviso indicándolo con un ejemplo.	39
7.	Gestión del Proyecto	40

1. Introducción

A continuación, vamos a realizar una breve explicación de los componentes de la entrega final.

- 1.- Archivo con la lista de componentes actuales del grupo.
- 2.- El Parte de Actividades actualizado de todo el año, entrega del parte con la actividad por cada entrega del primer y segundo cuatrimestre, entregado de forma individual por miembro del equipo.
- 3.- Documentos finales de SRS y Plan del Proyecto, documentos corregidos del primer cuatrimestre con las indicaciones del profesor, con los diagramas de los distintos subsistemas.
- 4.- Proyecto RSA con todos los diagramas y la implementación en java.
- 5.- Ejecutable JAR con la funcionalidad principal del programa.

2. Análisis y Diseño

2.1 Breve descripción de las Sesiones implementadas

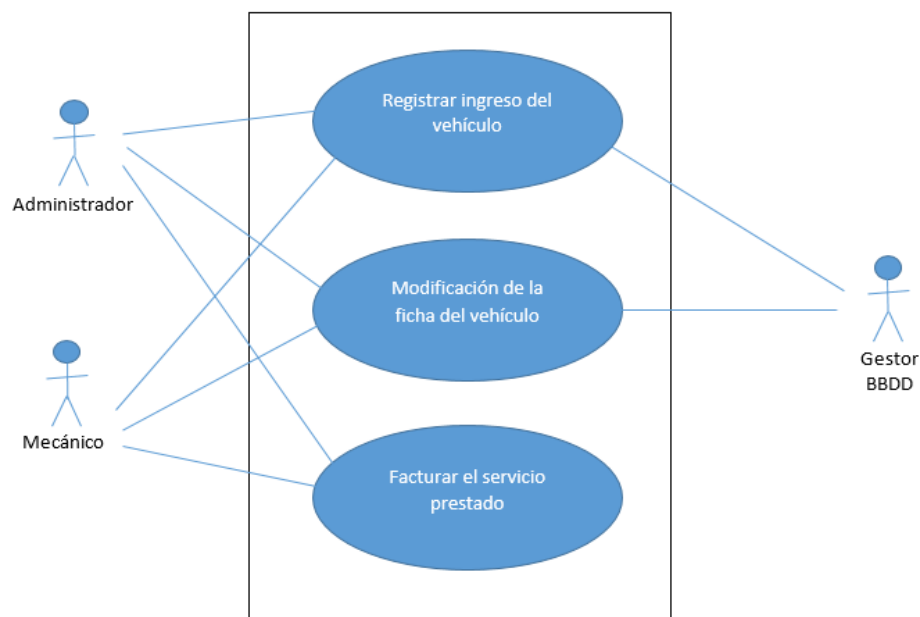
Hemos realizado la implementación de las sesiones del Taller y el Lavadero del aparcamiento.

En la sesión del taller el personal podrá registrar el ingreso de vehículos al taller automáticamente, también podrá guardar cualquier cambio en la ficha del vehículo y realizar una factura detallada de todos los servicios prestados.

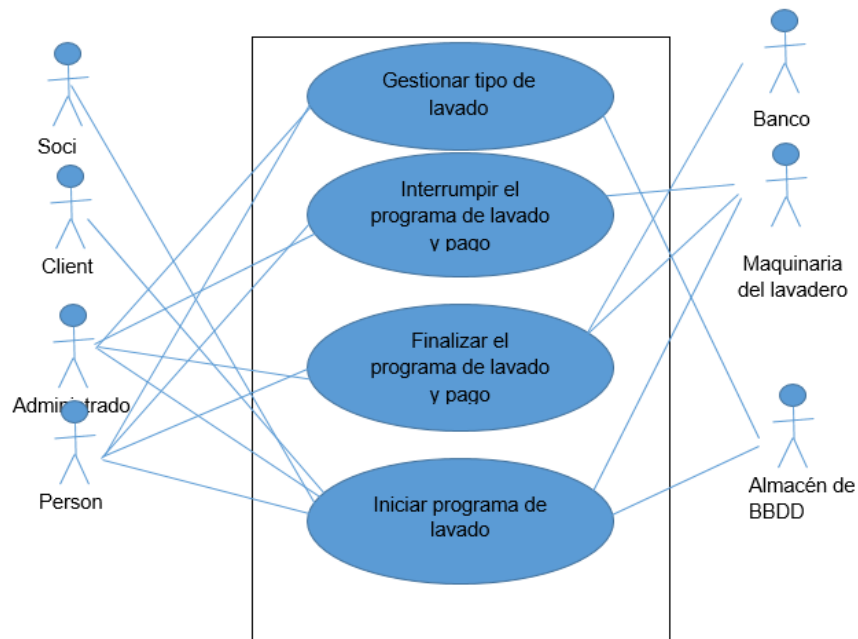
En la sesión del Lavadero el personal podrá gestionar el tipo de lavado deseado por el cliente, podrá iniciarlo una vez elegido el tipo de lavado, podrá interrumpirlo y finalizarlo en cualquier momento.

2.2 Diagramas de Casos de Uso

Subsistema Taller:



Subsistema Lavadero:



3. Casos de Uso

3.1 Subsistema Taller:

3.1.1 Descripción del Caso de Uso: Registrar ingreso del vehículo

Caso de uso: Registro de ingreso del vehículo en el taller.

Identificador: RegIngresoTaller

Objetivo en contexto: Se registra, mediante una máquina que lee la matrícula, el ingreso del vehículo en el taller. Si es la primera vez el cliente/socio debe ir al mostrador a crearse una ficha del taller.

Actor principal:

Actores secundarios: Gestor de BBDD, Máquina lectora de matrículas.

Qué datos usa: CRC's FichaVehiculo y IngresoVehiculo y pantallas ModFichP1 y DatosCliP1

Precondiciones: El cliente debe entrar en el taller con su vehículo

Postcondiciones:

Éxito: se registra el vehículo.

Fallo: la máquina no lo registra y lo hace el personal del taller.

Flujo principal

1. La máquina registra la matrícula y crea una ficha con la hora de entrada al taller.
2. El empleado del taller rellena la ficha con los datos del cliente/socio y los guarda en el sistema.

Flujos Secundarios

- 1.a Falla la máquina y el empleado registra la matrícula en el paso 2
- 2.a Fallo al guardar, vuelve al paso 1

3.1.2 Descripción del caso de Uso: Modificar ficha de vehículo

Caso de uso: Modificar ficha vehículo

Identificador: ModFicha

Objetivo en contexto: Añadir o eliminar información sobre el vehículo, como averías, reparaciones...

Actor principal: Personal del taller

Actores secundarios: gestor de BBDD.

Qué datos usa: CRC's FichaVehiculo y ModificacionFichaVehiculo y pantallas ModFichP1 y DatosCliP1

Precondiciones: El vehículo tiene que tener un registro de ingreso en el taller tal como indica el Caso de Uso: "Registro de ingreso del vehículo en el taller" del mismo subsistema. El personal tiene que haber iniciado sesión en el sistema.

Postcondiciones:

Éxito: La ficha del vehículo queda actualizada con nueva información.

Fallo: El vehículo no ha sido registrado en el taller y debe hacerlo.

