

# **INFORMATIZAÇÃO DO DIÁRIO DE CLASSE ESCOLAR UTILIZANDO AS NOVAS TECNOLOGIAS DA INFORMAÇÃO**

**ARAÚJO, Allison Vieira<sup>1</sup>, AZEVEDO, Fernando<sup>1</sup>, SILVA, Silas Prado<sup>1</sup>, TOLEDO, Thales<sup>1</sup>.**

**CARVALHO, Marcos Alberto de<sup>2</sup>**

**<sup>1</sup> Acadêmicos do 8º período do curso de Computação – UNIFENAS - Alfenas.**

**<sup>2</sup> Professor do curso de Computação, UNIFENAS - Alfenas.**

## **RESUMO**

Com o crescimento vasto, acelerado e abrangente da internet, os computadores, tablets, smartphones entre outros dispositivos móveis tornaram-se ferramentas poderosas que transmitem informações. No ambiente educacional, o uso de um diário de classe escolar ainda é imprescindível, mesmo que de forma manuscrita para obtenção e manuseio das informações sobre os alunos durante o período de ensino; de forma fácil e intuitiva a aplicação Web Diário de Classe Escolar Online, reformulará o modo com que os professores obterão o controle sobre a vida escolar dos alunos, tornando os aparelhos eletrônicos que dispõem de conexão com a Internet indispensáveis para este novo mecanismo. Com isso, o propósito desse trabalho foi o de apresentar uma ferramenta que possa servir como auxílio aos professores. Para tal, foi utilizada a linguagem CSharp em conjunto com a tecnologia ASP.NET MVC. Tal tecnologia permitiu o desenvolvimento de um software distribuído utilizando a Internet, permitindo assim manusear diversas informações. Os professores acessam a aplicação através de um navegador WEB, sendo possível ter controle de todas informações dos alunos, tais como: notas, faltas e atividades propostas pelos docentes. Os resultados e informações são armazenados em um servidor Web no qual serão obtidos quaisquer relatórios sobre a vida escolar dos alunos.

**Palavras-chave:** educação – diário de classe – auxílio pedagógico – linguagem CSharp – ASP.NET MVC

## **ABSTRACT**

With vast growth, accelerated and comprehensive Internet, OS computers, tablets, smartphones between Other Mobiles have become tools Powerful que transmit INFORMATION. No educational environment, the use of hum school class Daily STILL AND imperative, EVEN That's handwritten paragraph obtaining and handling of information about the students during the teaching period; Easy to intuitively and a Web application School Class Journal online, reshape the mode that teachers will get the Control About the school life of students, making OS devices Electronic que boasts Connecting to the Internet Indispensable For this new mechanism. With IT, the purpose of this paper was to present a tool que MAY serve How Aid TO Teachers. For such a CSharp language was used in set with ASP.NET MVC technology. Such technology allowed the hum Distributed Software development using the Internet, allowing SO handling Several Information. Teachers access an application through the hum WEB Browser, being possible to have control of all INFORMATION Student Such As: grades, absences and activities proposed by teachers. The results and information they are stored in Web server hum qua not obtained any reports about the school life of students.

## **1 INTRODUÇÃO**

Os softwares utilizados em algumas instituições de ensino têm como objetivo organizar diversos tipos de informações. Dentre elas estão informações sobre professores com suas disciplinas e métodos de avaliação e os alunos com o controle de notas e frequências para obtenção ou não da aprovação durante o período de estudo.

Mesmo com este tipo de software, os professores necessitam de um controle e detalhamento mais rigoroso destas informações na qual o mesmo não provê. Sendo assim, o meio utilizado nos dias de hoje ainda é manuscrito, através do “diário de classe” e ou outras anotações optadas por eles.

Partindo desse princípio, surgiu a ideia de desenvolver um sistema capaz de satisfazer as necessidades encontradas por eles, possibilitando um fácil manuseio, controle e acesso as informações.

Com o atual avanço das tecnologias de desenvolvimento voltadas para a internet, este sistema poderá se tornar um software, uma aplicação web, com dados, interface e lógica centralizados em um servidor web, podendo ser acessado de qualquer computador, smartphone, tablets com acesso à internet, sendo automaticamente disponibilizado para todos os usuários, quando realizado alguma alteração.

