AVALIAÇÃO FINAL

Nome: _____

1. Qual é o objetivo principal do Django?

A) Complicar o desenvolvimento web

B) Permitir a criação de códigos complexos

C) Facilitar o desenvolvimento rápido e com código limpo

D) Criar jogos em Python

E) Substituir o Python por outra linguagem

2. Qual é uma das características marcantes do Django?

A) Rejeição a componentes reutilizáveis

B) Uso exclusivo de SQL bruto

C) Arquitetura baseada em componentes reutilizáveis

D) Documentação limitada e confusa

E) Falta de suporte para segurança

3. O que é gerado automaticamente pelo Django ao definir modelos de dados?

A) Servidor de produção

B) Interface administrativa

C) Arquivos HTML

D) Banco de dados NoSQL

E) Arquivos de configuração JSON

4. Qual comando ativa um ambiente virtual no Windows?

A) python --activate

B) venv activate

C) meu_ambiente\Scripts\activate

D) source meu_ambiente/bin/activate

E) pip activate

5. Qual das opções abaixo não é uma proteção de segurança embutida no Django?

A) Cross-Site Scripting (XSS)

B) SQL Injection

C) Cross-Site Request Forgery (CSRF)

D) Phishing

E) Segurança de dados

6. Qual é o comando para criar um novo projeto no Django?

A) python manage.py newproject

B) django-admin startproject

C) django-admin create

D) pip startproject

E) startproject new

7. Para que serve o arquivo settings.py?

A) Para gerenciar rotas

B) Para definir configurações do projeto

C) Para criar views

D) Para testar o código

E) Para criar um banco de dados

8. Qual comando é usado para iniciar o servidor de desenvolvimento no Django?

A) python manage.py startserver

B) django-admin server

C) python manage.py runserver

D) django startserver

E) runserver django

9.O que significa "aplicação" no Django?

A) Uma biblioteca externa

B) Um módulo para gerenciar o banco de dados

C) Uma coleção de códigos que realiza uma tarefa específica

D) O servidor de produção do Django

E) Uma função para criar modelos

10. Qual é a vantagem principal de criar aplicações separadas no Django?

A) Menor flexibilidade

B) Dificuldade de testes

C) Promover modularidade e colaboração

D) Reduzir a reutilização de código

E) Complicar a estrutura do projeto

11. Qual comando cria uma nova aplicação no Django?

A) django-admin newapp

B) python manage.py createapp

C) python manage.py startapp

D) startapp django

E) django startapp

12. Qual arquivo contém os modelos de dados em uma aplicação Django?

A) views.py

B) models.py

C) urls.py

D) admin.py

E) settings.py

13. Para que serve o diretório migrations?

A) Armazenar views

B) Registrar rotas

C) Gerenciar alterações nos modelos de dados

D) Configurar a interface administrativa

E) Criar formulários HTML

14. O que deve ser feito após criar uma nova aplicação no Django?

A) Alterar o arquivo manage.py

B) Adicionar a aplicação em INSTALLED_APPS

C) Criar um novo projeto

D) Excluir o diretório migrations

E) Substituir o arquivo settings.py

15. Qual é a função do arquivo urls.py?

A) Configurar o banco de dados

B) Definir as rotas do projeto

C) Gerenciar os modelos de dados

D) Criar as views do projeto

E) Escrever os testes automatizados

16. Qual é a extensão padrão do arquivo de configuração do Django?

A) .xml

B) .yaml

C) .json

D) .py

E) .ini

17. O que é necessário para verificar se o Django foi instalado corretamente?

A) pip check django

B) django --check

C) python -m django --version

D) manage.py check

E) django-admin check

18. O que é o manage.py?

A) Um gerenciador de banco de dados

B) Um script para administração do projeto

C) Um servidor de produção

D) Uma ferramenta para configurar views

E) Um arquivo de documentação

19. Qual das opções é uma vantagem do uso de modelos no Django?

A) Evitar o uso de bancos de dados

B) Reduzir a necessidade de código reutilizável

C) Representar os dados no banco de dados de forma estruturada

D) Evitar o uso de tabelas relacionais

E) Ignorar a lógica de negócios

20. Qual das opções a seguir não é gerada pelo comando startapp?

A) views.py

B) admin.py

C) urls.py

D) models.py

E) apps.py

21.Qual dos seguintes tipos de campo não é padrão em modelos do Django?

