

TESTE ÁREA DE DESENVOLVIMENTO

ENTREGA/INSTRUÇÕES:

A entrega deve ser feita em um repositório Github, privado (se o repositório estiver público, será automaticamente eliminado);
Após finalizar é necessário adicionar o usuário **deva5solutions** como colaborador para que possamos ter acesso ao seu código;
Readme explicando as etapas necessárias para rodar o projeto;
Código limpo será um diferencial;
Use e abuse de :Criatividade, Teste Unitários, MVC, SOLID, Boas Práticas de Codificação, Performance.

TESTE FUNCIONAL:

1. Observe o seguinte algoritmo:

```
início
    leia x
    leia y
    z <= x*y + 5
    se Z <= 0 então
        Resultado <= "A"
    senão
        se z <= 100 então
            Resultado <= "B"
        senão
            Resultado <= "C"
    fim-se
fim-se
escrever: Z, Resultado
fim
```

Simule a execução do algoritmo, e em seguida complete o quadro com os valores finais das variáveis Z e Resultado para os respectivos valores de entrada de X e Y:

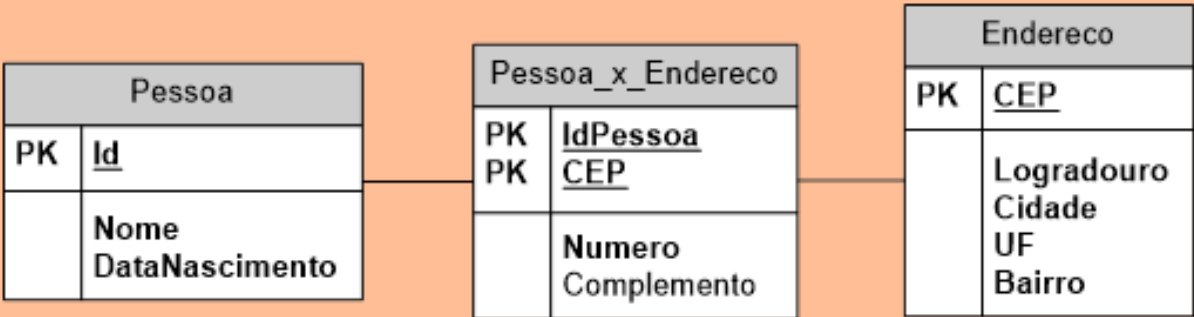
Variáveis			
X	Y	Z	Resultado
3	2	11	B
150	3	455	C
7	-1	-2	A
-2	5	15	B
50	3	115	C

2. Em uma escola, a média final é dada pela média aritmética de três notas. E a mesma tem o seguinte esquema de avaliação:

Média	Resultado
0 até 4,9	Aluno em recuperação
5 até 6.9	Aluno em prova final
7 até 10	Aluno aprovado

Desenvolva um fluxograma que a partir da entrada das três notas mostre a situação do aluno.
No caso do aluno em recuperação e prova final, mostre também quanto falta para o aluno necessita para atingir a média.

TESTE BANCO DE DADOS:



3. De acordo com o modelo acima, construa os comandos para inserir e para atualizar os campos da tabela “Pessoa”:

4. De acordo com o modelo acima, construa o comando para retornar o nome da pessoa e o endereço completo (CEP, Logradouro, Bairro, Cidade, UF, Numero e Complemento).

DESAFIO

Você foi contratado para integrar equipe de desenvolvimento da **"Godzilla Local Filmes"** uma empresa de locadora de filmes que só aluga filmes da saga **Gojira** e devido à grande procura só é permitido que um cliente alugue um filme de cada vez. Com isso você ficou responsável por criar uma API REST que gerencie uma lista de filmes para **Godzilla Local Filmes**, de modo que tarefas como, busca por título ou ano, aluguel, e consultas de filmes possam ser realizadas pelo usuário.

Crie um banco de dados, que armazena os filmes alugados. Esse banco deve garantir a unicidade, ou seja, apenas 1 registro por aluguel.

Para isso, você precisa desenvolver um programa utilizando a linguagem JAVA ou C#.

Exemplo:

```
{
  "filmes": [
    {
      "filmeId": 1,
      "titulo": "Godzilla 200",
      "diretor": "Takao Okawara",
      "estoque": 1
    },
    {
      "filmeId": 12,
      "titulo": "Godzilla Contra Ataca",
      "diretor": "Motoyoshi Oda",
      "estoque": 5
    },...
  ]
}
```

TESTE 1:

Você deve disponibilizar um endpoint **"/godzilla"**. Esse serviço permite o aluguel de um filme somente se a locadora possuir este filme em estoque.

