

1. (0,5) Defina Herança, um dos princípios de orientação a objetos.

2. (1.0) Qual o verbo/método utilizado para atualização parcial dos dados no padrão RestFull:
 - a. POST
 - b. OPTIONS
 - c. PUT
 - d. UPDATE
 - e. Nenhuma das anteriores

3. (0.5) Qual das alternativas abaixo tem como resultado 25?
 - a. $2 * 5 / (2 - 1)$
 - b. $2 * 5 / (2 - 1) * 2$
 - c. $2 * 5 / ((2 - 1) * 2)$
 - d. $(2 * 5 / ((2 - 1) * 2))^2$
 - e. $2 * 5 / ((2 - 2) * 2)^2$

4. (1.0) A trecho de código abaixo está correto? Justifique sua resposta.

```
class Pessoa:

    def __init__(self):
        pass

    def andar(self, qtd_passos):
        return qtd_passos

    def andar(self, qtd_passos, *args, **kargs):
        return qtd_passos * args[0]

if __name__ == '__main__':

    joao = Pessoa()

    joao.andar(2)
    joao.andar(2, 2)
```

1. Use o sqlite como banco de dados padrão;
2. Um cliente pode ter mais de um endereço cadastrado
3. Um cliente tem que selecionar qual dos endereços cadastrados é o endereço principal.

Cliente	Endereço
Foto (usar upload de arquivo)	Logradouro
Nome	Bairro
Sobrenome	Cidade
CPF	Estado
RG	Número
Telefone	
Email	

1. (2.0) Faça um CRUD das tabelas Cliente e Endereço através do Django Admin.
2. (5.0) Desenvolva uma WebApi utilizando o Django REST Framework, com a seguinte estrutura:
 - Rota: api/cliente/
 - GET: Retorna todos clientes;
 - POST: Cadastra um novo cliente;
 - Rota: api/cliente/<pk-cliente>/
 - GET: Retorna um cliente específico;
 - PUT: Editar cliente específico completamente;
 - PATCH: Editar um cliente parcialmente;
 - DELETE: Remove logicamente o cliente;
 - Rota: api/cliente/<pk-cliente>/endereco/
 - GET: Retorna os endereços de um cliente;
 - POST: Cadastra um novo endereço para o cliente;
 - Rota: api/cliente/<pk-cliente>/endereco/<pk-endereço>/
 - GET: Retorna um endereço específico caso pertença ao cliente;
 - PUT: Editar endereço específico completamente caso pertença ao cliente;
 - PATCH: Editar endereço específico parcialmente caso pertença ao cliente;
 - DELETE: Remove logicamente o endereço do cliente caso pertença ao cliente;

Considerando Status Codes 200 para sucesso de GETs, PUTs e PATCHs; 201 para sucesso de POSTs e; 422 para JSONs inválidos.