

Proceso de creación del sistema de renta de carros

Documento de análisis

Grupo 7 sección 2

Tomas Camilo Ostos Medina - 202123214

Samara Martínez Jacome - 202221057

Tomas Diaz Vargas - 202220658

Índice

| | Pagina |
|--|--------|
| 1. Introducción | 3 |
| 2. Objetivos de los proyectos | 4 |
| 3. Decisiones | 5 |
| 3.1. Diseño..... | 7 |
| 3.2. Implementación..... | 9 |
| 3.3. Resultado..... | 11 |
| 3.4. Aciertos..... | 11 |
| 3.5. Desaciertos..... | 14 |
| 3.6. Retos..... | 16 |
| 4. Reflexión general | 17 |

Introducción

En el siguiente documento se buscará el análisis del proceso de creación del sistema de renta de vehículos en el entorno de java, en el paradigma de programación orientada a objetos. Con la implementación de las herramientas del curso DPOO para obtener resultados innovadores, modulando el código fuente permitiendo reutilizar el código, su organización y su mantenencia en el tiempo. Se expondrán las ventajas y desventajas de las decisiones tomadas por el grupo a lo largo de las entregas correspondientes con el impacto general en la aplicación del sistema. Consecuente a esto, se reflexionará sobre los objetivos alcanzados por el grupo, no solo para retroalimentar el mismo trabajo y mejorar futuros proyectos con este paradigma y el aprendizaje de cualquiera que tenga acceso a la información de desarrollo e implementación de este trabajo.

Objetivos de los proyectos

Proyecto 1

En el proyecto 1 se buscaba la creación de un sistema de renta de carros, con 3 subsistemas (sistema de alquiler, sistema de información de la compañía, sistema de inventario de vehículos), que cohesionados permitirían la funcionalidad total del alquiler de un vehículo. Tres puntos clave acompañan la creación del proyecto:

1. Abstracción de entidades del dominio, características y sus relaciones
2. Creación de diagramas UML para la esquematización del sistema
3. Diseño de la aplicación con el paradigma orientado a objetos

Proyecto 2

En el proyecto 2 se buscaba la mejoría del proyecto 1, con la implementación de una interfaz gráfica basada en swing con el objetivo de una mejor comunicación del sistema con el usuario y un uso más intuitivo.

Tres puntos clave acompañan la creación de este proyecto:

1. Diseño y documentación de interfaces graficas
2. Implementación de interfaces graficas
3. Diseño de sistemas en base a frameworks

Proyecto 3

En el proyecto 3 se busca el análisis total del sistema, utilizando pruebas con JUnit y mediante el manejo de excepciones. Además del acoplamiento de nuevas funcionalidades y restricciones al sistema.

Cinco puntos clave acompañan la creación de este proyecto:

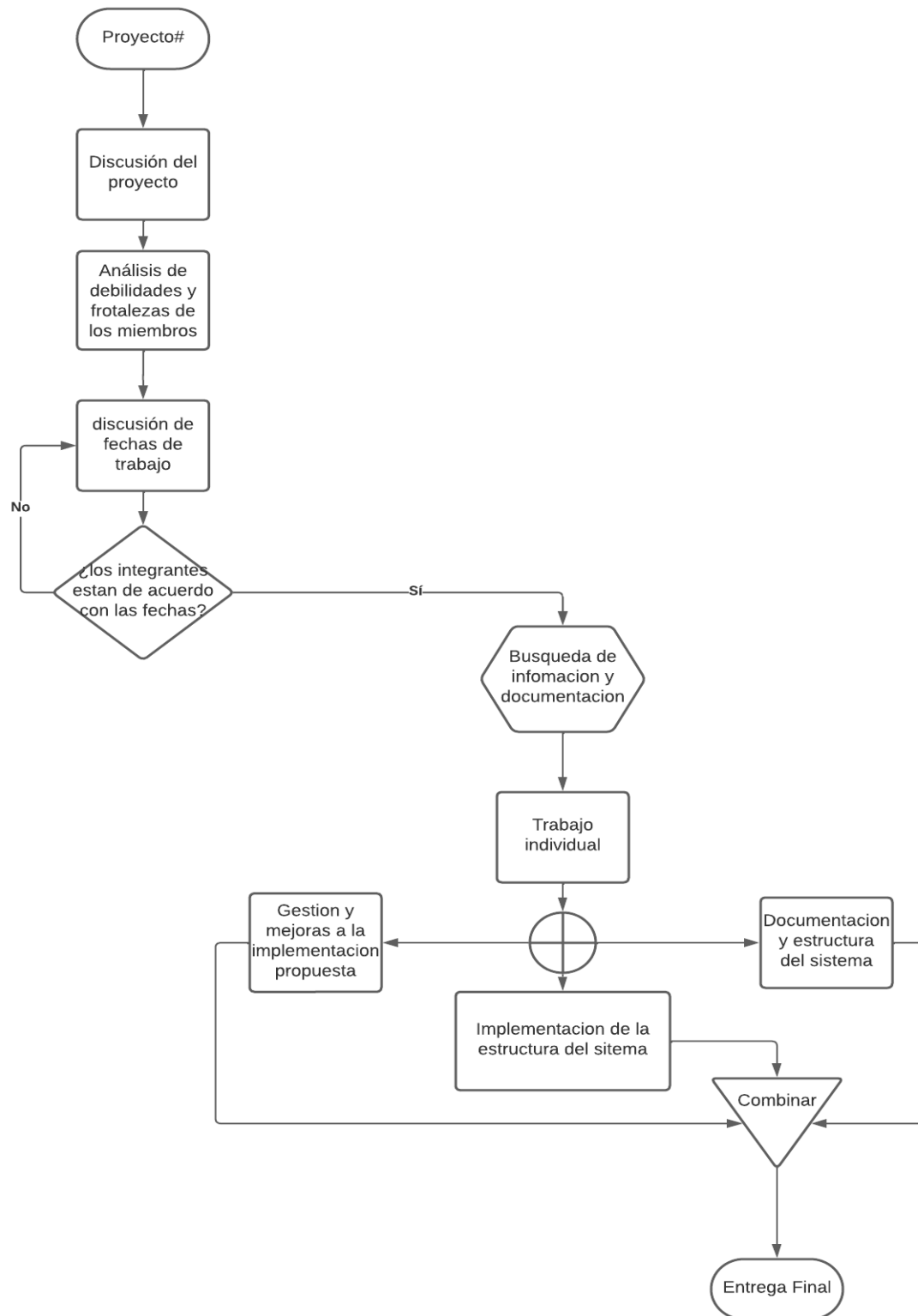
1. Análisis crítico del sistema
2. Acoplamiento de nuevas funcionalidades

3. Acoplamiento del sistema a nuevas restricciones
4. Implementación del sistema de manejo de excepciones
5. Implementación de pruebas con JUnit

Decisiones

El grupo siguió un procedimiento a la hora de realizar los trabajos pertinentes a cada entrega. Inicialmente se leía el proyecto para luego analizar las fortalezas y debilidades de los integrantes que permitirían resolver los retos presentados y discutir las fechas de trabajo. Durante el tiempo estipulado para la entrega se buscaba información y documentación necesaria el trabajo asignado a cada integrante, los integrantes realizaban su parte para luego unirse en el repositorio para su entrega final. Las decisiones involucradas en el diseño e implementación final se discutían en los dos primeros procesos y en el desarrollo del trabajo individual (pese a ser individual se incluían las propuestas de los otros integrantes), durante el desarrollo se analizaba si las decisiones en las primeras discusiones eran adecuadas o tenían que ser reemplazadas.

A continuación, una representación del proceso de trabajo del grupo.



Las decisiones que afectarían al final la implementación se pueden ver en el proceso de trabajo individual

Decisiones de diseño

Una de las principales decisiones que se tomaron fue la estructura del sistema y las funcionalidades de los principales actores (Empleados, administradores y clientes)



Se establecieron tres funcionalidades esenciales:

1. Gestión
2. Información
3. Agregación

De estas funcionalidades se desprenden todos los requerimientos y labores de los diferentes actores del sistema. Los administradores de mayor rango (administradores generales) tienen acceso a toda la gerencia de la empresa y a toda su información. Una posición menor en la jerarquía del sistema implicaría la restricción de ciertas funcionalidades.