Flujo principal

1. Se identifica el mecánico en el programa (pantalla 1).
2. Verificación de la existencia del mecánico.
3. Seleccionar opción de modificar la ficha.
4. Introducir matrícula del vehículo
5. Validación de los datos.
6. Añadir/eliminar datos del cliente, del vehículo y el tipo de avería
7. Guardar y salir.

Flujos Secundarios

2.a: Mensaje de error: "Personal no autorizado. Consulte al Administrador", volver al paso 1.

5.a: Matrícula no registrada en el taller, ir al Caso de Uso: "Registro de ingreso del vehículo en el taller"

7.a: Error al guardar, volver al paso 6.

3.1.3 Descripción del caso de Uso: Facturar el servicio prestado

Caso de uso: Facturar el servicio prestado

Identificador: FactrServicio

Objetivo en contexto: El personal del taller genera una factura de la reparación de las averías del vehículo y la mano de obra.

Actor principal: Personal del taller.

Actores secundarios: Gestor de BBDD.

Qué datos usa: CRC's FichaVehiculo y FacturaServicio y pantallas FacturaServP1

Precondiciones: El cliente debe haber ingresado el vehículo en el taller y solicitado una reparación.

Postcondiciones:

Éxito: se imprime la factura y se guarda en la base de datos. El cliente puede retirar su vehículo.

Fallo: se muestra mensaje de error y no se imprime la factura.

Flujo principal

1. El empleado introduce la matrícula y el DNI del cliente.
2. El sistema comprueba que la matrícula y el DNI están registrados en la base de datos.
3. El empleado introduce el tipo de reparación y el coste de la misma.
4. El sistema calcula el importe de la reparación y la mano de obra.
5. El sistema muestra en la pantalla los datos de la factura.
6. El empleado pulsa el botón de imprimir y se imprime la factura.

Flujos Secundarios

- 2.a La matrícula o el DNI no están registrados en la base de datos.
Mensaje de error y vuelve al paso 1.
- 6.a Falla la impresora y no se puede imprimir

3.2 Subsistema Lavadero:

3.2.1 Descripción del caso de Uso: Interrumpir el programa de lavado y pago

Identificador: InterrumpirLavado

Objetivo en contexto: Interrupción del programa de lavado que está realizando el usuario y por tanto el pago por este servicio no se realizará.

Actor principal: Personal

Actores secundarios: gestor de BBDD, Maquina Lavadero.

Qué datos usa: CRC's LavadoCoches e StopLavado y pantallas lavIniP1 y lavProcP1

Precondiciones: El lavado debe estar en proceso

Postcondiciones:

Éxito: Se procede a cancelar el lavado del vehículo y el pago de los servicios

Fallo: Mensaje de error y finaliza.

Flujo principal

1. (pantalla 1).
2. El personal selecciona cancelar el programa de lavado.
3. Verificación de la cancelación del lavado.
4. (pantalla 2).
5. Mensaje su servicio ha sido cancelado.
6. (pantalla 3)
7. Lavado y pago interrumpido.

Flujos Secundarios

- 3.a: Si no verifica la cancelación, seguirá con el proceso de lavado.

3.2.2 Descripción del caso de Uso: Finalizar el programa de lavado y pago

Caso de uso: Finalizar el programa de lavado y pago

Identificador: FinalizarLavado

Objetivo en contexto: Finalización del programa de lavado que se estaba realizando y por tanto se procede al pago por este servicio.

Actor principal: Personal

Actores secundarios: gestor de BBDD, Maquina Lavadero, banco.

Qué datos usa: CRC's LavadoCoches y FacturaLavado y pantallas lavIniP1 y lavPagoP1

Precondiciones: El lavado debe estar en finalizado con éxito.

Postcondiciones:

Éxito: Se procede a cobrar el lavado del vehículo.

Fallo: Mensaje de error y finaliza.

Flujo principal

1. (pantalla 1).
2. El Personal selecciona la forma de pago por la cual el cliente quiere pagar.
3. Si selecciona en caja lo pagara en efectivo.
4. (pantalla 2).
5. Confirmar al cliente que pase por caja.

Flujos Secundarios

- 3.a: Si selecciona tarjeta pasamos a 4.a
- 4.a: Entra en juego un usuario externo el datafono del cliente

3.2.3 Descripción del caso de Uso: Gestionar tipo Lavado

Caso de uso: Gestionar el tipo de lavado

Identificador: GestionarLavado

Objetivo en contexto: Gestión del tipo de lavado que quiere realizar el usuario pudiendo añadir más opciones.

Actor principal: Personal

Actores secundarios: gestor de BBDD, Maquina Lavadero, banco.

Qué datos usa: CRC's LavadoCoches y TipoLavado y pantallas lavIniP1 y lavTipoP1

Precondiciones: El tipo de lavado y las opciones deben existir en el sistema.

Postcondiciones:

Éxito: Se procede al lavado del vehículo.

Fallo: Mensaje de error y finaliza.

Flujo principal

1. (pantalla 1).
2. El personal selecciona el tipo de lavado
3. Verificación de la disponibilidad del lavado.
4. (pantalla 2).
5. Seleccionar opciones extras de lavado.
6. Verificación de la disponibilidad de los lavados extras.
7. (pantalla 3).
8. ¿Está seguro que desea proceder al lavado?
9. (pantalla 4).
- 10.: Lavado procesado con éxito.

Flujos Secundarios

- 3.a: Mensaje de error: "El lavado no está disponible, Consulte al Administrador para saber el stock disponible", volver al paso 2.
- 6.a: Mensaje de error: "El lavado extra no está disponible, Consulte stock de material para saber existencias disponibles", volver al paso 5.
- 8.a: volver al paso 2.
- 4.- Descripción del caso de Uso: Iniciar el programa de lavado

3.2.4 Descripción del Caso de uso: Iniciar el programa de lavado

Identificador: IniciarLavado

Objetivo en contexto: Inicia el programa de lavado por tanto se procede a prestar este servicio.

Actor principal: Personal

Actores secundarios: gestor de BBDD, Maquina Lavadero.

Qué datos usa: CRC's LavadoCoches y pantallas lavIniP1

Precondiciones: El túnel de lavado debe estar vacío para poder iniciar el lavado con éxito

Postcondiciones:

Éxito: Se procede al lavado del vehículo.

Fallo: Mensaje de error y finaliza.

Flujo principal

1. (pantalla 1).
2. El Personal siguiendo las instrucciones del cliente llama al caso de uso "Gestionar Lavado" para seleccionar el tipo de lavado.
3. El personal recibe los datos del caso de uso "Gestionar Lavado" e inicia el proceso de lavado.
4. Aceptar las condiciones de lavado de la acción 2.
5. (pantalla 2).
6. Nuestro software confirmara al personal que el cliente puede pasar al túnel de lavado.

Flujos Secundarios

- 3.a: Mensaje de error: "Debe aceptar las condiciones de uso", volver al paso 3.