## **2 REFERENCIAL TEÓRICO**

### **2.1 MODELO CLIENTE/SERVIDOR**

Os computadores na rede seguem um modelo bem definido, chamado modelo cliente/servidor. Onde os clientes seriam basicamente os computadores dos usuários de um sistema, que estão sempre requisitando alguma informação, e o servidor seria o computador ligado à internet com a responsabilidade de fornecer essa informação (ANTONIO,2006, p.378).

Esse tipo de arquitetura é centralizado e hierárquico, onde todas as informações requeridas estão presentes no servidor, e ele a repassa a quem pede. "É como uma sala de aula: os clientes são os alunos, detentores do direito de requisitar informações ao indivíduo que as possui e tem a obrigação de fornecê-las (o professor - no caso, servidor)" (ANTONIO,2006, p.378).

A relação entre o cliente e o servidor é descrita por [Antônio \(2006, p.378\)](#) da seguinte forma:

- 1) O servidor está de prontidão, aguardando que algum cliente solicite algo;
- 2) O cliente pede uma informação ao servidor apropriado (localiza-o pelo endereço);
- 3) O servidor então, respondendo a requisição feita, fornece as informações pedidas ao cliente que a solicitou.

O uso do modelo cliente-servidor leva a uma padronização dos tipos de comunicação existentes no sistema. Onde o cliente apenas faz solicitações aos servidores e espera a resposta, enquanto o servidor apenas espera a solicitação de alguma informação e a repassa como resposta a um determinado cliente. O papel de um servidor é o mesmo para diferentes tipos de clientes, estando eles na mesma máquina ou em máquinas diferentes, ligadas a rede. A distinção fica com o software que controla a comunicação ([GLENN,2008, p.122](#)).

## **2.2 DESENVOLVIMENTO WEB**

O Desenvolvimento Web refere-se a um processo de construção de sistemas específicos para a Web, onde a finalidade principal é de proporcionar ao usuário do sistema um conjunto de funções que satisfaçam as necessidades do mesmo.

"O desenvolvimento de um sistema computacional não é uma atividade trivial, pois envolve analisar e compreender determinado problema. Quando o software é desenvolvido para a plataforma web, vários aspectos são incorporados de modo que ele possa ser acessado de forma remota e segura por meio de um navegador" ([MILETTO; BERTAGNOLLI, 2014, p. 30](#)).

De acordo com ([MILETO; BERTAGNOLLI, 2014, p. 30](#)) ao

desenvolver uma aplicação Web é necessário levar em consideração alguns aspectos, como:

- Quais tecnologias podem ser utilizadas;
- Como é realizado o processamento do sistema;
- Qual é o perfil do usuário e quais são suas tarefas na interface gráfica da aplicação;
- Como armazenar os dados.

Outro fator importante no desenvolvimento Web é a questão da segurança. Como os dados são expostos na rede, o sistema deve fornecer mecanismos seguros para a transmissão dos dados.

Segundo (NIELSEN; LORAGER,2000), "Confiança e credibilidade são questões importantes na Web, que mesmo a maior empresa só existe como algumas palavras e imagens dentro da janela de um navegador".

## 2.3 TECNOLOGIA ASP.NET

ASP.NET ou Active Server Pages.NET é versão mais recente da tecnologia de programação para web. Uma maneira que a Microsoft achou para construir potentes sites da Web que vão além de simples páginas HTML. Payne(2001,p.3)

O ASP.NET é uma solução poderosa e fácil de ser usada para desenvolver sites um pouco mais complexos, em relação a páginas Web que são usadas apenas para visualização. É possível criar páginas Web com as quais os visitantes podem interagir de diversas maneiras. As páginas ASP.NET permitem que você adicione uma nova dimensão à experiência da Web". Payne(2001,p.3)

No início, os sites Web eram muito simples, totalmente estáticos, e com o passar dos tempos novos recursos foram adicionados a eles como imagens, tabelas, etc. Mas ainda faltava algo dinâmico e então o processamento de Servidor foi introduzido. ASP.NET é uma tecnologia de servidor que engloba diferentes páginas da Web para dar aos desenvolvedores

mais poder do que nunca antes (PAYNE, 2001, p.4).

