Teste Prático

Conhecimentos necessários:

* C# *Rest Web Api* com padrões de projeto
  + Utilização do *Entity Framework*
  + Utilização do ASP.NET Core para criação de Web *Api* com *Swagger*;
  + Utilização do padrão de projeto *Repository*
  + Utilização do padrão de projeto *Dependency Injection*
  + Utilização do padrão de projeto *Inversion of Control*
  + Utilização do padrão de projeto *CQRS*
  + *Utilização do padrão DDD*

Bloco 1

Neste bloco é avaliado:

* C# *Rest Web Api*
  + Utilização do *Entity Framework*
  + Utilização do ASP.NET Core para criação de Web *Api* com *Swagger*;
  + Utilização do padrão de projeto *Repository*
  + Utilização do padrão de projeto *Dependency Injection*
  + Utilização do padrão de projeto *Inversion of Control*
  + Utilização do padrão de projeto *CQRS*
  + *Utilização do padrão DDD*

Perspectivas da avaliação:

* Completude da tarefa;
* Acurácia da solução;
* Qualidade do código;
* Clareza de leitura;
* Testes de software automáticos.

# Tarefas – Bloco 1

1. Abra a solução *SistemaCompras* presente na pasta *Prova.SISPREV*;
2. O projeto Prova.SISPREV consiste em uma API simples para os objetos Produto e SolicitacaoCompra. Para objeto Produto já existe uma *controller* com métodos já implementados. Sua tarefa é evoluir, seguindo os padrões existentes na solução, o objeto de SolicitacaoCompra
3. A entidade SolicitacaoCompra no domínio, já está implementada, adicione método necessário de forma a garantir que as seguintes regras de negócio sejam validadas: **2,0** 
   1. Se o **Total Geral** for maior que 50000 a **condição de pagamento** deve ser 30 dias.
   2. O total de **itens** de compra deve ser maior que 0.
4. Crie a interface que contém o método RegistrarCompra do repositório para SolicitacaoCompra; **0,5**
5. Crie o repositório da entidade SolicitacaoCompra*;* **1**
6. Usando o comando do EntityFramework gere as migrações para a entidade recém-criada; **1,5**
7. Aplique as migrações recém-criadas utilizando o comando do EntityFramework. **0,5**

1. No projeto SistemaCompras.Application, utilizando o padrão CQRS, será necessário criar classes e interfaces do comando, são elas: **2,0** 
   1. *RegistrarCompraCommand*
   2. *RegistrarCompraCommandHandler*
2. No projeto SistemaCompras.API, crie a *controller* de SolicitacaoCompra **1,0**
3. Crie o método necessário utilizando o command criado no projeto *SistemaCompras.Application.* **1,5**