Subsistema Cliente

4. Arquitectura de tres capas en tu diseño

4.1 Diagramas de Clases y de Secuencia

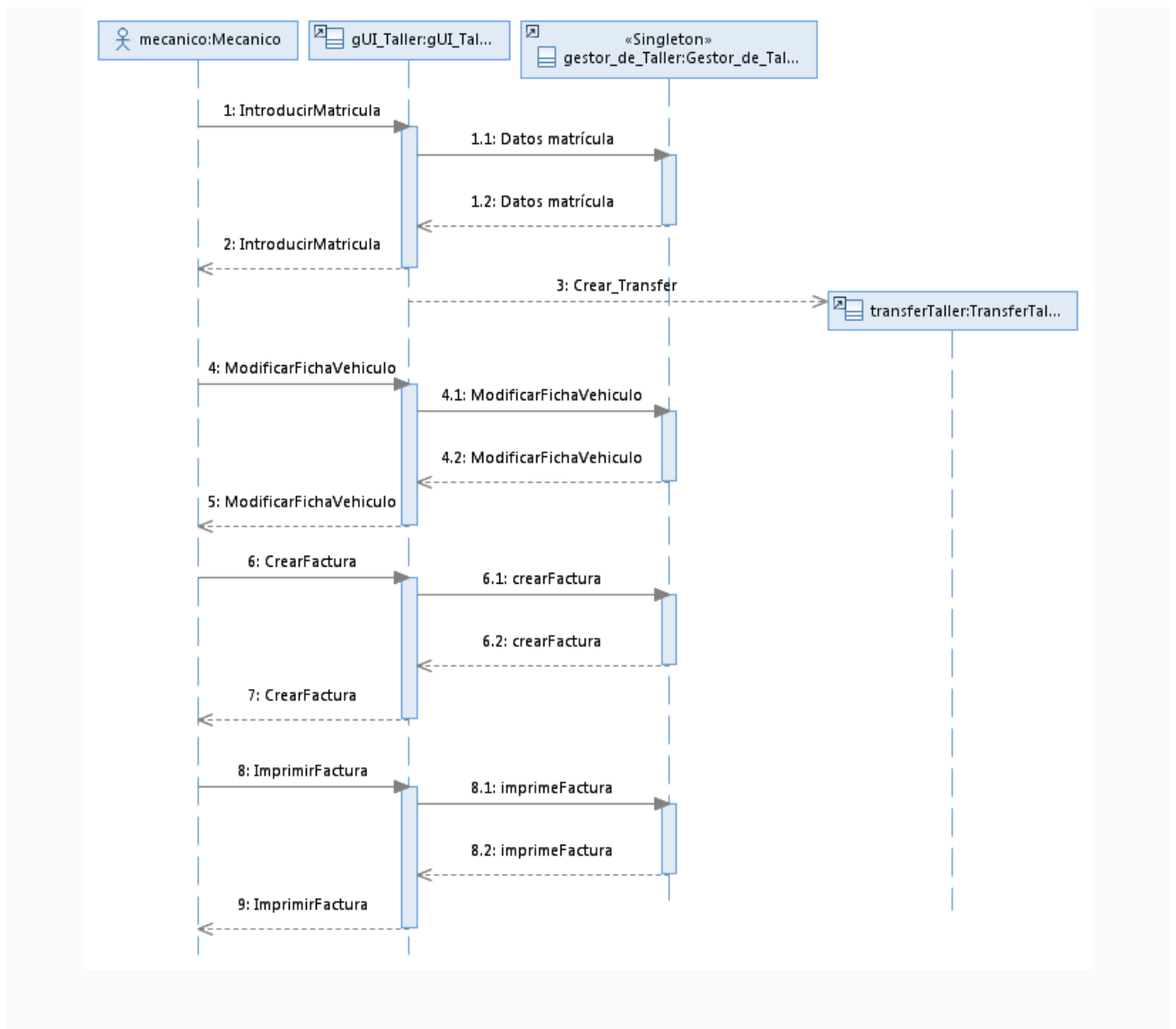
4.1.1 Diagrama de clases

Debido a que el diagrama de clases tiene un tamaño desmesurado, hemos generado un archivo pdf llamado ["Diagrama de clases de lo implementado.pdf"](#) para poder verlo completo de una forma cómoda.

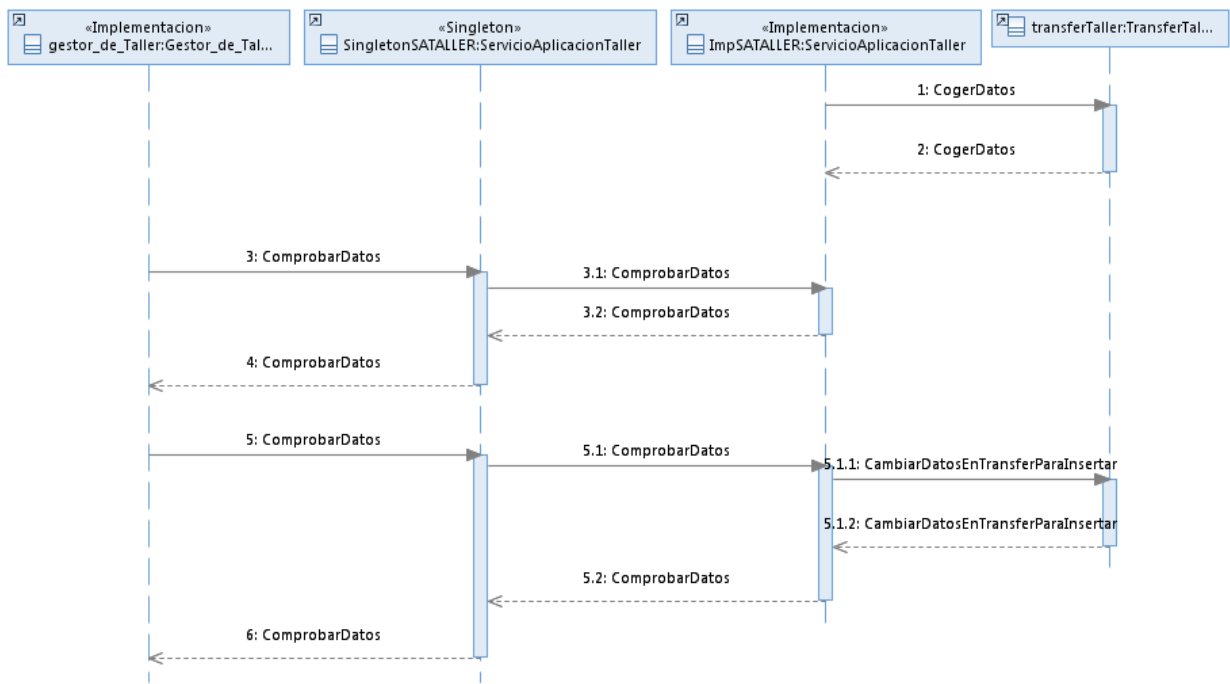
4.1.2 Subsistema Taller:

El diagrama de Secuencia del subsistema es el siguiente:

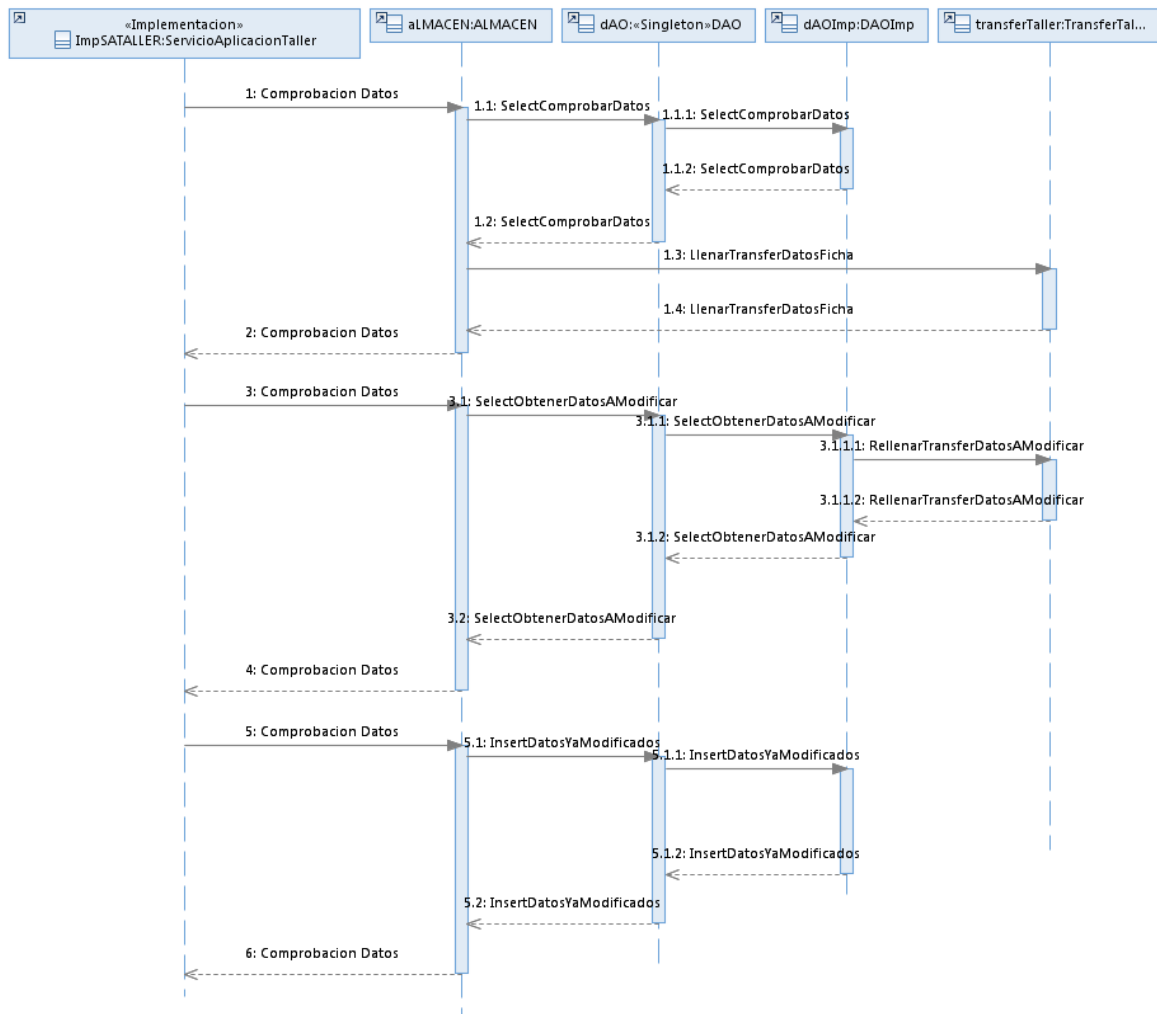
Capa de presentación



Capa de negocio



Capa de integración

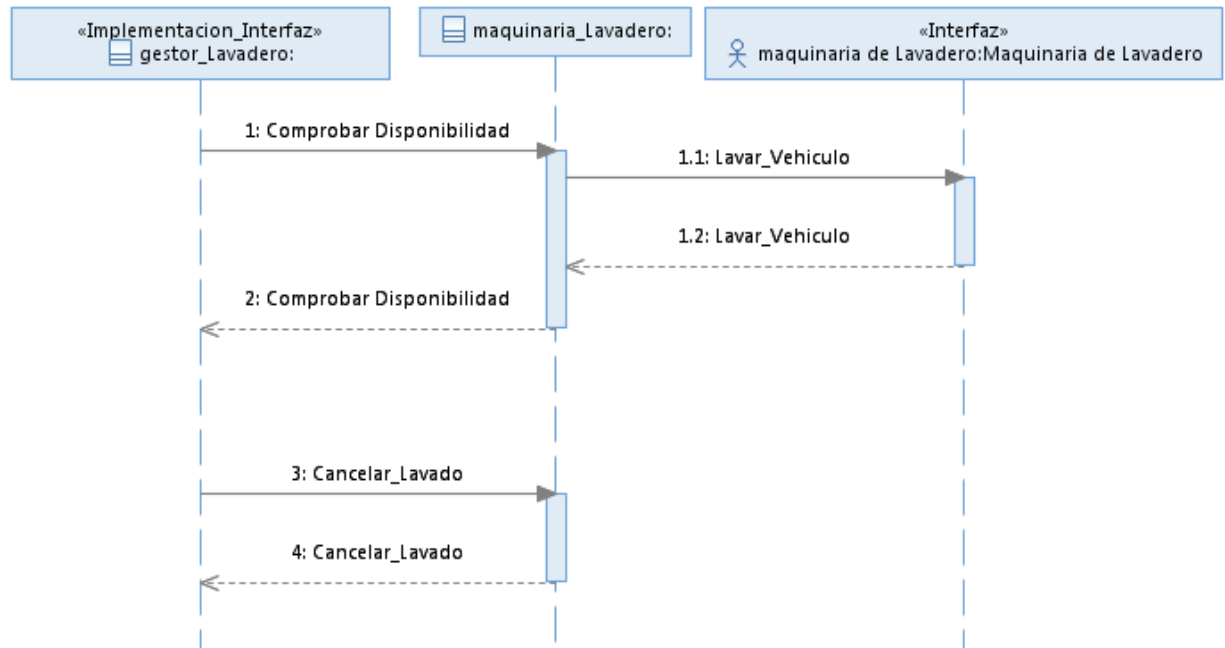


4.1.3 Subsistema Lavadero:

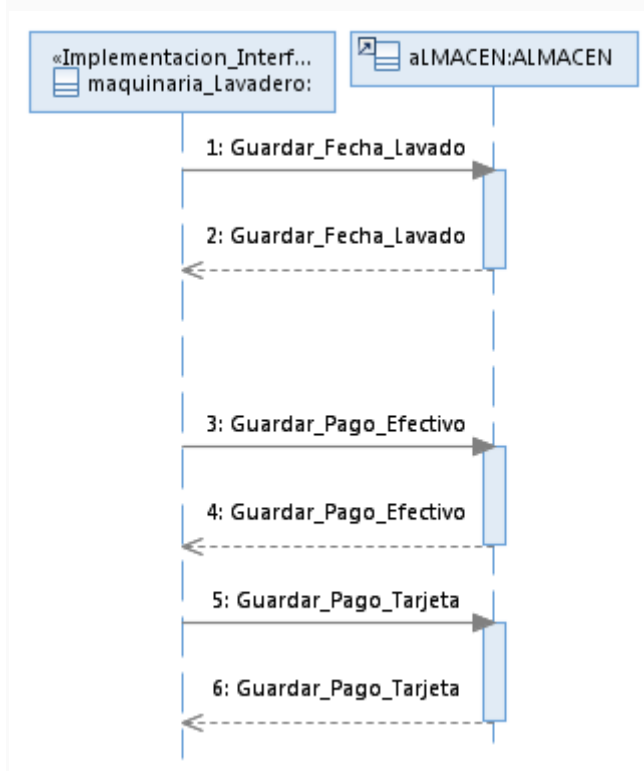
Capa de presentación:



Capa de Negocio



Capa de integración



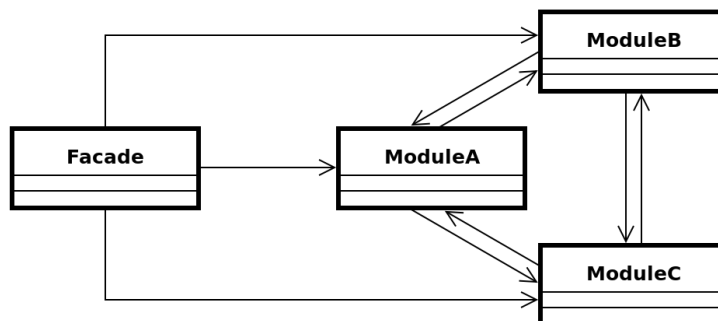
4.2 Capa de Presentación

4.2.1 Clases Participantes

Las clases participantes en la capa de presentación que utiliza el patrón arquitectónico fachada referente a los dos subsistemas implementados son GUI_Lavadero y GUI_Taller, y sus clases contenedoras.

4.2.2 Patrones Arquitectónicos Implementados

4.2.2.1 Fachada



Sirve para estructurar un entorno de programación y reducir su complejidad con la división en subsistemas, minimizando comunicaciones y dependencias entre ellos.

Participantes principales:

- Fachada: conoce qué clases son responsables de cada petición que realice el cliente.
- Subclases: implementan la funcionalidad del sistema realizando las operaciones correspondientes. Son independientes del cliente e incluso de la interfaz.

¿Por qué lo hemos elegido?

Para proporcionar una interfaz simple para un subsistema complejo, desacoplando así al cliente y haciéndolo más independiente, portable y reutilizable.

Porque para realizar una modificación del subsistema sólo habría que modificar la fachada y el cliente permanecería ajeno.

4.2.3 Patrones de Diseño Implementados

No hemos llegado a implementar ningún patrón de diseño en la capa de presentación.

4.3 Capa de Negocio

4.3.1 Clases Participantes

Las clases participantes en la capa de Negocio que utiliza el patrón arquitectónico Servicio de Aplicaciones referente a los dos subsistemas implementados son la clase creada para el propio patrón, *ServicioAplicacion* y *controlador imp*. Estas dos son usadas por el transfer, dicho patrón tiene

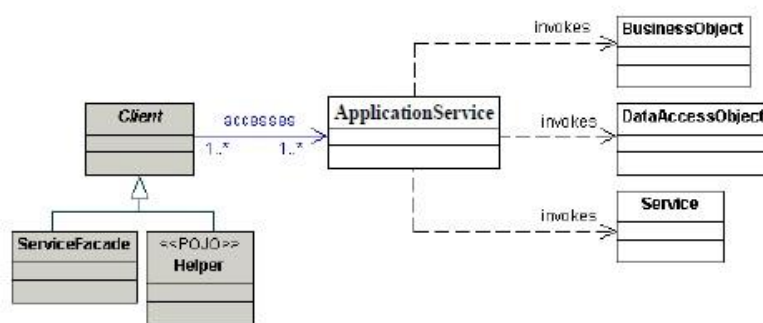
una clase llamada *Transfer* <<interfaz>> de donde se implementas las clases de dominio *Transfer* *Talle* y *Transfer Lavadero*.

El patrón MVC está presente en todo el proyecto, las clases participantes de dicho patrón son el conjunto de clases que componen nuestro proyecto.

Con respecto a los patrones de diseño que hemos implementado en el proyecto se encuentra el singleton, está presente en varias clases del proyecto en la capa de Negocio la propia clase del servicio de aplicación es un singleton junto con el controlador.

4.3.2 Patrones Arquitectónicos Implementados

4.3.2.1 Servicio de Aplicación



Centraliza la lógica del negocio.

Se debe utilizar cuando se quiera representar una lógica del negocio que actúe sobre distintos servicios u objetos del negocio, se quiera agrupar funcionalidades relacionadas o se quiera encapsular la lógica que no está representada por objetos del negocio.

¿Por qué lo hemos elegido?

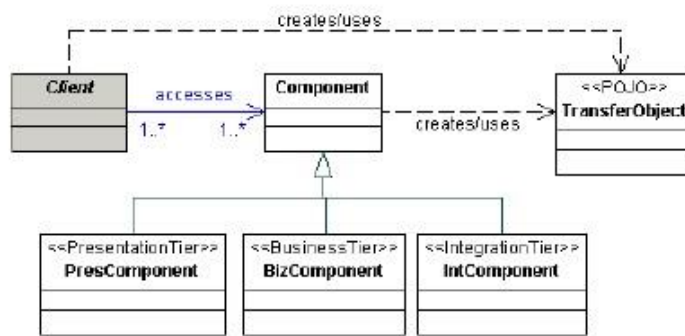
Porque centraliza la lógica del negocio.

Mejora la reusabilidad del código.

Evita duplicación de código.

Simplifica la implementación de fachadas.

4.3.2.2 Transfer



Tiene como objetivo independizar el intercambio de datos entre capas.

Se debe aplicar en el caso de que no se desee conocer la representación interna de una entidad dentro de una capa, por lo que usaremos objetos serializables.

En la capa de negocio es donde residen las funcionalidades de la aplicación, se reutilizan las peticiones de usuario, se procesará la información y se enviará una respuesta.

Se comunica con la capa de presentación para recibir las solicitudes y representar los resultados, también se comunica con la capa de integración para solicitar al gestor de base de datos almacenar o recuperar datos de él.

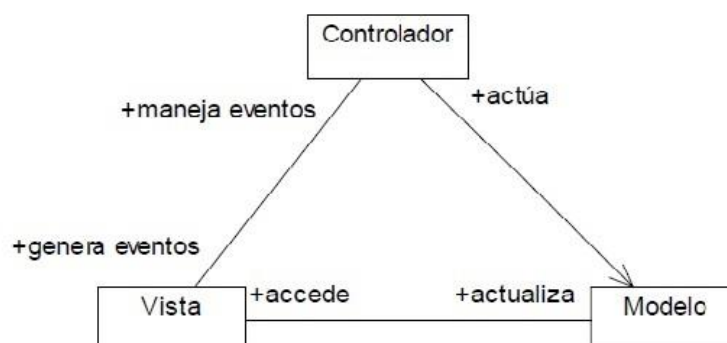
¿Por qué lo hemos elegido?

Por la necesidad de independizar capas.

Para reducir el número de llamadas remotas.

Porque mejora el rendimiento.

4.3.2.3 MVC



Es un patrón que consta de 3 partes:

1. Modelo: el cual contiene toda la parte funcional de la aplicación.
2. Vista: es una representación visual del modelo para que el usuario pueda interactuar con éste.
3. Controlador: determina que modificaciones debe hacer en el modelo cuando se interactúa con la vista. Puede solicitar tanto modificaciones de la vista como del modelo.

¿Por qué lo hemos elegido?