O código das aplicações desenvolvidas em ASP.NET é compilado, o que aumenta muito sua performance. Além disso ele implementa uma avançada estrutura de autenticação e autorização de acesso aos recursos. Com o ASP.NET é possível que se mantenham salvas informações entre as páginas visitadas pelo usuário e pelo gerenciamento do estado de uma aplicação.

## 2.4 ARQUITETURA MODEL VIEW CONTROLLER - MVC

O padrão arquitetural MVC foi criado e introduzido pela primeira vez na Smalltalk-76 (Linguagem de programação) em 1978, por Reenskaug enquanto ele era um cientista visitante no Palo Alto Research Laboratory Xerox (PARC), como uma solução para o problema geral de dar aos usuários controle sobre suas informações. A documentação definida naquela época era de quatro divisões, *Model*, *View*, *Controller* e *Editor*. O *Editor* é um componente temporário que a *View* cria em demanda, como a interface, que se encontra entre a visão e os dispositivos de entrada, como mouse e teclado.

Na década de 1980, Jim Althoff e outros implementaram uma versão do MVC na Smalltalk-80, usando o *Controller* diferentemente de Reenskaug, observado pelo mesmo. Em implementações mais tarde feitas por Reenskaug, descrevendo a *View*, aceitaria e lidaria com a entrada do usuário relevante para si, e o *Controller* aceitaria e lidaria com a entrada relevante para a montagem do *Controller/View* como um todo, assim passa a se chamar de Ferramenta a junção desses dois componentes.

A principal característica do MVC é a facilidade que os usuários têm no emprego de uma aplicação, pois há um espaço entre o modelo mental e o modelo digital que existe no computador conforme ilustrado na FIG. 1. O MVC traz a facilidade para o usuário na visualização e na manipulação dessas informações de forma evidente. Assim, o usuário poderá ver o mesmo elemento do modelo em diferentes situações (REENSKAUG, 1978).