A) CharField

B) DateTimeField

C) FileField

D) TableField

E) IntegerField

22. Qual comando cria um diretório para um ambiente virtual no Python?

A) python createenv

B) python -m venv nome_ambiente

C) pip create virtualenv

D) django virtualenv

E) createenv django

23. Por que o uso de ambientes virtuais é recomendado no Django?

A) Para melhorar a performance do servidor

B) Para evitar conflitos entre dependências de projetos

C) Para aumentar a segurança do banco de dados

D) Para substituir o Python por outra linguagem

E) Para isolar as views

24. Como você acessa a aplicação Django no navegador local?

A) http://localhost:3000

B) http://127.0.0.1:8000

C) http://127.0.0.1:5000

D) http://localhost:8001

E) http://127.0.0.1:8080

25. Durante o desenvolvimento de uma aplicação Django que permite o upload de imagens, é necessário configurar o diretório de mídia para armazenar esses arquivos. Suponha que o arquivo settings.py contenha a seguinte configuração:

python

Copiar código

```
import os
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

Pergunta:

Qual é o objetivo da configuração da variável MEDIA_URL no Django?

A) Definir a URL base para o acesso aos arquivos estáticos.

B) Definir a URL base para o acesso aos arquivos de mídia no navegador.

C) Configurar o caminho do banco de dados onde os arquivos de mídia serão armazenados.
D) Especificar o caminho absoluto no sistema de arquivos para os arquivos de mídia.
E) Determinar o tipo de arquivo aceito pelo Django para upload de imagens.

26.O código abaixo representa a modelagem de um item com um campo de imagem:

python
Copiar código

```
from django.db import models

class Item(models.Model):
    nome = models.CharField(max_length=100)
    descricao = models.TextField()
    preco = models.DecimalField(max_digits=10, decimal_places=2)
    criado_em = models.DateTimeField(auto_now_add=True)
    imagem = models.ImageField(upload_to='itens_imagens/', null=True, blank=True)

    def __str__(self):
        return self.nome
```

Pergunta:
Qual é o propósito do parâmetro upload_to no campo imagem?
A) Definir o diretório absoluto onde as imagens serão armazenadas.
B) Configurar a URL de onde as imagens poderão ser acessadas.
C) Especificar o nome do arquivo a ser atribuído à imagem carregada.
D) Definir o diretório relativo dentro de MEDIA_ROOT onde as imagens serão armazenadas.
E) Definir o tipo de imagem que será aceito pelo campo imagem.

27. Ao criar um formulário baseado em um ModelForm para permitir o upload de imagens no Django, o seguinte código foi implementado:

python

Copiar código

