

## Atividade prática ATVII

Professor Dr. Eng. Gerson Penha

Softwares sugeridos:

- VSCode IDE
- Linguagem TypeScript

Contextualização:

Encontrar desafios é algo comum durante o desenvolvimento de um software. Mas, conforme se adquire experiência percebe-se que determinados problemas são recorrentes, mesmo em projetos diferentes. Estes problemas recorrentes foram estudados e para cada um uma solução padrão foi proposta. Estas soluções são denominadas de padrões de projeto (design patterns).

Padrões de projetos são soluções generalistas para problemas recorrentes durante o desenvolvimento de um software. Não se trata de um framework ou um código pronto, mas de uma definição de alto nível de como um problema comum pode ser solucionado.

A origem do conceito não é nova e, aliás, não nasceu por causa do desenvolvimento de software. Em 1978 os arquitetos Christopher Alexander, Sara Ishikawa e Murray Silverstein escreveram um livro chamado “A Pattern Language: Towns, Buildings, Construction”. Neste livro os autores catalogaram 253 tipos de problemas (ou desafios de projeto) e analisaram os detalhes de cada situação, descrevendo-as na sua essência e propondo uma solução padrão.

Em 1987, durante a segunda edição da OOPSLA (Object-Oriented Programming, Systems, Languages, and Applications), o engenheiro de software Kent Back, que posteriormente foi um dos criadores das metodologias Extreme Programming e Test Driven Development, junto com Ward Cunningham apresentaram uma palestra intitulada “Using Pattern Languages for Object-Oriented Programs”. Nesta palestra eles propuseram cinco padrões de projetos no campo da ciência da computação.

Somente em 1994 os padrões de projeto ficaram conhecidos e famosos no campo da ciência da computação. Isto quando os engenheiros de software Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides escreveram o livro “Design Patterns: Elements of Reusable Object-Oriented Software” com o objetivo de catalogar problemas comuns aos projetos de desenvolvimento de software e as formas de resolver esses problemas. Os autores catalogaram 23 padrões que utilizaram ao longo de suas carreiras. Este livro teve mais 500.000 exemplares vendidos e foi publicado em 13 idiomas. Os autores do livro ficaram conhecidos como Gang of Four (Gangue dos quatro) ou “GoF”.

Depois da publicação surgiram muitas críticas e novos padrões foram pensados e publicados. Contudo, até hoje, o Gof ainda é uma referência e, muitas vezes, a base para o desenvolvimento de novos padrões.

Atividade:

O desenvolvimento do primeiro MVP do sistema Atlantis está incompleto e isto desagradou muito você, que decidiu cobrar da equipe de desenvolvimento uma versão melhorada, com mais funcionalidades. Depois de muito esforço e duas sprints uma nova versão foi entregue. Nesta versão utilizou-se dois padrões de projetos e princípios SOLID. Os padrões utilizados foram o Singleton e o Strategy.

A nova versão, entregue pela equipe de desenvolvimento, segue como um sistema do tipo CLI, assim como a versão anterior. Neste sistema não há comunicação com algum SGBD, portanto, utilizou-se uma classe para armazenar as informações enquanto o sistema é executado, como uma base de dados em memória. Esta classe foi denominada de “Armazém”. Esta classe implementa o padrão Singleton, por isso possui o método construtor privado. A decisão de utilizar o Singleton partiu de um dos seus engenheiros de software e, por hora, será a principal forma de armazenar dados no sistema durante sua execução.

Além do padrão Singleton, a equipe de desenvolvimento fez amplo uso do padrão Strategy - aplicado em várias partes por todo o sistema. Este padrão fica evidenciado no diagrama de classe mostrado na Figura 1. Na Figura 1 se percebe que existe uma interface, denominada Menu. A partir da interface, vários outros algoritmos são definidos por classes que implementam a interface Menu.

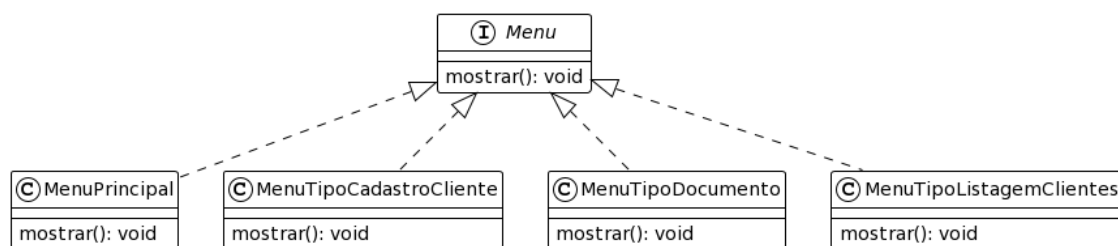


Figura 1. Aplicação do padrão Strategy.

Como você já sabe, padrões de projeto não são um framework ou um código pronto, são uma definição de alto nível de como um problema comum pode ser solucionado. Então, a equipe de desenvolvimento decidiu aplicar o padrão Strategy de um modo diferente, com uma classe abstrata, denominada de Processo. Um processo defini qualquer ação ou comportamento que o sistema deva fazer. Esta aplicação diferenciada é evidenciada no diagrama de classe mostrado na Figura 2.

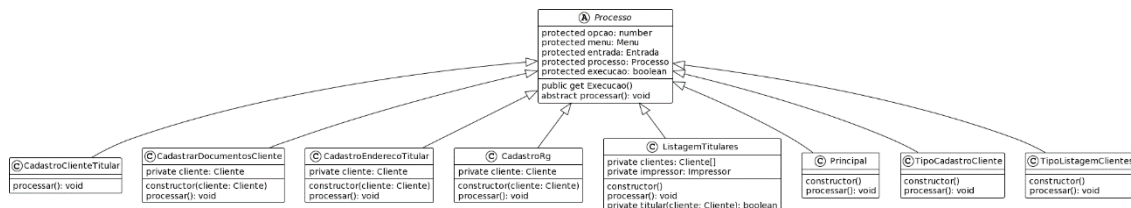


Figura 2. Implementação do padrão Strategy com classe abstrata.

O sistema começou a “ganhar corpo” e está muito próximo da sua primeira versão estável, pronta para ser apresentada para futuros clientes e colocada em produção. Contudo, ainda não suficientemente bom para um MVP. De acordo com a ideia inicial, quando você pensou em fundar a empresa, algumas funcionalidades são obrigatórias e ainda não estão implementadas.

O objetivo agora é implementar as funcionalidades obrigatórias para o primeiro MVP, mantendo a aplicação dos padrões já utilizados e, se necessário, incluindo novos padrões que você considere importantes para o sistema. Você é o responsável por esta parte e pelo preparo do sistema para seu primeiro lançamento.

A funcionalidades que devem ser implementadas são:

- CRUD de cliente, incluindo clientes dependentes.

- Listagem de clientes dependentes para um titular específico.
- Listagem do titular para o cliente dependente específico.

O projeto desenvolvido pela equipe e que servirá de base para você iniciar um novo projeto ou continuar o atual está disponível no endereço <https://github.com/gerson-pn/atvii-atlantis> ou pode ser acesso pela Figura 3.



Figura 3. Acesso a ATVII do projeto Atlantis.