Reference

[1] Barbosa, F. and Aguiar, A., 2013. Removing Code Duplication with Roles. 2013 IEEE 12th International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT),.

[2] Koller, H., 2016. Effects of Clean Code on Understandability: An Experiment and Analysis. Oslo, Norway: Department of Informatics University of Oslo.

[3] Fanta, R. and Rajlich, V., 1999. Removing clones from the code. Journal of Software Maintenance: Research and Practice, 11(4), pp.223-243.

[4] Martin, R., 2017. Clean Architecture: A Craftsman's Guide To Software Structure And Design. [S.l.]: Prentice Hall.

[5] Aguilar, P. and Figueira, L., 2020. Clean Architecture is not only about business logic. In: I Workshop de Tecnologia da Fatec Ribeirão Preto. [online] São Paulo.

[6] Fowler, M., 2018. Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional.

[7] Gill, G. and Kemerer, C., 1991. Cyclomatic complexity density and software maintenance productivity. IEEE Transactions on Software Engineering, 17(12), pp.1284-1288.

[8] Ivanics, P., 2017. An Introduction to Clean Software Architecture. University of Helsinki.

[9] Feathers, M., 2004. Working Effectively with Legacy Code. 1st ed. Pearson.

[10] McConnell, S., 1993. Code Complete: A Practical Handbook of Software Construction. 1st ed. Microsoft Press.

[11] Gamma, E., Helm, R., Johnson, R. and Vlissides, J., 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Pearson Deutschland GmbH.

[12] Casalino, G., Turetta, A. and Simetti, E., 2009. A three-layered architecture for real time path planning and obstacle avoidance for surveillance USVs operating in harbour fields. OCEANS 2009-EUROPE,.

[13] Moser, R., Abrahamsson, P., Pedrycz, W., Sillitti, A. and Succi, G., 2008. A Case Study on the Impact of Refactoring on Quality and Productivity in an Agile Team. Balancing Agility and Formalism in Software Engineering, pp.252-266.

[14] Runeson, P., 2006. A survey of unit testing practices. IEEE Software, 23(4), pp.22-29.