```
from django import forms
from .models import Item

class ItemForm(forms.ModelForm):
    class Meta:
        model = Item
        fields = ['nome', 'descricao', 'preco', 'imagem']
```

Pergunta:

Durante o envio de um formulário de upload de imagens, o que o Django usa para acessar os arquivos enviados?

A) request.GET

B) request.FILES

C) request.POST

D) request.DATA

E) request.REQUEST

28. No Django, as variáveis globais podem ser úteis para armazenar configurações ou dados que permanecem consistentes durante a execução da aplicação. Qual das alternativas abaixo é uma abordagem correta para criar variáveis globais em um projeto Django?

A) As variáveis globais devem ser criadas apenas no arquivo settings.py.

B) Variáveis globais podem ser definidas diretamente no arquivo settings.py ou em um módulo dedicado.

C) Variáveis globais podem ser armazenadas apenas em variáveis de ambiente, e não no código do Django.

D) As variáveis globais devem ser armazenadas apenas no banco de dados.

E) As variáveis globais podem ser armazenadas em arquivos CSV.

29. Quando você usa context processors no Django para disponibilizar variáveis globais em templates, como elas devem ser acessadas no arquivo de template HTML?

A) As variáveis globais podem ser acessadas diretamente no código Python usando o

comando settings.VARIABLE_NAME.
B) As variáveis globais podem ser acessadas diretamente no template HTML usando {{ VARIABLE_NAME }}, desde que estejam definidas no context processor.
C) As variáveis globais devem ser sempre passadas manualmente para cada template no arquivo views.py.
D) Não é possível acessar variáveis globais em templates, elas só podem ser usadas em views.
E) As variáveis globais precisam ser salvas em arquivos JSON para serem acessadas nos templates.

30. Qual das boas práticas mencionadas na apostila deve ser seguida ao usar variáveis globais em um projeto Django?
A) Colocar todas as variáveis globais em templates para garantir sua acessibilidade.
B) Usar variáveis globais para armazenar senhas e chaves de API diretamente no código.
C) Limitar o uso de variáveis globais, preferindo encapsular lógica dentro de funções ou classes.
D) Criar variáveis globais para todos os dados sensíveis da aplicação.
E) Usar variáveis globais em todos os componentes do projeto sem restrições.

31. Quais são as principais diferenças entre bancos de dados relacionais e não relacionais, e em que situações cada tipo é mais adequado?
A) Bancos de dados relacionais são ideais para grandes volumes de dados não estruturados, enquanto bancos não relacionais são ideais para dados com estrutura fixa.
B) Bancos de dados relacionais utilizam tabelas com linhas e colunas, sendo ideais para dados com estrutura fixa e relacionamentos bem definidos. Já bancos de dados não relacionais são mais flexíveis e são usados para dados não estruturados ou semi-estruturados.
C) Bancos de dados não relacionais são sempre mais rápidos que os relacionais em qualquer tipo de operação.
D) Bancos de dados relacionais não permitem consultas complexas, enquanto bancos não relacionais são mais eficientes em consultas avançadas.

E) Bancos de dados não relacionais são usados apenas para armazenar dados em tempo real, enquanto bancos relacionais são usados exclusivamente para transações financeiras.

32. No processo de modelagem de dados em bancos de dados relacionais, qual é o objetivo da normalização?

A) Criar um modelo de dados físico sem preocupações com redundâncias.
B) Dividir os dados em várias tabelas para evitar redundâncias e garantir a integridade dos dados.
C) Agrupar todos os dados em uma única tabela para facilitar a consulta.
D) Ignorar as relações entre tabelas para simplificar a estrutura do banco de dados.
E) Criar tabelas separadas para cada usuário do sistema.

33. Qual comando SQL é utilizado para inserir novos dados em uma tabela e qual é o formato correto da sintaxe para adicionar um usuário chamado "Maria", de 25 anos, à tabela "usuarios"?

A) O comando **UPDATE** é utilizado, com a sintaxe:

sql
Copiar código
UPDATE usuarios SET nome = 'Maria', idade = 25;

B) O comando **SELECT** é utilizado, com a sintaxe:

sql
Copiar código
SELECT * FROM usuarios WHERE nome = 'Maria';

