



## ATIVIDADE 01

<b>Acadêmico:</b> André Luis de Souza Lima	<b>R.A.:</b> 21150930-5
<b>Curso:</b> Engenharia de Software	
<b>Disciplina:</b> ATIVIDADE 1 - ESOFT - PROJETO, IMPLEMENTAÇÃO E TESTE DE SOFTWARE - 53/2023	
<b>Valor da atividade:</b> 0,50	<b>Prazo:</b> 31/07/2023 08:00 a 18/08/2023 23:59 Horário de Brasília

- a) Explique de forma detalhada o padrão MVC (*Model-View-Controller*);
- b) Como ele promove a separação de responsabilidades em um sistema de software;
- c) Discorra também sobre as vantagens e desvantagens do uso desse padrão de projeto, destacando casos em que sua aplicação pode ser mais adequada.

### **RESPOSTAS:**

a) É um padrão de arquitetura de desenvolvimento de sistemas que separa e desenha a concepção de softwares em camadas independentes e o transcreve em documentações formais, usando os diagramas da UML (organizam a estrutura do software). A documentação serve de consulta para a equipe do projeto e implementação, e com isso, a implantação do software é feita em três escopos distintos, com base na capacidade de atuação/domínio das informações que percorrem na implementação computacional, resolvendo um problema da vida real (um negócio). O padrão é definido em 3 camadas, a camada *Model*, a *View* e a *Controller* interagindo-se. Por meio de uma interação na View, seja via navegador - browser, via aplicativo *Mobile*, ou seja, uma interface de usuário definida pelo sistema, ela vai se comunicar via protocolo de *http*, por exemplo, com o Banco de Dados, que é a camada *Model* – onde possui as informações do negócio salva em Classes – domínio do negócio, regras, dados pessoais – sem antes passar pela camada *Controller*, a qual possui os métodos ou funções possíveis que determinado



sistema pretende resolver, consumindo um serviço, que chamamos de API (*Interface de Programação de Aplicação*), em um tipo de formatação de dados, o XML, que a informação deve se adequar para responder às requisições do usuário lá na View (interação numa interface qualquer) e devolver em respostas a ele, após o dado ter percorrido o Banco de Dados, alterando-o ou não (apenas consulta).

**b)** Segundo Sommerville (2011) o software que é projetado usando o padrão MVC, permite que as funções do sistema sejam projetadas de forma discretas, ou seja, independentes, corroborando para o baixo acoplamento do sistema. Ou seja, o padrão MVC separa as responsabilidades do software quando ele construído em separado. O “interfaceamento” construído na View fica responsável pelo *Front-End* do sistema, o qual trata da interação com o usuário (telas, botões, formulários – onde ocorre a interação do sistema com o meio externo ao domínio computacional). As camadas Controller e Model representam a o *Back-End* do sistema, onde são estruturadas as informações em padrões de Arquitetura de Software bem difundidas na Engenharia de Software como sistema cliente-servidor, microserviços, baseada em componentes e outras) que se comunicarão ao Banco de Dados do Sistema.

**c)** Um projeto arquitetônico construído no padrão MVC proporciona à equipe de desenvolvimento uma melhor eficácia na manutenção do programa, pois o sistema é dividido em escopos bem definidos. Além disso, é possível que se reaproveite parte de estruturas já criadas e testadas de programas funcionais em projetos futuros (baixo acoplamento), uma vez que as classes (Controller) estão desatreladas do View.

## **REFERÊNCIAS**

PERSEGUINE, V. R.; NASCIMENTO, E. S. **MODELAGEM DE SOFTWARE**. Maringá-Pr.: Unicesumar, 2021.

Em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>.

Acessado em 04/08/2022.