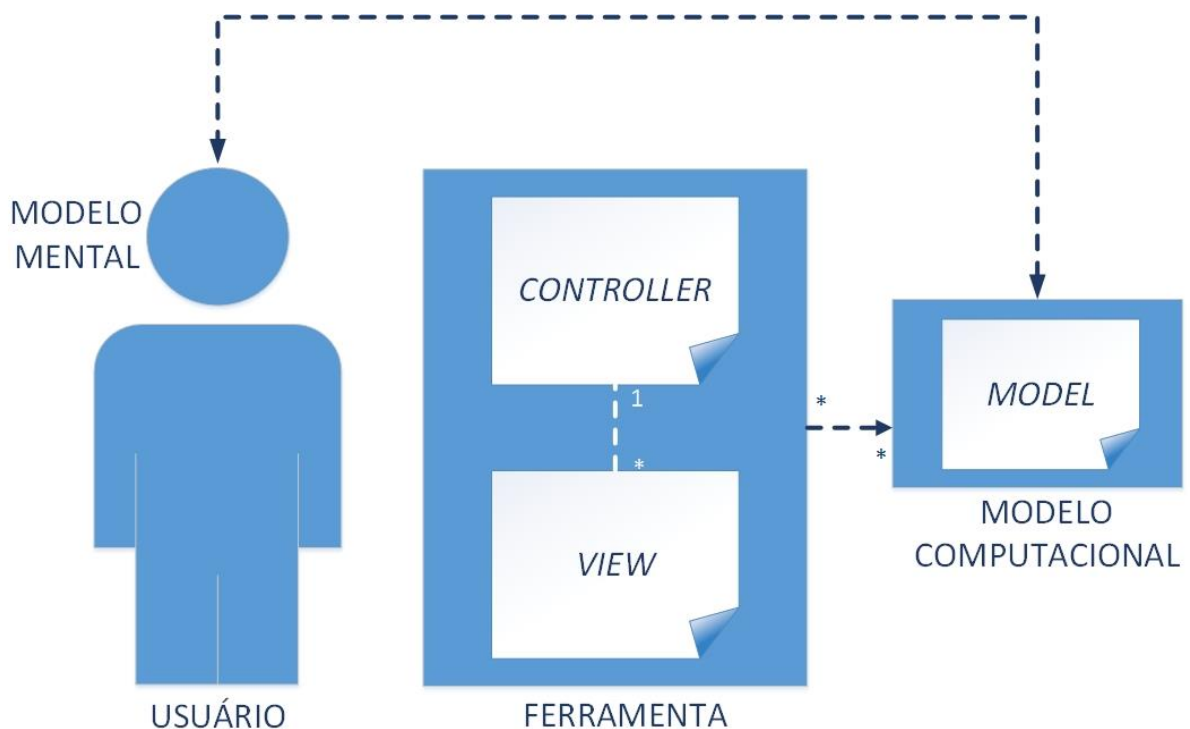


FIGURA 1 – Modelo mental MVC.

#### 2.4.1 CARACTERÍSTICAS DOS COMPONENTES *MODEL*, *VIEW* E *CONTROLLER*

A arquitetura MVC auxilia os desenvolvedores a criar aplicações separando seus principais componentes, entre eles a manipulação e armazenamento dos dados, as funções que irão trabalhar com entradas dos dados e a visualização do usuário. A arquitetura MVC especifica onde cada tipo de lógica deve estar localizada na aplicação (SANTOS et al., 2010).

O componente *Model* é o objeto de aplicação, a *View* é a interface visualizada pelo usuário e o *Controller* trabalha em relação às entradas de uma *View* e como as mesmas irão reagir (GAMMA et al, 2000).

A *Model* contém os métodos de comunicação com os dados armazenados que serão visualizados na *View* e podendo ser armazenado em um banco de dados, em um arquivo XML ou em qualquer meio de tal funcionalidade. Somente na *Model* que as operações de *create*, *reader*, *update*

e *delete* (CRUD), operações básicas em um banco de dados, podem ocorrer.

A *View* é a camada de apresentação da aplicação. Contém o que será visualizado pelo usuário final, não importando quais dados e de qual lugar tenham vindo, e sim, de como serão exibidas tais informações.

A *Controller* é responsável por todo o fluxo da aplicação. É o que move a mesma. A lógica trabalha com os dados de entrada da *View*, resolvendo qual operação utilizará da camada *Model*. A FIG. 2 faz uma representação de algumas funções e a comunicação de cada componente (ZEMEL, 2009).

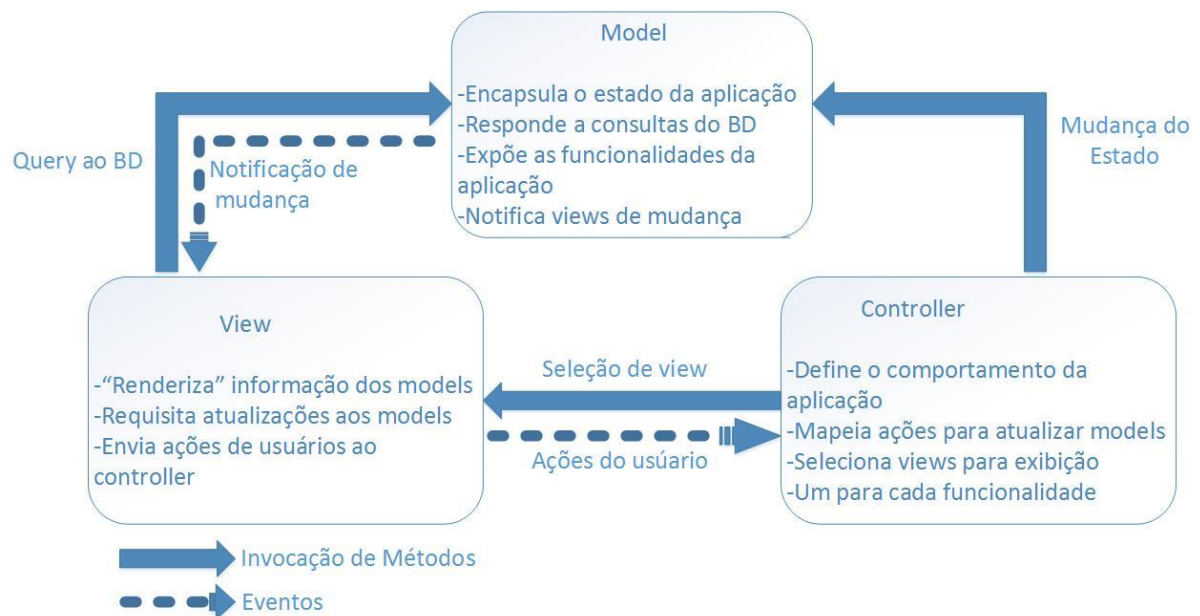


FIGURA 2 – Modelo MVC.