C) O comando **INSERT** é utilizado, com a sintaxe:

sql
Copiar código
INSERT INTO usuarios (nome, idade) VALUES ('Maria', 25);

D) O comando **DELETE** é utilizado, com a sintaxe:

sql
Copiar código
DELETE FROM usuarios WHERE nome = 'Maria';

E) O comando **JOIN** é utilizado, com a sintaxe:

sql
Copiar código
JOIN usuarios ON nome = 'Maria';

34. Qual é a diferença entre o comando INNER JOIN e os outros tipos de junção no SQL?

A) O **INNER JOIN** retorna todas as linhas de ambas as tabelas, mesmo que não haja correspondência entre elas.
B) O **INNER JOIN** retorna todas as linhas de uma tabela, mesmo que não haja correspondência na outra.
C) O **INNER JOIN** retorna apenas as linhas que têm correspondência em ambas as tabelas, enquanto outras junções podem retornar linhas de uma ou ambas as tabelas, mesmo sem correspondência.
D) O **INNER JOIN** é utilizado apenas em tabelas relacionadas por uma chave primária.
E) O **INNER JOIN** não permite fazer combinações de dados de mais de duas tabelas.

35. O Django utiliza o SQLite como banco de dados padrão para o desenvolvimento de aplicações. Qual das seguintes configurações é necessária para utilizar o SQLite no Django?

A) É necessário instalar o SQLite manualmente e configurar o banco no arquivo settings.py.
B) O Django automaticamente cria o arquivo do banco de dados no diretório do projeto após rodar o comando de migração.
C) O banco de dados SQLite precisa ser configurado com as credenciais de acesso no arquivo settings.py.
D) É necessário rodar um comando para instalar o driver SQLite antes de começar a

utilizar o banco de dados.
E) O banco de dados SQLite precisa de um servidor externo para funcionar corretamente.

36.Caso você queira usar o banco de dados PostgreSQL em uma aplicação Django, além de modificar o arquivo settings.py, qual dos seguintes passos adicionais é necessário?

A) Instalar a biblioteca mysqlclient para conectar o Django ao PostgreSQL.
B) Configurar o banco de dados PostgreSQL como default no arquivo settings.py.
C) Instalar a biblioteca psycopg2 para permitir a conexão entre o Django e o PostgreSQL.
D) Definir o banco de dados como postgresql e configurar a variável HOST como localhost no arquivo settings.py.
E) Configurar o banco de dados PostgreSQL com as credenciais diretamente no painel de administração do Django.

37. No Django, um modelo é utilizado para representar o banco de dados de forma abstrata. Qual das afirmações abaixo é verdadeira sobre a definição de modelos no Django?

A) Cada modelo no Django representa diretamente uma tabela no banco de dados, com seus atributos correspondendo às colunas dessa tabela.
B) O Django permite que modelos sejam definidos apenas como listas de dados, sem a necessidade de herdar de qualquer classe.
C) Os atributos de um modelo no Django não podem ser configurados, sendo fixos e definidos automaticamente pelo sistema.
D) Ao definir um modelo, o Django cria automaticamente a tabela no banco de dados, sem necessidade de migrações.
E) Os modelos no Django são definidos utilizando apenas campos numéricos, sem a possibilidade de usar texto.

38. Quais dos seguintes tipos de campos são usados no Django para armazenar, respectivamente, texto curto e números inteiros?

A) CharField para números inteiros e IntegerField para texto curto.

B) IntegerField para números inteiros e CharField para texto curto.

C) DateTimeField para texto curto e IntegerField para números inteiros.

D) BooleanField para números inteiros e CharField para texto curto.

E) EmailField para números inteiros e BooleanField para texto curto.

39. Após modificar um modelo no Django, qual é o comando correto para gerar as migrações necessárias e aplicar as alterações ao banco de dados?

A) python manage.py update para atualizar o banco de dados automaticamente.

B) python manage.py makemigrations para gerar as migrações, seguido de python manage.py migrate para aplicá-las.

C) python manage.py syncdb para criar as tabelas automaticamente.

D) python manage.py applymigrations para aplicar migrações sem gerar arquivos.

E) python manage.py resetdb para apagar e recriar o banco de dados com as alterações.

40. Quando utilizamos o Django ORM para realizar consultas, o que o método .filter() faz?