Porque tiene una buena separación entre la interfaz y la lógica del negocio

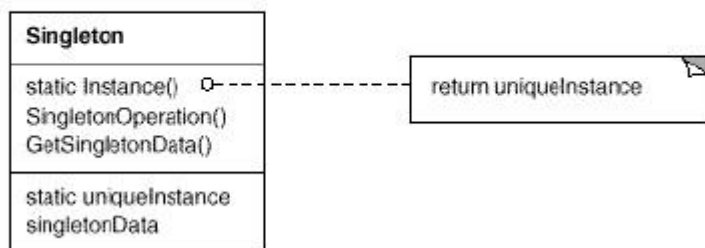
Se pueden implementar varias vistas para un mismo modelo.

Se pueden reutilizar componentes.

Tiene una eficiente y fácil mantenimiento.

4.3.3 Patrones de Diseño Implementados

4.3.3.1 Singleton



Este patrón garantiza que solo habrá una instancia de clase, y ésta debe ser accesible a los clientes desde un punto de acceso conocido. Dicha instancia debe ser redefinida mediante herencia.

¿Por qué lo hemos elegido?

Por su acceso controlado a la única instancia.

Tiene un espacio de nombres reducido.

Permite el refinamiento de operaciones y la representación.

Permite un número variable de instancias.

Más flexibles que las operaciones de clase estáticas.

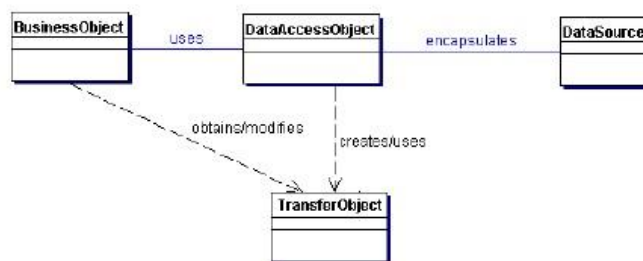
4.4 Capa de Integración

4.4.1 Clases Participantes

En la capa de integración solo hemos implementado el patrón DAO, clases participantes, DAO que a su vez es un Singleton, que está conectada con el servicio de aplicaciones. De la clase DAO se implementa la clase DAOImp que se comunica con la clase almacén y con los transfers.

4.4.2 Patrones Arquitectónicos Implementados

4.4.2.1 DAO



Permite acceder a la capa de datos, proporcionando representaciones orientadas a objetos a sus clientes.

Se usa para abstraer a la capa de negocio de las cuestiones relacionada con el acceso a los datos (accesos y consultas a bases de datos, conocimiento de la estructura de la bbdd...).

¿Por qué lo hemos elegido?

Para independizar la representación y acceso a los datos de su procesamiento.

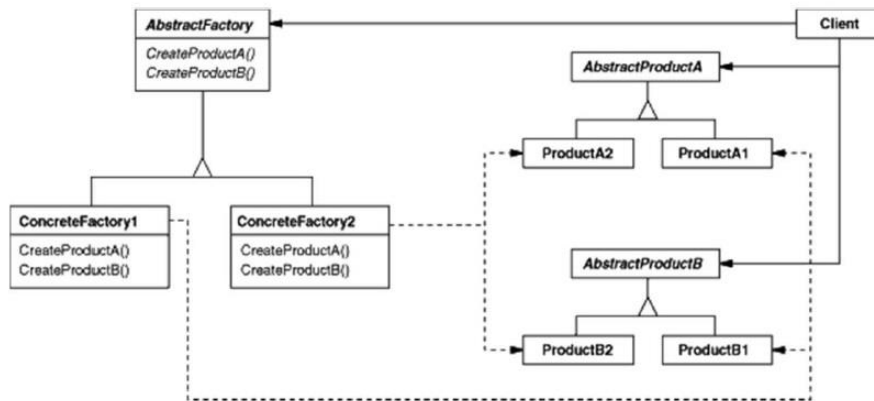
Independizar el tratamiento de los datos de su acceso y estructura.

4.4.3 Patrones de Diseño Implementados

No hemos llegado a implementar ningún patrón de diseño en la capa de integración.

5. Otros Patrones que se pueden aplicar al Proyecto

5.1 Factoría



Se aplica cuando:

Un sistema debe ser independiente de cómo se crean, componen y representan sus productos.

Un sistema debe ser configurado con una familia de productos de entre varias.

Una familia de objetos producto relacionados está diseñada para ser usada conjuntamente, y es necesario hacer cumplir esta restricción.

Quiere proporcionar una biblioteca de clases de productos, y sólo quiere revelar sus interfaces, no sus implementaciones.

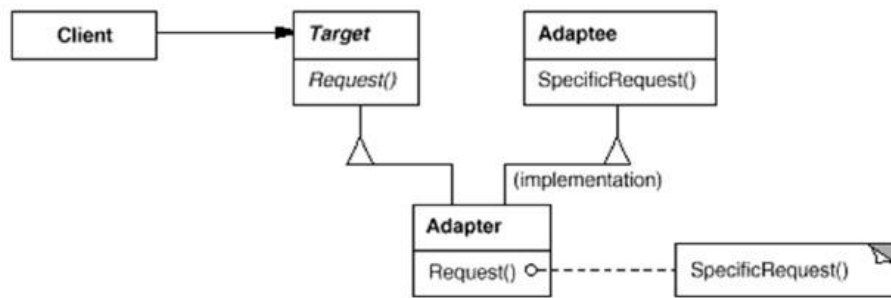
Lo podríamos haber utilizado porque:

Aísla las clases concretas de sus clientes.

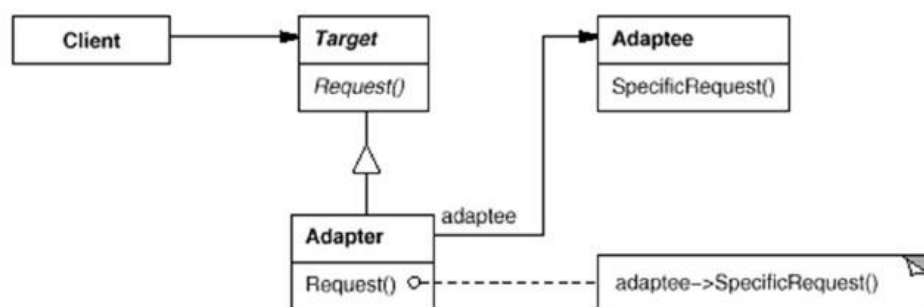
Facilita el intercambio de familias de productos.

Promueve la consistencia entre productos.

5.2 Adapter



Estructura de un Adaptador de Clases



Estructura de un Adaptador de objetos

Debe aplicarse cuando:

Se quiere usar una clase existente y su interfaz no concuerda con la que se necesita.

Se quiere crear una clase reutilizable que coopere con las clases no relacionadas o que no han sido previstas, es decir, clases que no tiene por qué tener interfaces compatibles.

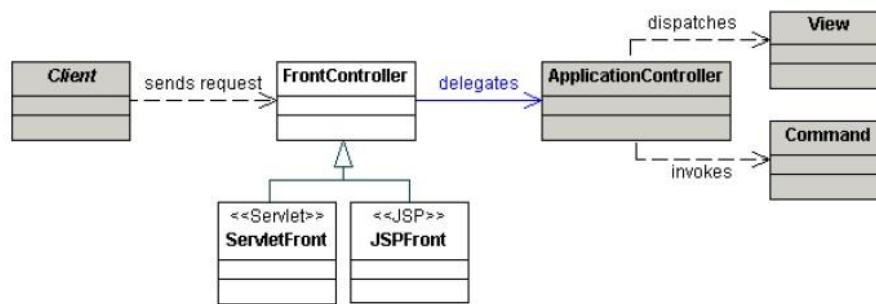
En el caso de un adaptador de objetos, es necesario usar varias subclases existentes, pero no resulta práctico adaptar su interfaz heredando de cada una de ellas. Un adaptador de objetos puede adaptar la interfaz de su clase padre.

