

### Questão 1

Uma classe Anemica é aquela que possui apenas propriedades e não possui métodos, ela não é inútil, pois às vezes você não necessita de métodos em sua classe. Um exemplo de uma classe anêmica é a classe ReistroJogo onde o objetivo dela é só ser uma representação dos dados. Poderíamos fazer com que ela se torne uma classe rica colocando métodos nela, por exemplo: Método para validar os números e ver se não possuem números repetidos.

### Questão 2

O SOLID não é necessário para programar, porém, é necessário para fazer um código que qualquer pessoa poderá dar manutenção e terá alta legibilidade. Um exemplo do (S)OLID é a separação em camadas e classes. Outro (O) seria o uso de um método para ativar uma entidade inativa, ao invés de alterarmos diretamente a variável Ativo, criamos um método Ativar e um Desativar. (L) Usamos bastante abstrações, normalmente eu crio um IServiceBase que implementa o CRUD e uma classe ServiceBase que implementa a interface, quando vou criar um service, uso ela para satisfazer as condições do CRUD. (I) Eu divido os serviços em ServiceBase com o CRUD e o repositório específico com as coisas específicas daquela entidade. (D) Quando usamos o dependency injection para os serviços e repositórios e registramos na classe program as equivalências como, Scoped, Transient ou Singleton.

### Questão 3

Temos a validação do view model utilizando atributos, essa validação são validações únicas para cada tipo de dado, por exemplo a idade da pessoa não pode ser menor que 18, então usamos isso os atributos para que essa validação ocorra sem precisar chamar o método do endpoint. Outro tipo de validação é a do serviço, onde vamos validar se, por exemplo, o email da pessoa já não foi usado. Cada um deles tem seu objetivo, o motivo de usarmos eles são: Mais confiabilidade nos dados, exercer as regras de serviço e ter tratativas de dados melhores.