Exercícios H1

Gabriel Oliveira Borges

Atividades

1-

- Sem validação, a API pode se tornar vulnerável a ataques

- Com validações bem definidas, o usuário recebe mensagens claras sobre erros nos dados enviados.

- O desenvolvedor evita lidar com erros difíceis de rastrear na lógica da aplicação.

2a – NOME: Verificar se o campo existe (required)

- Verificar tamanho (length >= 3)

- Regex: Apenas letras e espaços (^[A-Za-zÀ-ú\s]+$)

CPF: - Regex ou função de validação do CPF (verifica dígitos verificadores)

- Normalizar (remover pontos e traço)

- Verificar duplicidade no banco de dados

DataNascimento : - Validar se é uma data

- Calcular idade com base na data atual

- Rejeitar se for futura ou se a idade < 18

EMAIL: - Regex para validar formato (^[\w\.-]+@[\w\.-]+\.\w{2,}$)

- Verificar duplicidade no banco

- Normalizar (ex: caixa baixa)

Telefone: - Regex: Ex: ^\(\d{2}\)\s?\d{4,5}-\d{4}$ (ex: (11) 91234-5678)

-Remover espaços, hífens e parênteses antes de salvar

Senha:

- Regex: ^(?=.\*[A-Za-z])(?=.\*\d)(?=.\*[@$!%\*#?&])[A-Za-z\d@$!%\*#?&]{8,}$

- Não armazenar em texto puro (hash com bcrypt ou similar)

CEP:

- Regex: ^\d{5}-?\d{3}$

- Opcional: Consultar uma API externa (ex: ViaCEP) para validar existência

RUA:

- Verificar existência e comprimento mínimo

Bairro:

- Verificar existência e comprimento mínimo

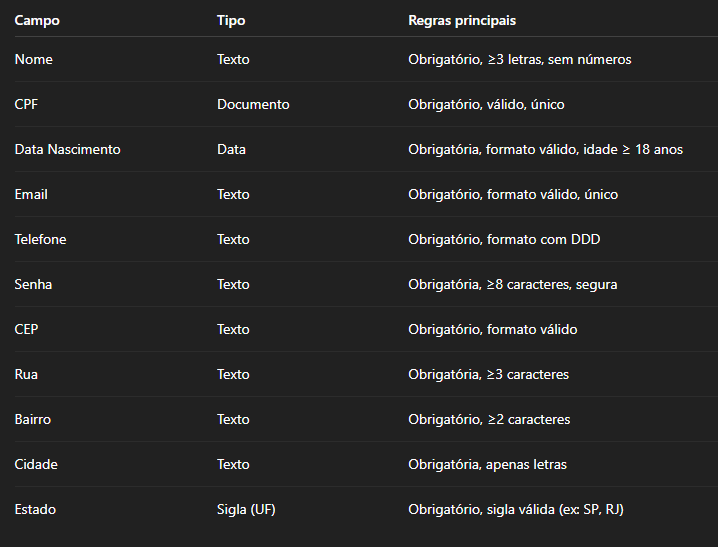
Cidade:

- Regex: ^[A-Za-zÀ-ú\s]+$

Estado:

Regex: ^[A-Z]{2}$

Validar contra uma lista de estados brasileiros (['AC', 'AL', 'AP', ..., 'SP'])

Resumo:

Cenário 2 – B

Nome:

- required

- length >= 3

- Regex: ^[A-Za-zÀ-ú0-9\s]+$ (permite letras, números e espaços)

Carga Horaria:

- required

- isNumeric

- >= 1 e <= 300 (exemplo)

Objetivo:

- Verificar se o texto tem sentido e tamanho suficiente para descrever o objetivo da disciplina

- length >= 10, length <= 1000

Ementa:

- required

- length >= 20

Semestre:

- required

- isInteger

- value in [1, 2] (ou até 10 em sistemas modulares)

Ano:

- required

- Regex: ^\d{4}$

- Validar se o ano está entre, por exemplo, 2000 e o ano atual + 1

Nome do professor:

- required

- Regex: ^[A-Za-zÀ-ú\s]+$

- Opcional: Validar se esse professor está cadastrado no sistema (chave estrangeira)

Exemplo:

3 –

Use if/else quando:

Precisa de lógica de validação mais complexa ou condicional.

Está fora de um framework que suporte data annotations.

Quer controle total da lógica.

Use Data Annotation quando:

Quer validações simples, rápidas e organizadas.

Está usando frameworks que suportam isso (como ASP.NET, Spring, etc).

Busca padronização e legibilidade.