A arquitetura MVC tem relação com a arquitetura da aplicação, e como os componentes *Model*, *View* e *Controller* se comunicam (MACORATTI, 2002b).

## 2.5 DIFERENÇAS ENTRE AS ARQUITETURA MVC E EM CAMADAS

A regra fundamental da arquitetura em camadas é que a camada de dados não irá receber a comunicação direta da camada de apresentação.



A comunicação entre elas é feita de forma linear, ou seja, sempre passa pela camada intermediária. Sendo também considerada bidirecional, pois na hora de transportar suas informações, irá percorrer um único caminho, indo e voltando.

Já na arquitetura MVC, é fornecido uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados em uma aplicação. Conforme na FIGURA 3 as comunicações entre as camadas são feitas de forma triangular: a camada de apresentação (View) irá enviar para a camada de controle (Controller) uma modificação que será feita. Após a camada de controle receber esse sinal, ela irá atualizar a camada de lógica da aplicação (Model), e logo em seguida a camada de apresentação será atualizada.

Concluindo-se que a comunicação na arquitetura MVC é unidirecional e ela não é arquitetura em três camadas (MACORATTI, 2002b).

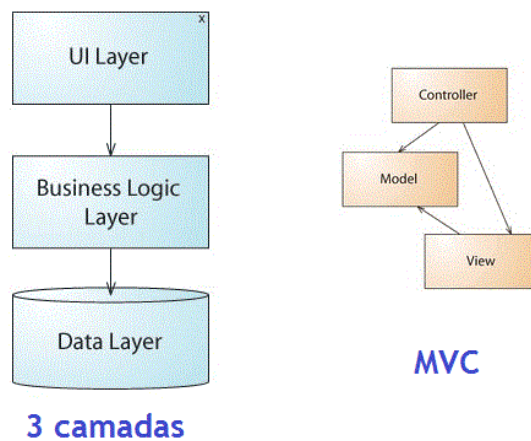


FIGURA 3. - Comunicação entre as camadas: MVC e 3 Camadas.

## 2.6 WEB DESIGN RESPONSIVO

Design Responsivo é uma abordagem de web design que tem como objetivo, elaborar sites que forneçam uma visualização fácil, uma melhora na leitura e navegação, sendo assim visualizadas em qualquer tipo de dispositivo

com acesso à rede, independentemente de sua resolução, de sua capacidade de cores, se ele é touch “sensível ao toque” ou não. Mesmo existindo diferenças entre dispositivos, tais como: smartphones, smart tv, tablets, desktops entre outros, a aplicação web continua bem apresentável, como exemplo na FIGURA 4. (Ferreira, 2013)

### **3 MATERIAL E MÉTODOS**

Para dar início à realização do projeto, foi feita a escolha do tema a ser abordado. Definiu-se utilizar o padrão arquitetural MVC e aplicá-lo no desenvolvimento de uma *WebApplication*, fazendo-se o uso da ferramenta Visual Studio 2013

Com o tema já definido, buscou-se fazer um levantamento bibliográfico em sites, livros e revistas com artigos científicos, visando a busca de informações para compreender e para enriquecer o desenvolvimento do trabalho.

Utilizando ferramentas e tecnologias escolhidas pelo grupo, será possível dar início ao desenvolvimento da aplicação, e, então, fazer uma análise detalhada do assunto proposto, das dificuldades encontradas e dos resultados obtidos.

### **4 RESULTADO E DISCUSSÕES**

#### **4.1 Vantagens e Desvantagem em usar Arquitetura Mvc**

##### **Vantagens**

O gerenciamento da complexidade se torna fácil devido à divisão da aplicação em Model, View e Controller. Esses componentes são independentes, sendo, então, possível o desenvolvimento paralelo. A inclusão de novos clientes é realizada de forma simples, apenas incluindo seus visualizadores e controles, obtendo-se um melhor reaproveitamento do código. Não só a inclusão, mas também a customização e ou a substituição é facilitada, pois o MVC é completamente extensível. Desenvolvedores que conhecem MVC terão facilidade com o código, pois o mesmo segue um padrão mais legível (legibilidade).