Manejo de la información

La información se manejaría en archivos tipo TXT, por su sencillez y practicidad de manejo. Se uso una estructura sencilla que según la posición en la línea del archivo (separada por “;”) se apreciaba una característica de la clase. Se crearon los archivos según las clases abstractas presentes en el sistema.

1. Personas
2. Vehículos
3. Sedes
4. Proveedores

Y por conveniencia en el manejo de la información se creó un archivo adicional con la información de los seguros.

Se estableció la sobreescritura de la información de los archivos para evitar un almacenamiento innecesario o duplicación de esta.

Creación de reservas

El sistema generalmente presenta un bajo acoplamiento, ya que eliminar a los actores de la gerencia como lo son los administradores y empleados, no afectaría la principal característica funcional del sistema, la creación de una reserva de un vehículo. Requerimientos esenciales como el sistema de pagos y de información de clientes para generar las reservas necesariamente presentan un alto acoplamiento.

Manejo y actualización del código según necesidades

El sistema presenta un patrón de diseño MVC, por las ventajas de su implementación en la actualización a futuro del código y la división de las principales capas del proyecto.

Diseño de interfaz de usuario















En la creación de la interfaz de usuario se buscó la simplicidad sobre la estética. La interfaz se simplifico para que fuera intuitiva para el usuario, mostrando solo la información necesaria para el actor que este interactuando en el sistema.

Implementación

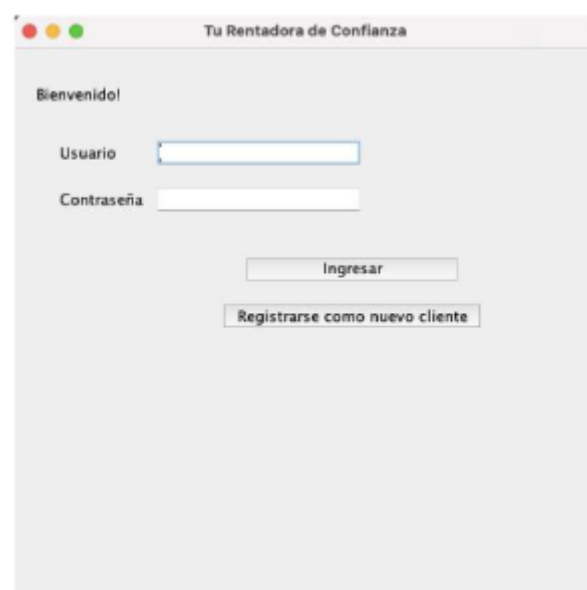
La implementación del diseño no tuvo mayor complicación o diferencia a lo expuesto en este. El diseño fue implementado en el entorno eclipse para java. La abstracción de los actores y agentes del sistema se realizó completa y con éxito. El patrón MVC se evidencia en el proyecto y la interfaz gráfica de usuario corresponde a lo diseñado inicialmente.



MVC*

| |
|---|
|  Categoria.java |
|  Cliente.java |
|  Devolucion.java |
|  Empleado.java |
|  EntregaFinal.java |
|  LicenciadeConduccion.java |
|  MetododePago.java |
|  Persona.java |
|  Proveedor.java |
|  Reserva.java |
|  Sede.java |
|  SeguroAdicional.java |
|  Tarifa.java |
|  Vehiculo.java |

Lógica*



Interfaz (inicio del sistema) *

Resultado

El resultado es el esperado por el sistema, cumple con los requisitos y su funcionamiento es el óptimo, un cliente nuevo puede registrarse y pueden generar su reserva sin ningún problema. La gestión de la información de la empresa y los empleados es la óptima. La información de uso y ejemplos de prueba se encuentran en el documento de diseño del proyecto 3.

Aciertos

Los aciertos se explicarán según las decisiones en las categorías implementadas anteriormente.

Estructura del sistema

Tomar tres funcionalidades esenciales permitió agilizar los procesos de implementación del sistema, pues muchas de las funciones de los actores se podrían realizar con cambios pequeños a una función más general.

Por ejemplo, los requerimientos que implicaran la gestión de empleados por parte de los administradores locales pudieron ser generados con la gestión de los administradores generales con los administradores locales.