Podríamos haber usado el adaptador clases porque:

Permite que el adaptador redefina parte del comportamiento del adaptable, al ser subclase suya.

Introduce un solo objeto, y no se necesita ningún puntero de dirección adicional para obtener el objeto adaptado.

5.3 Controlador Frontal



Proporciona un punto de acceso para el manejo de las peticiones de la capa de presentación.

Lo podríamos haber utilizado para:

Evitar lógica de control duplicada.

Aplicar una lógica común a distintas peticiones.

Separar la lógica de procesamiento del sistema de la vista.

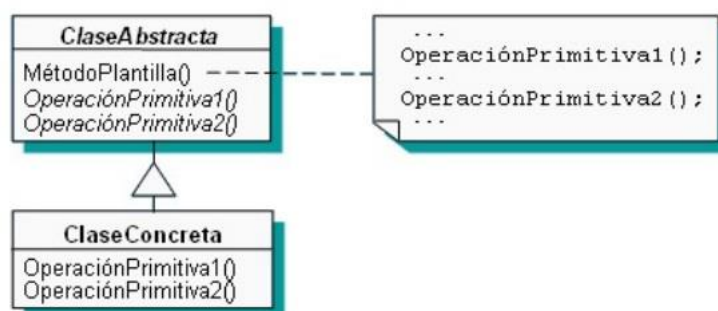
Tener puntos de acceso centralizado y controlado al sistema.

Tener un punto inicial de contacto para manejar las peticiones, centralizando la lógica de control y manejando las actividades de manejo de peticiones.

Mejorar la separación de roles.

Mejorar la gestión de la aplicación.

5.4 Plantilla



Define el esqueleto de programa de un algoritmo en un método, llamado método de plantilla, el cual difiere algunos pasos a las subclases. Permite redefinir ciertos pasos seguros de un algoritmo sin cambiar la estructura del algoritmo.

Lo podríamos haber utilizado para:

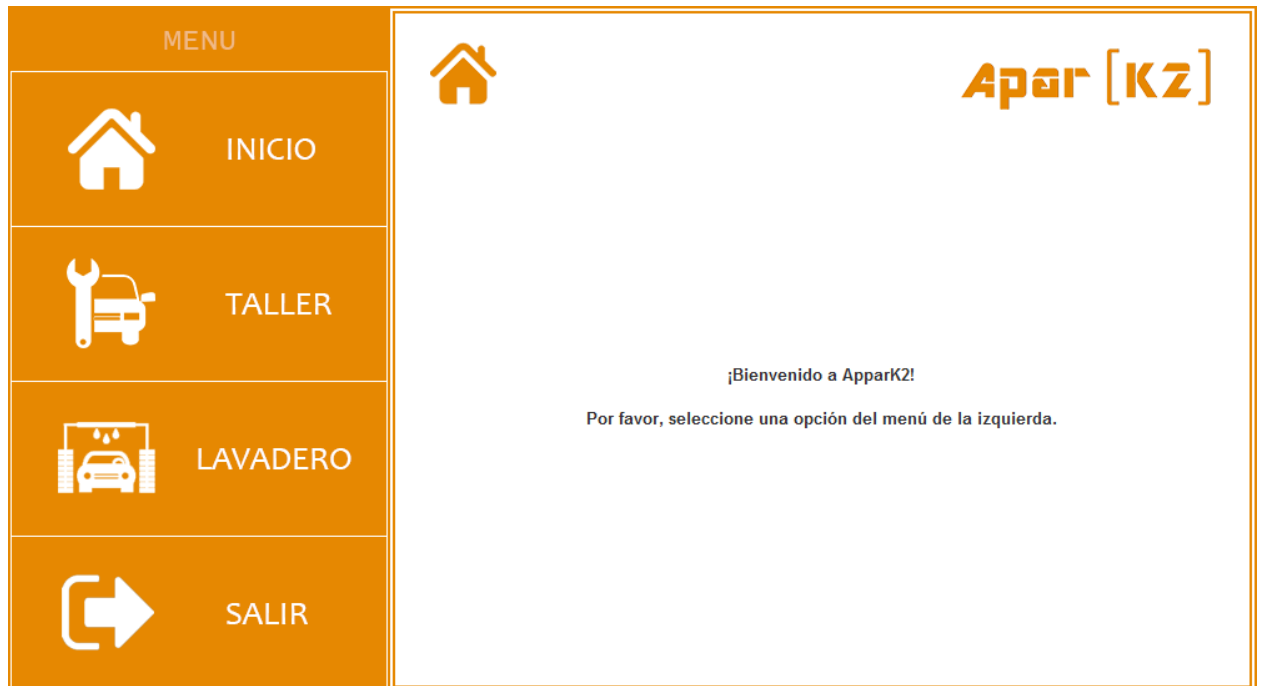
Implementar las partes fijas de un algoritmo una sola vez y dejar que las subclases implementen las partes variables.

Sacar factor común la conducta similar entre varias subclases: la parte común en un método plantilla, la parte distinta en operaciones abstractas que deben redefinirse.

Controlar las ampliaciones que puedan hacer las subclases: sólo pueden cambiar las operaciones abstractas.

