Algumas perguntas – Django e Python

Módulo 1: Fundamentos do Django e CRUD

Este módulo introduz os conceitos básicos do Django e o ciclo CRUD (Criar, Ler, Atualizar, Excluir), a base para construir aplicações web dinâmicas.

**Seção 1.1: Criando um Projeto e App Django**

**Pergunta 1:** Como criar um projeto Django?

Antes de tudo, por questões de boa prática, deve-se criar um ambiente virtual no diretório de seu projeto. Então, instala-se o Django (pip install Django) para depois, por fim, criar-se o projeto com o comando (django-admin startproject [nome do projeto]).

**Pergunta 2:** Como criar uma aplicação (app) dentro do projeto Django?

Após a instalação do Django, vários arquivos são criados no seu diretório, dentre eles o ‘manage.py’, por meio do qual pode-se executar diversas operações. Para criar uma aplicação, é necessário navegar para a pasta raiz do projeto e executar o comando (python manage.py startapp [nome do app]) [no Windows]. Lembre-se de acrescentar o nome do app à lista ‘INSTALLED\_APPS’ no arquivo ‘settings.py’.

**Pergunta 3:** Como configurar o banco de dados no Django?

Antes de tudo, é necessário notar que o Django utiliza ORM, ou Object-Relational Mapping. Essa tecnologia consiste, em resumo, na criação e manipulação de um banco de dados por meio da programação orientada a objetos (classes e atributos) no Python. Isso permite a criação dos dados diretamente no código do projeto, sem ter de lidar com outros programas ou até mesmo programar em SQL. No ORM, as antigas tabelas criadas no SGBD com SQL agora são chamadas de modelos, que são armazenadas em um arquivo ‘models.py’.

Para configurar o banco, deve-se entrar no arquivo ‘models.py’, criar os modelos com seus devidos atributos e fazer a migração, que é o processo de criar um arquivo com as alterações feitas no banco de dados e aplicá-las. Esse processo é descrito na próxima pergunta.

**Pergunta 5:** Como aplicar migrações no banco de dados?

Depois da criação ou atualização dos modelos, deve-se executar o comando (makemigrations) para criar as migrações e também (migrate) para aplicá-las, de fato, ao banco de dados.

**Seção 1.2: Modelos e o Django Admin**

**Pergunta 4:** Como criar o modelo de Paciente no Django?

Primeiramente, percebe-se que no arquivo ‘models.py’ há a importação (from django.db import models). Esse módulo será usado para criar nossos modelos. Para um modelo ‘Paciente’, pode-se considerar o exemplo abaixo:

from django.db import models

class Paciente(models.Model):

nome = models.CharField(max\_length=100)

sobrenome = models.CharField(max\_length=100)

data\_nascimento = models.DateField()

sexo = models.CharField(max\_length=1, choices=[('M', 'Masculino'), ('F', 'Feminino')])

cpf = models.CharField(max\_length=11, unique=True)

telefone = models.CharField(max\_length=15, null=True, blank=True)

email = models.EmailField(null=True, blank=True)

endereco = models.TextField(null=True, blank=True)

data\_cadastro = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return f"{self.nome} {self.sobrenome}"

class Meta:

verbose\_name = 'Paciente'

verbose\_name\_plural = 'Pacientes'

ordering = ['nome']

Nota-se que cada atributo da classe Paciente consiste numa coluna da tabela no SGBD. Um dado atributo pode ser um texto (CharField), uma data (DateField ou DateTimeField), um e-mail (EmailField) e assim por diante. Outros comandos garantem a unicidade de um dado (unique=True) ou seu comprimento (max\_length = 11).

Sobre a classe ‘Meta’: Ela é uma classe interna que permite definir comportamentos e opções adicionais para o modelo

**Pergunta 6:** Como criar uma interface de administração para gerenciar pacientes?

**Seção 1.3: Formulários e Views**

**Pergunta 7:** Como criar um formulário de cadastro de pacientes?

**Pergunta 8:** Como criar uma view para cadastrar pacientes?

**Pergunta 9:** Como exibir uma lista de pacientes cadastrados?

**Pergunta 11:** Como editar os dados de um paciente?

**Pergunta 12:** Como excluir um paciente?

**Seção 1.4: Relacionamentos entre Modelos**

**Pergunta 13:** Como criar a relação entre Paciente e Consulta?

Módulo 2: Interface do Usuário e Experiência do Usuário (UX)

Este módulo aborda a criação de interfaces amigáveis, intuitivas e acessíveis, garantindo uma boa experiência para os usuários do sistema.

**Seção 2.1: Navegação e Layout**

**Pergunta 101:** Como estruturar o menu principal do sistema para facilitar a navegação?

**Pergunta 102:** Como criar submenus para funcionalidades específicas, como cadastro e listagem?

**Pergunta 103:** Como melhorar a usabilidade usando breadcrumbs (migalhas de pão) para navegação?

**Pergunta 107:** Como criar um layout responsivo usando Bootstrap?

**Pergunta 109:** Como utilizar cores e ícones para melhorar a usabilidade?

**Pergunta 119:** Como organizar os campos de um formulário para melhorar a experiência do usuário?

**Seção 2.2: Interatividade com AJAX**

**Pergunta 106:** Como implementar AJAX para atualizações assíncronas na interface sem recarregar a página?

**Pergunta 111:** Como implementar feedback visual claro após ações como cadastro, edição ou exclusão?

**Seção 2.3: Acessibilidade e Usabilidade Avançada**

**Pergunta 104:** Como organizar as telas de listagem de médicos, pacientes e consultas para facilitar a visualização?

**Pergunta 105:** Como implementar feedback visual claro após ações como cadastro, edição ou exclusão?

**Pergunta 108:** Como melhorar a usabilidade das telas de cadastro no sistema?

**Pergunta 110:** Como organizar a tela de edição de dados para evitar erros de usuário?

**Pergunta 112:** Como usar breadcrumbs para melhorar a navegação no sistema?

**Pergunta 113:** Como garantir que o sistema seja acessível para usuários com deficiências?

**Pergunta 114:** Como otimizar a experiência do usuário na busca de médicos ou pacientes?

**Pergunta 115:** Como implementar uma dashboard inicial para mostrar informações importantes logo após o login?

**Pergunta 116:** Como organizar a interface de criação e visualização de relatórios?

**Pergunta 117:** Como utilizar cores para indicar o status das consultas?

**Pergunta 118:** Como melhorar a acessibilidade do sistema para usuários com deficiências?

**Pergunta 120:** Como organizar a tela de exclusão de dados para evitar exclusões acidentais?

**Pergunta 65:** Como incluir mensagens de feedback ao usuário, como “Cadastro realizado com sucesso”, na interface?

**Pergunta 77:** Como implementar alertas e notificações visuais no frontend?