Caso o aluguel do filme escolhido seja permitido, você deve retornar **HTTP 200-OK**, caso contrário um **HTTP 403-FORBIDDEN**

TESTE 2:

Disponibilizar um endpoint “localdora/godzilla” um **HTTP GET** que nos retorne os dados referente ao filme especificado. A pesquisa deve retornar o filme que contenham o parâmetro informado no título, não é necessário informar o título inteiro do filme, assim sendo, pode retornar mais de um filme.

TESTE 3:

Disponibilize um serviço extra de usuários, no qual permita incluir um novo cliente “usuários/usuário ” que responde um HTTP POST e ao realizar a autenticação do usuário ,deve retornar um o Token para ser utilizado nas próximas requisições.

Exemplo

```
{
  "email": "cliente@cliente.com",
  "senha": "123@"
}
```

Resposta:

```
{
  "auth": true,
  "usuario": {
    "id": 5,
    "email": " cliente@cliente.com ",
    "nome": "cliente"
  },
  "token": "
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQss"
}
```

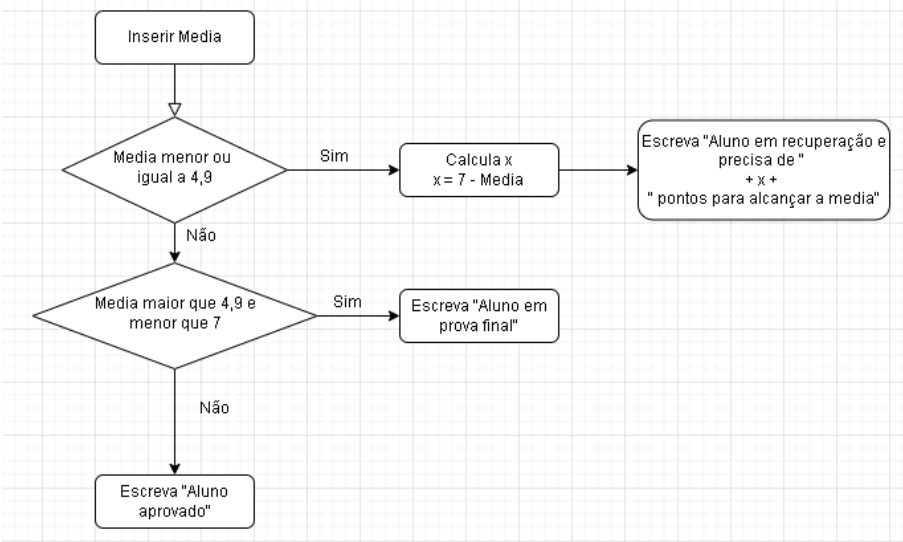
***** BOA SORTE *****

Respostas:

Questão 1:

Variáveis			
X	Y	Z	Resultado
3	2	11	B
150	3	455	C
7	-1	-2	A
-2	5	15	B
50	3	115	C

Questão 2:



Questão 3 e 4:

insert into Pessoa(nome, dataNascimento) Values
(‘Cesar Augusto’, ‘2002-12-05’),
(‘Pedro Pedrosa’, ‘1996-11-15’),
(‘Pedro Faustino’, ‘1998-11-15’);

insert into Endereco(cep, logradouro, cidade, uf, bairro) Values
(03612050, ‘Travessa neide’, ‘São Paulo’, ‘SP’, ‘Penha’),
(03920420, ‘Rua Meireles’, ‘São Paulo’, ‘SP’, ‘Centro’);

insert into Pessoa_x_Endereco(idPessoa, cep, numero, complemento) Values
(1, 03612050, 36, ‘Casa 2’),
(2, 03920420, 42, ‘Casa 6’);

update Pessoa set dataNascimento = ‘2002-12-06’ where id = 1;
delete from Pessoa where id = 3;
select * from Pessoa;

select p.nome, pe.numero, pe.complemento, e.* from Pessoa as p,
Endereco as e, Pessoa_x_Endereco as pe
where pe.idPessoa = p.id and pe.cep = e.cep;

```
insert into Pessoa(nome, dataNascimento) Values
('Cesar Augusto', '2002-12-05'),
('Pedro Pedrosa', '1996-11-15'),
('Pedro Faustino', '1998-11-15');
```

```
insert into Endereco(cep, logradouro, cidade, uf, bairro) Values
(03612050, 'Travessa neide', 'São Paulo', 'SP', 'Penha'),
(03920420, 'Rua Meireles', 'São Paulo', 'SP', 'Centro');
```

```
insert into Pessoa_x_Endereco(idPessoa, cep, numero, complemento) Values
(1, 03612050, 36, 'Casa 2'),
(2, 03920420, 42, 'Casa 6');
```

```
update Pessoa set dataNascimento = '2002-12-06' where id = 1;
delete from Pessoa where id = 3;
select * from Pessoa;
```

```
select p.nome, pe.numero, pe.complemento, e.* from Pessoa as p, Endereco as e, Pessoa_x_Endereco as pe
where pe.idPessoa = p.id and pe.cep = e.cep;
```