## **Desvantagens**

Desenvolvedores com pouca ou com nenhuma experiência em desenvolvimento de aplicações web encontram dificuldades no desenvolvimento, devido à ausência de controles prontos, ao modelo de programação orientada a eventos e ViewState. Assim, para projetos que irão utilizar o padrão MVC, a equipe terá de ter conhecimento especializado.

## **5 CONCLUSÃO**

Nesse trabalho foi desenvolvido uma aplicação WEB que pode ser utilizada nos ambientes escolares, ou não, tendo como necessidade o uso de computadores e ou dispositivos eletrônicos com acesso à Internet. Hoje em dia, muitas escolas ou mesmo os professores possuem este tipo de equipamento. Uma pergunta que pode ser feita, é a seguinte: Quais entidades de ensino possuem um sistema não manuscrito de auxílio ao professor para obtenção de informações dos alunos durante seu período de ensino?

É fato que os dispositivos eletrônicos facilitam a vida das pessoas, tanto na comunicação, lazer ou mesmo profissionalmente. Tendo isto em vista e a necessidade de facilitar o modo como os professores trabalham no controle das informações sobre os alunos, foi desenvolvida a aplicação Diário de Classe Escolar Online. No módulo do Professor, diversos conteúdos podem ser abordados. A maneira com que os mesmos lançam as notas, faltas e obtêm as informações sobre os alunos, são aprimoradas. Além disso de forma simples, é

possível se obter relatórios sobre estas informações.

Percebe-se que com esta aplicação e os meios funcionais de acesso a mesma, os professores têm mais conforto, simplicidade e segurança ao contrário da forma utilizada nos dias de hoje.

## REFERÊNCIAS

MILETO, Evandro Manara; BERTAGNOLLI, Silvia de Castro.

**Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML,CSS, JavaScript e PHP.** Bookman Editora, 2014

NIELSEN, Jakob; HOA, Loranger. **Usabilidade na Web: Projetando websites com qualidade.** 1 ed. Rio de Janeiro, 2007, 115 p.

PAYNE, Chris; **Aprenda em 21 dias Asp.net:** Editora Campus. São Paulo, 2001, 3p.

LOTAR, Alfredo; **Como Programar com Asp.net e C#.** Editora Novatec, 2d, São Paulo, 2010, 27p.

PAYNE, Chris; **Aprenda em 21 dias Asp.net:** Editora Campus. São Paulo, 2001 8p.

REENSKAUG, Trygve M. H. **MVC XEROX PARC.**1978. Disponível em: <<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>>. Acesso em: 1 janeiro de 2015.

SANTOS, Isaias et al. **Possibilidades e limitações da arquitetura mvc (model – view – controller) com ferramenta ide (integrated development environment).** 2010. 56f. Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) - Universidade José do Rosário Vellano, Alfenas, Mg.

GAMMA, Erich et al. **Padrões de Projeto: soluções reutilizáveis de software Orientado a Objetos.** Porto Alegre: Bookman, 2000.

ZEMEL, Tércio. **MVC (Model – View – Controller).** 2009. Disponível em: <<http://codeigniterbrasil.com/passos-iniciais/mvc-model-view-controller/>>. Acesso em: 7 novembro de 2014.

MACORATTI, José Carlos. **Padrões de Projeto: Design Patterns.** 15 mai.

**2002a.. Disponível** em: <[http://www.macoratti.net/vb\\_pd1.htm](http://www.macoratti.net/vb_pd1.htm)>. Acesso em: 20 de agosto de 2014.

FERREIRA, João Batista. **Media Queries, Web Design Responsivo. O que são?** 2013. Disponível em: <<http://johnylab.net/?id=126>>. Acesso em: 22 de agosto de 2014.

CARVALHO, João Antônio. **Informática para Concursos:** Editora Elsevier, 4 ed. 2009, 378p.

BROOKSHEAR, J. Glenn. **Ciência da Computação:** uma visão abrangente Editora Bookman, 7 ed, 2008, 122p.