

FACULDADE PITAGORAS DE BELO HORIZONTE

Alexandre Xavier

Plinio Luiz Ferreira de Oliveira

MVC (*Model – View – Controller*)

BELO HORIZONTE

2011

Alexandre Xavier
Plinio Luiz Ferreira de Oliveira

MVC (*Model – View – Controller*)

Projeto de pesquisa apresentado como trabalho interdisciplinar do 5º período do Curso Superior Tecnológico em Banco de Dados da Faculdade Pitágoras de Belo Horizonte.

Área de concentração: Tecnologia da Informação.

Orientador: Prof. Carlos José Giudice dos Santos

BELO HORIZONTE

2011

SUMÁRIO

	Página
1 INTRODUÇÃO.....	3
2 PROBLEMA.....	4
3 JUSTIFICATIVA	5
4 OBJETIVOS.....	6
4.1 GERAL.....	6
4.2 ESPECÍFICO.....	6
5 REFERENCIAL TEÓRICO.....	6
6 METODOLOGIA.....	10
REFERÊNCIAS.....	13

1 Introdução

A implantação de um Sistema de Informação em organizações ocorre geralmente de forma evolutiva e gradual. Inicialmente, apenas determinadas funções são automatizadas. Mais tarde, à medida que o uso da Informática vai se estabelecendo novas funções vão sendo informatizadas.

O desenvolvimento de sistemas de informação ocorre hoje quase que exclusivamente sobre banco de dados. As pesquisas na área de Banco de Dados resultaram também em um conjunto de técnicas, processos e notações para o projeto de Banco de Dados que inicialmente era feito com técnicas empíricas por alguns poucos especialistas em um Sistema Gerenciador do Banco de Dados específico. Hoje estes sistemas são desenvolvidos com auxílio de técnicas padronizadas e suportadas por ferramentas CASE (*Computer-Aided Software Engineering*), que é uma classificação que abrange todas ferramentas baseadas em computadores que auxiliam atividades de engenharia de software, desde análise de requisitos e modelagem até programação e testes. Formou-se ao longo do tempo um conjunto de conhecimentos sobre projeto de Banco de Dados que é largamente aceito e deve ser dominado por qualquer profissional de TI.

Estes conhecimentos são ministrados nas universidades, já em cursos de graduação, nas disciplinas de fundamentos de banco de dados ou mesmo em disciplinas específicas de projeto de banco de dados. O projeto de um banco de dados usualmente ocorre em três etapas. A primeira etapa, a modelagem conceitual, procura capturar formalmente os requisitos de informação de um banco de dados. A segunda etapa, o projeto lógico, objetiva definir, ao nível de Sistemas Gerenciadores de Banco de Dados, as estruturas de dados que implementarão os requisitos identificados na modelagem conceitual. A terceira etapa, o projeto físico, define parâmetros físicos de acesso ao Banco de Dados, procurando otimizar a performance do sistema como um todo.

Este trabalho tem como objeto de estudo a fase do projeto lógico, utilizando uma arquitetura denominada MVC (*Model-View-Controller*), introduzida por desenvolvedores do Smalltalk da Xerox PARC (*Palo Alto Research Center*) no final dos anos 1970.

2 Problema

O conceito da arquitetura MVC é extremamente simples, mas a sua implementação não é tão fácil. Quando pensamos em regras de negócio, pensamos no modelo da aplicação. Na instância *Model* podemos ter validações, acesso a banco, acesso à arquivos, cálculos, etc. Na instância *View* podemos mostrar dados. A instância *Controller* recebe as requisições do usuário da *View* e envia para o *Model* fazer algo com estas requisições, sendo responsável por saber quem está pedindo uma informação e a quem enviará esta informação.

O objeto de estudo e o uso desta arquitetura de desenvolvimento será usado para criar um projeto de banco de dados para uma associação sem fins lucrativos, cujo objetivo é qualificar profissionalmente pessoas com deficiência. A proposta é construir um sistema que tenha todas as facilidades de inclusão, consulta, exclusão, impressão e alteração de dados de todos os serviços necessários ao objetivo da associação. Os serviços seguintes deverão ser automatizados: alunos, professores, funcionários, doadores, patrimônio, equipamentos, entre outros. Todas as funcionalidades do sistema deverão prever acesso, consulta, inclusão e alteração dos dados via Internet. As funcionalidades terão privilégios para cada usuário.

