### **Questão 1:**

**Segurança**: Previne ataques como SQL Injection, XSS e outros tipos que se utilizam de dados comprometidos, pois aceita apenas dados válidos e seguros para serem processados pela aplicação.

**Integridade dos dados**: Assegura que os dados armazenados no banco estejam consistentes e corretos, evitando problemas de corrupção de dados e garantindo a confiabilidade das informações.

**Suporte veloz:** Fornece feedback imediato e claro sobre erros de entrada, permitindo que o usuário corrija problemas rapidamente e melhore a usabilidade da aplicação.

### **Questão 2:**

### **a) Cenário 1: Cadastro de cliente**

**Nome**: Mínimo 2 caracteres, máximo 100, obrigatório, apenas letras e espaços

**CPF**: Formato válido (11 dígitos), algoritmo de validação do CPF, obrigatório, único

**Data Nascimento**: Formato de data válido, idade mínima 18 anos, não pode ser futura

**Email**: Formato de email válido, obrigatório, único no sistema

**Telefone**: Formato brasileiro válido, mínimo 10 dígitos, obrigatório

**Senha**: Mínimo 8 caracteres, pelo menos 1 maiúscula, 1 minúscula, 1 número, 1 caractere especial

**CEP**: Formato brasileiro (8 dígitos), validação via API dos Correios

**Rua**: Mínimo 5 caracteres, máximo 200, obrigatório

**Bairro**: Mínimo 2 caracteres, máximo 100, obrigatório

**Cidade**: Mínimo 2 caracteres, máximo 100, obrigatório

**Estado**: Lista de estados brasileiros válidos, obrigatório

**b) Cenário 2: Cadastro de Disciplina**

**Nome**: Mínimo 3 caracteres, máximo 100, obrigatório, único

**Carga horária**: Número positivo, mínimo 20h, máximo 200h, obrigatório

**Objetivo**: Mínimo 10 caracteres, máximo 500, obrigatório

**Ementa**: Mínimo 20 caracteres, máximo 2000, obrigatório

**Semestre**: Valores de 1 a 10, obrigatório

**Ano**: Ano atual ou futuro, formato YYYY, obrigatório

**Nome professor**: Mínimo 5 caracteres, máximo 100, obrigatório, deve existir na base de professores

### **Questão 3:**

### **IF/ELSE**

**Vantagens:**

Controle total sobre a lógica de validação com flexibilidade para validações complexas, debugging intuitivo e rápido pois as condicionais são dinâmicas.

**Desvantagens:**

Código repetitivo e longo, misturando regras de negócio com controle e com validações espalhadas pelo código, o que pode parecer confuso.

**DATA ANNOTATIONS**

**Vantagens:**

Código limpo pois a validação é automática pelo framework, de fácil reutilização e mais organizado pois há uma separação clara de responsabilidades

**Desvantagens:**

Limitações para validações muito complexas pois é menos flexível por depender do framework