

🧠 Lista de Exercícios: Construindo o Modelo Person e Car em Django

💡 Parte 1 – Fundamentos de Modelos Django

1. Crie um projeto e uma aplicação Django chamada `core`.
2. Configure o banco de dados e registre a aplicação no `settings.py`.
3. Crie um modelo simples chamado `Person` com os campos `name` e `email`.
4. Adicione o método `__str__` para retornar o nome da pessoa.
5. Execute as migrações e registre o modelo no admin.

📦 Parte 2 – Campos e Validações

6. Adicione os campos `gender`, `birthday`, `salary` e `status` ao modelo `Person`.
7. Crie um arquivo `validators.py` e implemente funções de validação para:
 - `validate_gender`
 - `validate_birthday_to_register`
 - `validate_salary`
 - `validate_email` (opcional, pois já há RegexValidator no campo)
8. Aplique os validadores personalizados nos campos correspondentes.
9. Adicione `choices` para o campo `gender` com as opções `M`, `F`, `O`.
10. Adicione `RegexValidator` ao campo `email` com uma expressão regular para validar o formato.

✖️ Parte 3 – Constraints e Meta

11. Adicione a classe `Meta` ao modelo `Person` com:
 - `db_table = 'person'`
 - `managed = True`
12. Adicione `UniqueConstraint` para garantir unicidade do `email`.
13. Adicione `CheckConstraint` para:
 - Validar valores permitidos em `gender`
 - Validar faixa de valores para `salary`
 - Validar que `status` seja booleano
14. Teste os constraints criando objetos válidos e inválidos.

Parte 4 – Métodos e URLs

15. Implemente o método `get_absolute_url` para retornar a URL de detalhes da pessoa.
16. Adicione o método `save` sobrescrito para chamar `full_clean()` antes de salvar.
17. Crie uma view e template simples para exibir os detalhes da pessoa usando `get_absolute_url`.

Parte 5 – Modelos Proxy e Ordenação

18. Crie o modelo `MyPerson` como proxy de `Person`.
19. Adicione ordenação por `name` e defina `verbose_name` e `verbose_name_plural`.
20. Teste o modelo proxy no admin e verifique a ordenação.

Parte 6 – Relacionamento com Outro Modelo

21. Crie o modelo `Car` com os campos:
 - `person` como `ForeignKey` para `Person`
 - `model` como `CharField` com `choices` de modelos de carros
22. Implemente o método `__str__` para exibir o carro e o dono.
23. Teste a criação de carros vinculados a pessoas.
24. Adicione o modelo `Car` ao admin e crie uma interface inline para gerenciar carros dentro do admin de `Person`.