Depois de um amplo estudo de viabilidade do negocio e de ter uma visão geral do funcionamento da associação, decidiu-se usar a arquitetura MVC para a construção do sistema e com isso surgiu a seguinte pergunta: até que ponto é possível haver ganho na eficiência de uso do sistema utilizando a arquitetura MVC na sua construção?

3 Justificativa

Neste estudo levamos em consideração a arquitetura MVC, que atualmente vê-se num dilema entre as tendências tradicionais e as novas concepções, segundo Alexander (2004).

Em primeiro lugar esta pesquisa para o banco de dados busca dar viabilidade para que a associação disponha de meios modernos, ágeis e de fácil acesso às informações já disponíveis. Embora as informações estejam disponíveis, não existe ainda um controle sistematizado, que permita fazer consultas, recuperando informações catalogadas, validação de dados, etc. Assim, justifica-se o uso da arquitetura MVC no desenvolvimento deste controle.

A partir do desenvolvimento deste controle, será possível observar uma demanda de diferentes informações com a disponibilização de conteúdo na Web e ambiente de comunicação de alta capacidade. Tais informações são imprescindíveis para o avanço das facilidades em todas as suas áreas e a integração dos usuários com outros centros. O MVC é considerado um padrão de arquitetura que existe para facilitar a manutenção da nossa aplicação, facilitar a adição de funcionalidades e facilitar a testabilidade da aplicação por muitos desenvolvedores e programadores de software, onde se buscam a proposição de investimentos que possam, de modo abrangente e agregador, constituir, a um só tempo, benefício efetivo e, também, de utilidade na proposição de outras demandas que certamente virão (KRASNER (1988), ALEXANDER (2004), DEACON (2009)).

4 Objetivos

4.1 Objetivo Geral

Analisar até que ponto é possível haver ganho na eficiência de uso do sistema utilizando a arquitetura MVC na sua construção.

4.2 Objetivo Específico

Verificar os benefícios MVC em várias categorias: separação de preocupações no código do programa, desenvolvedor de especialização e foco no negócio, o desenvolvimento paralelo por equipes separadas.

5 REFERENCIAL TEÓRICO

A condução de um projeto desenvolvido utilizando a arquitetura MVC facilita diretamente a compreensão do seu funcionamento sem perder a qualidade e principalmente sua manutenção, principal objetivo dos sistemas a serem desenvolvidos para à associação.

É necessário compreender como o MVC é organizado para podermos utilizá-lo de forma correta.

MVC - Model-View-Controller - é um padrão de design para a arquitetura de web aplicações. É um padrão amplamente adotado, em muitas línguas para implementação de frameworks, cujo objetivo é conseguir uma separação clara entre três componentes: a) Model: lógica de negócios e de processamento; b)View: interface de usuário (User Interface - UI), c)Controller: navegação e entrada.(ALEXANDER, 2004).

O mesmo autor afirma que as experiências feitas com a programação do sistema Smalltalk-76 mostraram que uma forma particular de modularidade - a separação de

três vias de componentes de aplicação - tem mais vantagens além de simplesmente tornar a vida do designer mais fácil. Esta divisão em três vias de um aplicativo implica na separação: 1 - das partes que representam o modelo subjacente da aplicação, 2 - da forma como o modelo é apresentado ao usuário e 3 - da forma como o usuário interage com ele.

Tornou-se amplamente aceita entre os engenheiros de software e arquitetos de sistemas de informação que a concepção de aplicações em conformidade explícita a esses padrões facilita a reutilização de conhecimento e experiência adquirida pelos melhores e mais brilhantes entre eles ao longo de milhares de esforços de desenvolvimento de software. Em suma, pensar em termos de padrões de design fará de você um programador melhor.

Atualmente diversos aplicativos tem acessos constantes a uma base de dados A forma de desenvolvimento desses aplicativos variam de maneira acentuada, talvez por atender um cenário ou para facilitar sua construção, ou ambos os casos. As consultas são feitas sem um planejamento prévio, comprometendo o desempenho dos acessos à base de dados e do funcionamento dos aplicativos. Com a utilização da arquitetura MVC toda a complexidade poderá ser dominada, o que irá facilitar o dia-a-dia da associação.

Dentro desta abordagem, destacamos que reconhecer os tipos de dados, as ferramentas de armazenamento e a estrutura das consultas é fundamental para a escolha adequada do tipo de aplicações. Vários problemas podem surgir quando as aplicações contêm uma mistura de acesso a dados, código de lógica comercial e código de apresentação. Tais aplicações são difíceis de manter, porque interdependências entre todos os componentes podem causar efeito cascata forte sempre que uma alteração é feita em qualquer lugar. A arquitetura de projeto *Model-View-Controller* resolve esses problemas, segundo CASTELLANO *et al* (2004), dissociando acesso a dados, lógica de negócios e dados de apresentação e interação com o usuário. De acordo com esta arquitetura, há três tipos de objetos: objetos *Model*, objetos *View*, e objetos *Controller*.