Manejo de la información

Al usar los archivos TXT y sobrescribir la información presente en ellos se puede generar un menor almacenamiento, pese a que este proyecto cuenta con poca información, para proyectos con mayor complejidad y que requieran un almacenamiento mucho mayor implicaría un ahorro al evitar su uso innecesario. Además, esto permite el funcionamiento del sistema en tiempo real convirtiéndose en un sistema dinámico y competente a las necesidades de sus usuarios. Por último, el acoplamiento de información real para su uso se puede hacer de forma rápida y efectiva por la baja complejidad de la carga de sus datos.

Otro punto a destacar que sirvió de manera positiva al proyecto es la separación de la información en archivos diferentes según la información requerida para el agente que interactúa en sistema. Pese a que esto pueda ser básico para cualquier filtro de información, no realizar esto implicaría inconvenientes en la lectura y carga de esta, trayendo consigo el no poder acoplar información diferente a la original.

Creación de reservas

Ya que la principal característica del sistema puede funcionar sin la mayoría de las otras características se puede considerar un éxito la implementación del proyecto. El bajo acoplamiento de estas permite la entrada a nuevos agentes al sistema, aumentando la vida útil a futuro del proyecto, evitando los cambios estructurales complejos en el código fuente. Con esto se puede evidenciar la implementación de las principales características del paradigma de programación orientada a objetos. La creación de las reservas de vehículos es intuitiva y le permite al usuario disponer del sistema un periodo corto de tiempo, agilizando los procesos que manual y presencialmente deberían realizar los empleados y clientes. Se satisface la principal necesidad del usuario sin incurrir en gastos y tiempos innecesarios.

Manejo y actualización del código según necesidades

Al optar por el patrón de diseño MVC se prevé la solución a posibles problemas que se puedan presentar en el sistema a futuro por la inclusión de nuevos agentes, nuevos requerimientos o nuevas restricciones. La solución de estos problemas se traduce en tiempo para los desarrolladores y consecuente a esto, dinero para la empresa. De no implementarse este patrón se puede crear una desorganización en el código que tendría que requerir una documentación muy específica para el entendimiento de su funcionamiento. Si los desarrolladores iniciales del proyecto no están en desarrollo, agrega un nuevo factor de tiempo a la ecuación: familiarizar a los desarrolladores con el proyecto y su estructura.

Diseño de interfaz de usuario

Al mantenerse simple el diseño de la interfaz de usuario se obtienen diferentes beneficios a la hora de la interacción con las personas del sistema. El primer beneficio que se obtiene de esto es la reducción de los tiempos de familiarización del usuario con el sistema, independientemente del que interactúe se puede obtener un resultado similar en tiempos de entendimiento del funcionamiento. El segundo beneficio es evitar la sobreexposición a información innecesaria para el usuario. La sobrecarga de información pese a que parezca beneficiosa para el sistema al proporcionar nuevos datos al usuario, resulta contraproducente al desviar la atención de la principal característica de este, la creación de una reserva de auto. Por último, el proceso de construcción de la interfaz se agiliza mucho al no tener que incluir nuevos atributos o nuevos detalles.

Desaciertos

Estructura del sistema

Al tener tres funcionalidades del sistema generales se puede incurrir en la repetitividad de funciones de los agentes. Por lo que la jerarquización pese a tener ventajas al restringir funcionalidades, no posee diferenciación en objetivos específicos de los cargos, existiendo innecesariamente en el sistema cargos sin mucha relevancia en el funcionamiento de este. Por ejemplo, el cargo de empleado perfectamente puede se le pueden asignar otras responsabilidades más allá de consulta o creación de reservas. Esto aumentaría la complejidad de su rol y su aporte a la articulación de los agentes del sistema, como lo sería el agregarle la consulta de los proveedores en caso de que un cliente quiera programar una reserva a futuro con un vehículo que no esté disponible en ese momento.

Manejo de la información

La información al estar creada y ejecutada en el tipo de archivo TXT, pese a que su acoplamiento a nueva información puede realizarse sin un trabajo excesivo, comúnmente no se presenta en este formato. La mayoría de los archivos que almacenan información se presentan en formato XSLX, este formato se utiliza en particular por la capacidad de filtrado de información y por las características de análisis que presenta el software. Así que, para realizar un sistema mucho más cercano a la realidad, se debe entender las necesidades reales y la estructura del negocio al que se le está diseñando el sistema.

