

**Desafio Delphi**

**TOTVS Hospitalidade**

Entendo que esteja falando de Patterns! Gosto muito de utilizar o Visítor: É ótimo para projetos legados onde consigo aplicar novas regras sem modificar muito o legado. Um cenário bem interessante que posso aplicar e já apliquei, é um cenário de vendas, onde temos produtos que podem ser vendidos como (a vista, prazo, cartão e etc), e para cada um criarmos regras que visitam a regra de pagamento que já existe.

Eu entendo que é um modelo estrutural que uso para criar meus programas. Onde defino em três componentes (Modelo, Visualização e Controle), em algumas situações adiciono outras partições do (DAO, DTO e classes).



**DESAFIO DELPHI**

2

**Questão**

**1**

O que você entende por arquitetura MVC?

**Questão 2**

Escolha um padrão de projeto e

escreva seu conceito, objetivo e um cenário de uso.

**Questão**

**3**

Temos um novo projeto na empresa,

onde precisaremos disponibilizar ao usuário do nosso sistema acessar informações

de endereço de um determinado CEP.

Para isso,

podemos contar com a

api de viacep (https://viacep.com.br/) que nos retorna essa informação.

Porém

,

não podemos contar que o viacep esteja online em 100% do tempo e pra não corrermos o risco de darmos um

erro ao nosso cliente, podemos contar também com a apicep (https://apice

p.com/api

-

de

-

consulta/) para o caso do

viacep estar fora do ar.

Para não darmos o azar de pegarmos os 2 serviços fora do ar, contaremos com a awesomeapi

(

https://docs.awesomeapi.com.br/api

-

cep) para diminuírmos ainda mais o risco do nosso cliente receber

uma resposta

de erro.

Você consegue resolver esse problema pra gente?

O desenvolvimento é livre, porém devem ser utilizados recursos nativos da IDE (recursos de terceiros devem ser

utilizados apenas via Boss.

N

ão use componentes de terceiros qu

e preci

se ser instalado na IDE).

A solução pode ser uma aplicação VCL, FMX ou pode ser uma api desenvolvida em Horse, Datasnap ou DelphiMVC.

**Diferencias (**

**mas não obrigatório);**



A

plicaç

ão de testes unit

ários ser

á um diferencial interessante.



Uso de padrões de projetos e sua criatividade.



D

esenvolver uma api

e um aplicação para consumir essa api.



Capricho em documentação tanto pra desenvolvedor quanto para usuário final.