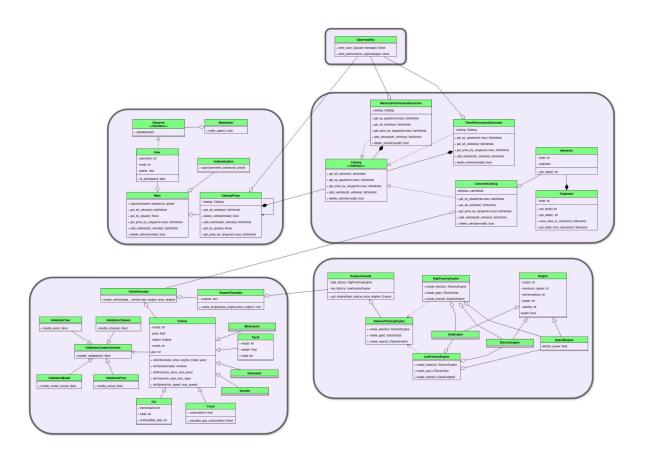
Technical Report

Juan Diego Lozada Gonzalez 20222020014

The workshop developed was a structure of some design patterns such as creationals, structural and behavioral. Some of the creationals patterns are implemented on the code are, singleton, abstract factory. Those patterns are used for the creation of the engines, due to, there is two types of engine (low, high) and the singleton for created the catalog, reducing the memory letting just create one instance of the class. For structural patterns, we implement facade for make subsystem into the system, hiding the logic the most possible, and trying to achieve Demeter Law, just the near classes know about it. The flyweight serve for reused inmutable objects, such as the engines, the engines are created just once and for reducing more and make more simple the code, when the instance is share, have the same memory reference, the proxy is used on the catalog and this add a new layer between two classes for make validations and for make a performance about the time and the memory for the catalog implement decorators, those patterns were implemented before, and some of the behavioral patterns i implement are, first it is necessary to know that the behavioral patterns are used for focus on how the classes distribute their responsabilities, taking account also the anti-patterns, SOLID and code smells. After read the tasks to complete the behavioral patterns to achieve the tasks are observer, due to it serves to notify a group of objects that have some familiar, chain of responsabilites for make the validations when we are creating a vehicle and for last memento the recover the info of the last 3 vehicles searched and the last vehicle deleted. On the diagram appear what is the idea of the functionality of it



The first step to execute the code was install the dependencies with poetry in a virtual environment of python, and run the .toml, then write on that file the initial script of the project, for then, through the CMD put 'poetry run cli_script'.

After fix the init of the project, the next step was modify the route of the JSON from 'core_subsystem' due to the project didn't recognize the file, it must be with absolute path. The idea of that file is just let enter the users that have those user and password (user and admin) respective, so i made a validation where if someone put an incorrect message and exit from the system. for that reason if someone wants to run the program NEED to change the path ot 'users.json'. But there is other option that is use a library named 'os', this library help us to make the absolute path in a dynamic way.

The next step was implemented all the patterns i said before (observer, memento and chain of responsibilities). The observer i put it on the subsystem 'core_subsystem' due to the admin will send the news to the users depending the email. The chain of responsibilities i put on the subsystem 'vehicles_subsystem' due to in this subsystem the vehicles are created so it is necessary the validations, and for the last implement was on 'catalog_subsystem' due to memento serve to restore the last versions of the vehicles in this case, both to add and delete vehicles. The rest of patterns i think it is not necessary to use it, due to, the system will be more complex and heavier. and maybe causing some anti patterns such as shotgun surgery and message chain.

And the other change was add other engine that was easy due to the structural pattern, it was just add the class HybridEngine and the code don't suffer any damage in other part. I already check with unit tests and the class and objects from this class are correctly.

In conclusion the code was hard to handle but with time it is possible but i have two exams in 4 hours and i still doing this so i decided to stop the workshop, also i presented some issues such as the push didn't serve and i tried to fix it and i don't know how but i deleted all the folder where i was working so i have to start from zero, the good thing i know how to make simple unit test and i know how to solve problems and help others (like 5 people wrote to help them to run the program) this program help us to know how is the functionality of a real system. Now i have more clear my final project.