

## Bootcamp IGTI

### Trabalho Prático

<b>Módulo 3</b>	<b>AST – Design Patterns, Estilos e Padrões Arquiteturais</b>
-----------------	---

#### Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Utilização de padrões, como AntiPatterns e Design Patterns.
- ✓ Análise de uso dos princípios SOLID.
- ✓ Aplicação dos princípios IoC e Injeção de Dependências.
- ✓ Análise da aplicação de práticas, como análise de *Code Smells* e Refatoração de código.

#### Enunciado

A *startup* Blueprints deseja evoluir um software de gestão de vendas para o varejo. O software foi desenvolvido utilizando uma linguagem de mercado que segue o paradigma de programação orientado a objetos. As funções atuais permitem que os clientes da Blueprints façam a gestão dos pedidos (compras), a gestão de estoque e o processamento de pagamentos por boleto bancário e cartão de crédito, além de rotinas de *background* para conciliação financeira, geração de relatórios de vendas e notificações sobre mudanças nos pedidos por e-mail e SMS.

As seguintes evoluções e alterações foram solicitadas:

- O sistema possui uma classe denominada *AutorizadorDePagamentos*. Ela concentra toda a regra de pagamento de um pedido, independente da forma de

pagamento selecionada. Entretanto, o sistema deverá aceitar outras formas de pagamento. Para o momento, foi requisitado o uso do Pix, mas a estrutura a ser criada deverá aceitar a inclusão, alteração ou remoção de outras formas de pagamento no futuro, com o menor impacto e esforço possível.

- Sempre que uma rotina de *background* precisa ser implementada, a equipe tem que criar vários artefatos do zero, como o projeto físico, as classes e a geração do *deploy*. A nova versão deverá disponibilizar um *scheduler* para a execução dessas rotinas e ser capaz de executar diferentes delas em tempos configuráveis. A equipe deverá focar apenas nas regras de negócio das rotinas. Os detalhes da execução serão configurados durante a implantação.
- Cada cliente (empresa que adquire o software) poderá configurar tipos de pedidos. Cada tipo poderá conter diferentes serviços. Quando um usuário final finalizar um pedido no sistema do cliente, esse sistema sensibilizará o software da Blueprints, e o pedido será criado com base nas configurações do tipo informado.
- Os tipos de pedidos podem ter os seguintes serviços: rastreamento da situação do pedido, pagamento por boleto, pagamento por cartão de crédito, pagamento por Pix e notificação de mudança no status do pedido (os canais de notificação podem ser escolhidos pelo cliente).

## Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Elabore um pré-projeto com uma proposta de implementação dessas mudanças. O projeto pode ser construído utilizando UML ou diretamente em Java (use como IDE o NetBeans) ou C# (use como IDE o Visual Studio 2019 Community).
2. Se optar por montar o projeto diretamente com código fonte, não será necessário implementar as regras de negócio, apenas a estrutura das classes (estados e assinatura de métodos).

3. Outros Design Patterns, além dos que foram explicitamente ilustrados no material didático, também podem ser empregados.
4. As regras de comunicação entre os objetos, incluindo a criação das instâncias dos objetos, devem ser explicitadas no projeto (seja via relacionamentos ou notas nos diagramas ou via implementação ou comentário no código).
5. Elabore um texto curto, de 2 a 3 parágrafos, descrevendo as principais decisões tomadas no projeto.