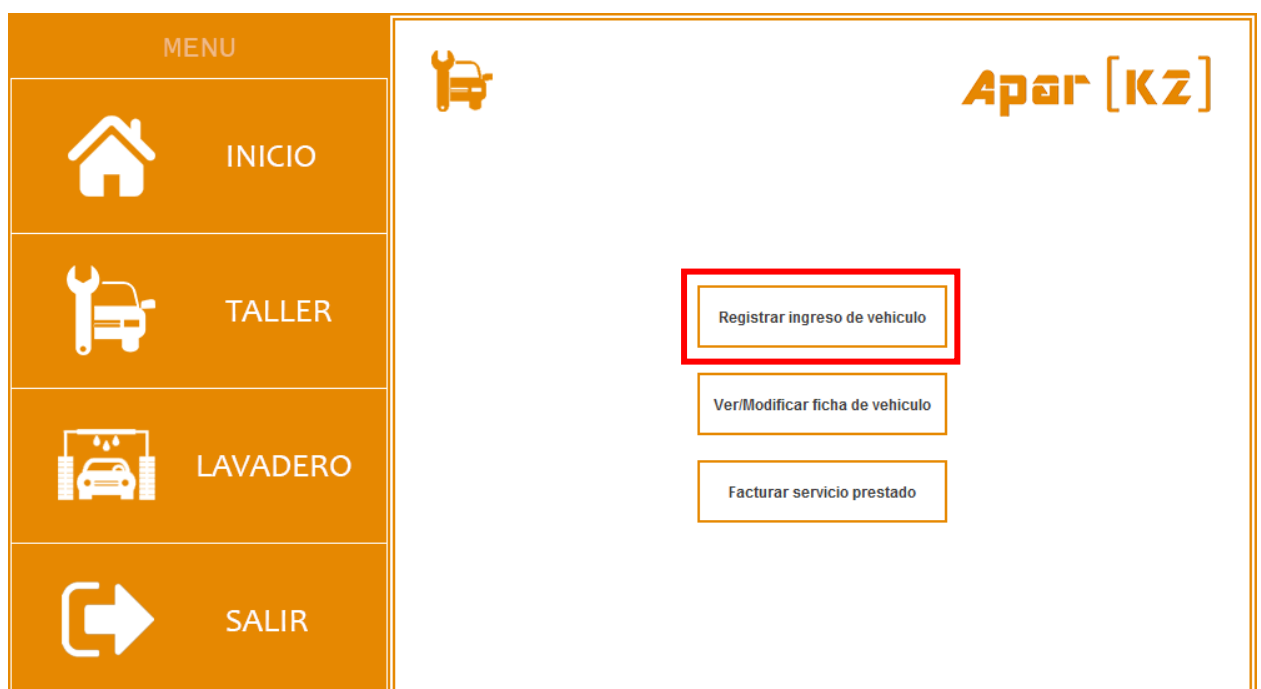
6. Casos de Prueba

6.1 Visión general




6.2 Subsistema Taller:


6.2.1 Caso de prueba: Registrar ingreso del vehículo



MENU

INICIO

TALLER

LAVADERO

SALIR



Apar [KZ]

Nombre:

Apellidos:

DNI:


Matricula:

Telefono:

Observaciones:

Guardar


Cancelar



Confirmación

Datos guardados con éxito.

Aceptar

 ¡Error!

Ya existe un cliente con esa matrícula

Aceptar





El vehículo se registra en el taller guardando sus datos personales, si se eligiera la opción de cancelar volveríamos a la pantalla de inicio.


6.2.2 Casos de prueba: Modificar ficha de vehículo

MENU	
	INICIO
	TALLER
	LAVADERO
	SALIR


Apar [KZ]


Buscamos una matrícula válida:

MENU	
	INICIO
	TALLER
	LAVADERO
	SALIR


Apar [KZ]


Matricula:


Si buscamos una matrícula que no es válida:



¡Error!


¡Matricula no encontrada!


MENU

 INICIO

 TALLER

 LAVADERO

 SALIR



Apar [KZ]

Nombre:

Apellidos:

DNI:

Matricula:

Telefono:


Observaciones:


Facturas:


Modificamos el campo deseado y le damos a guardar. Si quisiéramos cancelar, volvería la pantalla inicial.


6.2.1 Caso de prueba: Facturar el servicio prestado


MENU

 INICIO

 TALLER

 LAVADERO

 SALIR



Apar [KZ]

Buscamos una matrícula válida:

MENU

INICIO

TALLER

LAVADERO

SALIR

Apar [KZ]

Matricula:

1234ASD

Buscar

Nos aparecerá la pantalla de la factura, en la que podemos añadir las piezas que hemos utilizado.

MENU

INICIO

TALLER

LAVADERO

SALIR

Apar [KZ]

Nombre:

Piezas:

Añadir pieza

Importe:

Mano de obra:

Porcentaje de IVA (%):

Total de IVA:

TOTAL:

Guardar

Cancelar

Introducir nueva pieza:

Nombre:

Precio:

Añadir

Cancelar

MENU

INICIO

TALLER

LAVADERO

SALIR

Nombre:

Factura 1

Piezas:

Carburador: 89€

Añadir pieza

Importe:

89.0

Mano de obra:

45

Porcentaje de IVA (%):

21

Total de IVA:

28.14

TOTAL:

162.14

Guardar

Cancelar

Confirmación

Factura guardada con éxito.

Aceptar

Guardamos la factura para ese cliente, si queremos ver las facturas de un cliente, en el apartado de ver/modificar ficha de vehículo podremos pinchar en el botón ver facturas.

MENU

INICIO

TALLER

LAVADERO

SALIR

Registrar ingreso de vehiculo

Ver/Modificar ficha de vehiculo

Facturar servicio prestado





Página 33


MENU	
	INICIO
	TALLER
	LAVADERO
	SALIR



Apar [KZ]

Matricula:

MENU	
	INICIO
	TALLER
	LAVADERO
	SALIR



Apar [KZ]

Nombre:

Apellidos:

DNI:

Matricula:

Telefono:

Observaciones:

Facturas:

MENU

INICIO

TALLER

LAVADERO

SALIR



Apar [KZ]

Nombre:

Factura 1

Importe:

89.0 €

Mano de obra:

45.0 €

Porcentaje de IVA (%):

21 %

Total de IVA:

28,14 €

TOTAL:

162,14 €

Piezas:

Carburador: 89.0 €

Anterior

Siguiente

Volver

6.3 Subsistema Lavadero:

6.3.1 Caso de Prueba: Iniciar el programa de lavado

Elegimos la opciones que tiene un lavado e iniciamos.

MENU

 INICIO

 TALLER

 LAVADERO

 SALIR



Apar [KZ]

Tipo de agua

Agua caliente

Tipo de lavado

Lavado completo


Cristales

Ecológico

Llantas

Químico

Iniciar lavado

 Confirmación

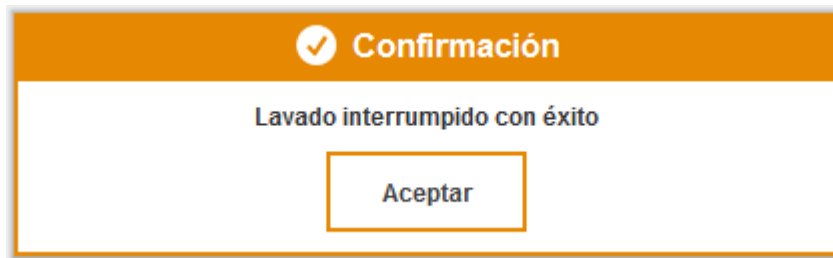
Lavado iniciado.

Aceptar

6.3.2 Caso de Prueba: Interrumpir el programa de lavado y pago

Mientras que estamos en la pantalla de lavado en curso, que llegamos a ella eligiendo interrumpir lavado en la pestaña de Lavadero, podemos elegir la opción de interrumpir lavado.





6.3.3 Caso de prueba: Finalizar el programa de lavado y pago



Elegimos entre la opción de Tarjeta o Efectivo y nos indica el precio a pagar:



Atención

Introduzca su tarjeta

Aceptar

Atención

Por favor, pase por caja para finalizar el pago.

Aceptar

Confirmación

Pago correcto. ¡Gracias por su visita!

Aceptar

6.3.4 Caso de prueba: Gestionar tipo Lavado

La gestión del lavado está dentro de la opción de iniciar lavado, cuando puedes elegir todas las opciones de lavado que permite el Lavadero.

MENU

INICIO

TALLER

LAVADERO

SALIR


Apar [KZ]

Tipo de agua
Agua caliente

Tipo de lavado
Lavado completo

Cristales
Ecológico

Llantas
Químico

Iniciar lavado

6.4 Mensajes de error: parsers

Para los campos vacíos o erróneos saltará un aviso indicándolo con un ejemplo.

No nos han surgido problemas con la planificacion de la ultima entrega, hemos ido realizando cada tarea según lo planificado.