Creación de reservas

En este punto, se cree que las decisiones de diseño e implementación del sistema de creación de reservas fue el óptimo, pues se cumple con el objetivo y de forma rápida para el usuario. El sistema logro la implementación de las ventajas del paradigma de programación orientada a objetos.

Manejo y actualización del código según necesidades

Al igual que el punto anterior, se cree que la implementación del patrón MVC no posee contras en su implementación para este sistema, pues cumple con los objetivos sin crear nuevos problemas a resolver. El único “trabajo adicional ”es que la implementación de este consume un poco más de tiempo de creación en el código, pero esto es necesario para evitar inconvenientes a futuro en el proyecto.

Diseño de interfaz de usuario

La interfaz de usuario pese a ser sencilla y cumplir con el objetivo de ser una ruta de comunicación asertiva del sistema con el agente que interactúa, carece de identidad. La identidad de la empresa a la que se le desarrolla la interfaz de usuario debe estar si o si plasmada en su interfaz. El reconocimiento de lo que representa la marca, sus valores y objetivos es fundamental a la hora de la interacción con los

clientes. La psicología del consumidor debe considerarse porque eso se traduce en ganancias o pérdidas para el negocio. Un ejemplo de esto puede ser como el color cambia las experiencias y sensaciones del consumidor al interactuar, por lo que el uso estratégico de estos proporciona el cumplimiento de los objetivos y necesidades a solucionar de la empresa.

Retos

Los retos a los que se enfrentó el grupo a lo largo del diseño e implementación del sistema se resumen en tres categorías:

1. Retos de conocimiento del entorno de programación
2. Retos de abstracción de entidades del sistema
3. Retos en la coordinación del trabajo en equipo

El primer reto se produjo por el desconocimiento de la mayoría de los integrantes sobre el entorno Eclipse, Además de la estructura y funcionamiento del lenguaje de programación Java. Esto implicaría un mayor tiempo de trabajo enfocado específicamente en el aprendizaje de las funciones, requerimientos y acoplamiento de las necesidades del proyecto a la tecnología que se tenía disponible.

El segundo reto se produjo por la introducción a un paradigma desconocido para los integrantes del grupo. Por lo cual abstraer las necesidades del proyecto a este tipo de programación requería práctica y documentación, Además de que el problema a resolver al que nos enfrentábamos tenía tantas soluciones posibles que seguir un posible camino y no realizar cambios en el proceso resultaba ineficaz.

El tercer reto se debe específicamente a la coordinación del grupo en el trabajo, pese a que todos realizaron un trabajo formidable y cumplieron con las metas propuestas, el conocimiento de las fortalezas y debilidades de los integrantes con las repercusiones de estas en el trabajo final implicaba tiempo de discusiones que se podrían traducir en menor tiempo de ejecución.

Reflexión general

La reflexión general del proyecto se basa en los mecanismos necesarios para un desarrollo óptimo de los sistemas a implementar en futuras ocasiones. Se necesita organizar y distribuir el trabajo en el equipo, y profundizar en los nuevos conocimientos de utilidad al poner en marcha las soluciones a cada problema presente. La comunicación entre las partes es fundamental para el ahorro de tiempo en acoplamiento y revisión del funcionamiento de lo desarrollado.

La creatividad es esencial para poder crear el proyecto, a que al ser problemas con soluciones abstractas se requiere la capacidad de compactarlas y jerarquizarlas en una estructura más compleja, ya que con soluciones simples a subproblemas se llega a la solución de un problema mayor.

El manejo del tiempo por parte del grupo permitirá la consecución de los objetivos sin sobrecargas para los integrantes. Analizar y plantear una ruta a los retos que se enfrentan aportara a un menor tiempo de desarrollo con resultados mucho más gratificantes para el equipo. Finalmente, un entorno de trabajo con estas características permitirá la presentación de un sistema consecuente a las necesidades de la empresa o la persona que pidió el desarrollo de este.