Trabalho Individual – Willian Sales

# Engenharia de software aplicada

# Factory Method / Composite / Command

## Problema a ser resolvido:

Na maioria das vezes é difícil conseguir organizar as tarefas do dia-a-dia. Fazer compras, horário marcado no dentista, ligar para o fornecedor e etc... Esses são alguns exemplos comuns, de tarefas, que toda pessoa pode vir a ter. Lembrar das atividades e compromissos que estão pendentes é muito importante para o crescimento profissional e pessoal.

Para ajudar nessa organização, irei criar um sistema de ToDoList. Esse sistema irá permitir ao usuário cadastrar as suas tarefas pendentes. O usuário poderá cadastrar tarefas individuais, ou também, projetos com títulos e subtarefas atreladas a esse projeto. Depois do usuário cadastrar as suas tarefas, o sistema irá permitir a consulta das mesmas.

## Solução:

Sistema desenvolvido em C# .NetCore, separado em camadas seguindo o DDD. Seguem abaixo, os padrões utilizados para ajudar na construção do sistema:

**Factory Method:**

O Factory Method me permitiu abstrair qual tipo de repositório ou ToDoList instanciar. As classes implementam uma interface em comum, com isso, apenas passando, por exemplo, o tipo do repositório que o cliente deseja, o Factory se responsabiliza por instanciar a devida classe concreta.

**Composite:**

O Composite me permitiu tratar de maneira uniforme o projeto (objeto composto), e o ToDo (objeto individual). Criando uma interface comum aos dois objetos, não precisei me preocupar em saber se era um objeto individual ou um composto que estava sendo usado.

**Command;**

O Command me permitiu abstrair qual front-end está invocando os métodos do ToDo ou do Projeto, criando uma interface em comum. O front-end apenas dispara o comando que deseja executar, e o Command se responsabiliza por qual método de qual classes chamar.