A) Retorna todos os objetos da tabela sem nenhum filtro.

B) Retorna um único objeto que atende ao critério especificado.

C) Retorna um queryset com objetos que atendem a um critério específico.

D) Exclui todos os objetos que atendem ao critério especificado.

E) Retorna um queryset com objetos que não atendem ao critério especificado.

41. O que é um relacionamento de um-para-um no Django?

a) Quando um registro de uma tabela está associado a múltiplos registros de outra tabela.

b) Quando um registro de uma tabela está diretamente associado a um único registro de outra tabela.

c) Quando múltiplos registros de uma tabela estão associados a múltiplos registros de outra tabela.
d) Quando não há nenhum tipo de relacionamento entre as tabelas.

42. Qual campo é utilizado no Django para implementar um relacionamento de um-para-um?
a) ForeignKey
b) ManyToManyField
c) OneToOneField
d) CharField

43. Qual é a principal diferença entre os relacionamentos de um-para-muitos e muitos-para-muitos no Django?
a) No relacionamento um-para-muitos, um registro de uma tabela está associado a vários registros da outra tabela, mas não vice-versa.
b) No relacionamento muitos-para-muitos, um registro de uma tabela pode estar relacionado a apenas um registro de outra tabela.
c) No relacionamento um-para-muitos, ambos os lados têm registros relacionados de forma bidirecional.
d) No relacionamento muitos-para-muitos, não há necessidade de uma tabela intermediária.

44. Como você implementaria um relacionamento de um-para-muitos entre os modelos Produto e Categoria no Django?
a) Usando o campo OneToOneField no modelo Produto.
b) Usando o campo ForeignKey no modelo Produto.
c) Usando o campo ManyToManyField no modelo Produto.
d) Usando o campo ForeignKey no modelo Categoria.

45. O que acontece quando o parâmetro on_delete=models.CASCADE é utilizado em um relacionamento de chave estrangeira no Django?
a) O objeto relacionado é apenas desmarcado, mas não excluído.
b) O objeto relacionado é mantido, mas seu campo de chave estrangeira é removido.

c) O objeto relacionado é excluído quando o objeto principal é excluído.
d) O objeto principal é excluído quando o objeto relacionado é excluído.

46. Qual campo no Django é utilizado para criar relacionamentos de muitos-para-muitos?
a) ForeignKey
b) OneToOneField
c) ManyToManyField
d) CharField

47. O que a função select_related() no Django ORM faz?
a) Ela realiza consultas mais rápidas, evitando o uso de chaves estrangeiras.
b) Ela faz uma consulta de junção para otimizar o carregamento de dados relacionados, útil para relacionamentos um-para-um e um-para-muitos.
c) Ela carrega dados relacionados apenas quando solicitado pelo usuário.
d) Ela cria índices de pesquisa para melhorar a performance das consultas.

48. Como você validaria o campo email para garantir que ele termine com o domínio '@example.com' no Django?
a) Usando a validação padrão do campo EmailField.
b) Criando um método clean() no modelo que verifica o domínio do e-mail.
c) Definindo o parâmetro unique=True no campo email.
d) Usando o campo CharField com a validação manual.

49. Qual método do Django é chamado automaticamente para validar e limpar os dados antes de salvar um objeto?
a) clean_<campo>()
b) save()
c) clean()
d) validate()

50. Como você pode personalizar a interface administrativa do Django para exibir determinados campos e adicionar filtros de pesquisa?

a) Modificando a classe ModelAdmin e registrando-a no admin.site.

b) Usando a função add_filters() no arquivo models.py.

c) Alterando diretamente a tabela do banco de dados.

d) Configurando o arquivo settings.py para definir a interface.

GABARITO

Nome: _____

1.	11.	21.	31.	41.	
2.	12.	22.	32.	42.	
3.	13.	23.	33.	43.	
4.	14.	24.	34.	44.	
5.	15.	25.	35.	45.	
6.	16.	26.	36.	46.	
7.	17.	27.	37.	47.	
8.	18.	28.	38.	48.	
9.	19.	29.	39.	49.C	
10.	20.	30.	40.	50.	

GABARITO OFICIAL

1. C	11. C	21. D	31.B	41. B	
2. C	12. B	22. B	32. B	42. C	
3. B	13. C	23. B	33. C	43. A	
4. C	14. B	24. B	34. C	44. B	
5. D	15. B	25. B	35. B	45. C	
6. B	16. D	26. D	36. C	46. C	
7. B	17. C	27. B	37.A	47. B	
8. C	18. B	28. B	38. B	48. B	
9. C	19. C	29. B	39. B	49.C	
10. C	20. C	30. C	40. C	50. A	