

Proyecto 3

Reflexión y análisis del proceso

A lo largo del desarrollo del proyecto de renta de vehículos hubo muchas decisiones de diseño, y muchas cosas que salieron bien o mal, para empezar en la etapa inicial de diseño por medio del diagrama UML fue muy complicada debido a que no se tenía claridad respecto al uso de objetos e instancias en un proyecto, por ende diseñar toda la etapa inicial de una aplicación de renta de vehículos fue muy complicado y en ocasiones tuvo que haber mucha prueba de error, es decir, hacer código y una vez que funcionara lo plasmamos en el UML, no es la forma correcta pero es lo que nos facilitó nuestro primer diseño. En el proyecto 1 lo más difícil fue empezar a programar porque era difícil crear muchas clases, muchos atributos y muchos métodos y luego intentar conectar y plasmar todo de forma que funcionara correctamente, por ende optamos por realizar pequeños bloques de código y ir poco a poco conectándolo con nuestra clase principal Renta Vehículos y de ahí a la consola para ir probando el funcionamiento de los requerimientos y la carga de datos. El concepto de las instancias fue bastante confuso al principio y nos generó bastantes problemas, debido a que habían muchas clases que requieren mutuamente de otras clases y de instancias, pero nos enredamos y tuvimos muchos problemas con `AdministradorGeneral` es null, `temporada` es null, `RentaVehiculos` es null, etc. Pero al final logramos comprender el funcionamiento para no cometer ese error. Salió muy bien el uso de `archivos.txt` para almacenar toda la información de los procedimientos porque pudimos cargar información, almacenar información, leer y escribir en esos archivos con facilidad, haciendo que la aplicación sea estable. En el proyecto 2 la interfaz gráfica salió muy bien debido a que teníamos un diseño consistente y fue relativamente fácil implementarla, tuvimos que aprender a crear paneles, botones, frames y muchas otras herramientas pero conectar la lógica de la aplicación no fue muy complicado sin contar algunos problemas menores como el `RentaVehiculos` es null. En general consideramos que hubo decisiones buenas, y no considero que hubo decisiones malas, sino podrían considerarse como decisiones que no fueron precisas o que no tomaron en cuenta el futuro de la aplicación que se basan en cosas simples como el constructor de los Vehículos y el constructor de las Sedes, es decir, omitimos algunos atributos desde el comienzo que nos impidieron la realización de algunos requerimientos, debido a que implementarlos significaba cambiar en gran medida todo el diseño de la app, un ejemplo es que Las Sedes tuvieran una lista de carros dependiendo de la ubicación del carro, eso no lo implementamos desde el comienzo e implementarlo después habría que cambiar varias cosas que podrían dañar algunos requerimientos, entonces optamos por no tomar el riesgo. En el proyecto 3 fue complicado importar las librerías como `iText` y `fastxml`, pero al final lo logramos por medio de los `.JAR`, de igual forma no hubo problemas con la factura ni con la pasarela de pagos, sin embargo no implementamos algunos requerimientos debido a que eso implicaría cambiar la lógica de la aplicación y un posible fallo con otros requerimientos, esto con añadir más categorías a los vehículos y el porcentaje adicional a la prima del seguro, por otro lado, se especificó que no se quería código repetido para la aplicación para clientes, pero realmente no entendimos cómo no repetir código. Finalmente las pruebas unitarias y de integración son un muy buen recurso pero fueron complicadas en el sentido de que hacerlas conlleva una gran cantidad de trabajo debido a que hay que simular cada uno de los posibles casos, lo que puede llevar bastante tiempo. Sin duda hubo muchas cosas difíciles, pero es inmenso el conocimiento adquirido al enfrentar todos los problemas y adversidades de la aplicación.