DESAFIO DELPHI

Questão 1

O que você entende por arquitetura MVC?

MVC (Model-View-Controller) consiste em um padrão de projeto de software utilizado principalmente em softwares Desktop, este padrão é utilizado para organizar e estruturar uma aplicação, separando as responsabilidades em três componentes

Model (Modelo): Onde será implementado os dados da aplicação e suas respectivas regras de negócio. Ele é responsável por gerir os dados, realizar cálculos e responder a solicitações dentre outros processos ligados a estrutura de regra de negocio do projeto.

View (Visão): É a interface do usuário (UI), este é responsável por apresentar os dados ao usuário de maneira visual permitindo que sejam feitas as interações entre a aplicação e a camada de Controle que falaremos a seguir.

Controller (Controle): Atua como intermediário entre o Model e a View, esta classe recebe as informações que o usuário inseriu na camada visão.

Questão 2

Escolha um padrão de projeto e escreva seu conceito, objetivo e um cenário de uso.

Conceito OO (Orientado a Objeto):
Podemos dizer que padrões de projeto OO (Orientado a Objeto) servem para que possamos reutilizar códigos, conceitos e design na construção de softwares. Basicamente seria como uma receita de um prato que podemos seguir as instruções e com certeza funcionara sem maiores problemas. A programação orientada a objetos se resume em utilizar padrões de projeto fornecendo uma forma de organizar as classes e objetos de uma forma onde podemos desenvolver sistemas mais flexíveis e com excelente reutilização de códigos, fazendo com que o projeto fique mais fácil de se manter deixando-o de maneira escalavel e de fácil manutenção.

Cenário de Uso:
Digamos que estamos desenvolvendo um sistema de venda pela internet. Precisaremos lidar com diversas formas de pagamentos, por exemplo cartão de crédito, boleto, PIX. Ao invés de criarmos uma classe para cada forma de pagamento, podemos utilizar o padrão de projeto **"***Strategy*" que permite o encapsulamento de códigos em classes separadas. Utilizando desta forma, podemos definir uma interface para o processamento de pagamentos e criar classes concretas para cada tipo de pagamento, implementando apenas a interface.

Questão 3

https://github.com/18itec/TesteAPI

Heber Cinti
(18)99752-8707