Esses mesmos autores afirmam que o MVC é uma arquitetura flexível, capaz de buscar dados para definir e validar um processo, geralmente aplicáveis em diferentes setores de negócios, para fornecer novo valor acrescentando conhecimento de serviços. A arquitetura é projetada sobre o padrão *Model-View-Controller* a fim de obter uma clara separação dos componentes, funcionalidades e cobre todo o processo de *Knowledge-Discovery in Databases* (KDD - Conhecimento das Bases de Dados) e *Knowledge-Discovery in Texts* (KDT - Conhecimento das Bases de Dados em Texto) para a extração de padrões a partir de dados estruturados e não estruturados.

A flexibilidade da arquitetura MVC tem se mostrado eficiente na construção e execução dos aplicativos com relevância determinante para escalabilidade, manutenção e funcionalidades dos sistemas; com relação direta à base de dados a arquitetura MVC consegue cobrir e modelar os dados de maneira simples e eficaz, facilitando no desenvolvimento e divisão de tarefas entre os profissionais da associação.

Conforme também afirma Morse (2004), o MVC é uma arquitetura popular para aplicações interativas. Tem considerável popularidade em programas como o Java da Sun Microsystems e aplicações corporativas. A arquitetura MVC não é limitada a Java; frameworks MVC existem para muitas línguas e sistemas, incluindo Python, PHP, ASP Microsoft e a aplicação Cocoa no ambiente Apple.

O mesmo autor considera que isto leva a um número de propriedades desejáveis que incluem: o mesmo conjunto de lógica de negócios (*Model*) pode ser usado com diferentes visões para fornecer diferentes interfaces de usuário para a mesma aplicação subjacente. Estes poderiam incluir a exibição da página web, GUIs tradicionais, ou um serviço de web, e ao separar o *Model* da *View* e do *Controller*, equipes separadas de desenvolvedores podem trabalhar em cada componente, seja em série ou em paralelo.

Diversos frameworks de diversas línguas de desenvolvimento têm migrado para a arquitetura MVC por sua flexibilidade e metodologia de trabalho. Atualmente torna-se mais rentável para a equipe de TI de uma empresa possuir o profissional que desenvolve seus aplicativos, o profissional que cuida base de dados e o profissional

que atua na conexão entre as duas partes citadas acima, pois cada profissional responsável por sua parte facilita e disponibiliza o restante da equipe para outros serviços.

O MVC é a aplicação em que objetos de diferentes classes assumem as operações relacionadas com o domínio da aplicação (*Model*), a exibição do estado do aplicativo (*View*), e a interação com o usuário (*Controller*), resume Krasner e Pope (2004)

Neste sentido, os mesmos autores afirmam que na construção de aplicações interativas, como acontece com outros programas, a modularidade dos componentes tem enormes benefícios. Isolar unidades funcionais do outro, tanto quanto possível torna mais fácil para o designer de aplicativo de entender e modificar cada unidade particular, sem ter que saber tudo sobre as outras unidades.

De suas raízes Smalltalk, o conceito MVC recentemente ganhou mais reconhecimento e foi aplicado ao nível de design, como nos padrões de design em *Java2 Platform Enterprise Edition* (J2EE). Note-se que MVC é normalmente usado como uma arquitetura específica ao invés de um quadro mais amplo de modelagem e, como tal, pode haver preocupações sobre até que ponto isso limita a sua aplicabilidade à modelagem de uma gama mais ampla de aplicações e sistemas. Um estudo aprofundado das abordagens de modelagens existentes, especialmente os que suportam a arquitetura de informação razoavelmente bem, como *Object-Oriented Hypermedia Design Method* (OOHDM) e *Web Modeling Language* (WebML), demonstra que a separação de entidades de modelagem no modelo conceitual e as entidades de interface na apresentação e navegação tem sido usados para proporcionar vantagens a essas abordagens bem estabelecidas de acordo com Lowe *et al* (2007).

Estes mesmos autores ainda sugerem que um estudo aprofundado das abordagens de modelagens existentes, especialmente os que suportam a arquitetura de informação razoavelmente bem, como OOHDM e WebML, demonstra que a separação de entidades de modelagem no modelo conceitual e as entidades de interface na apresentação e navegação tem sido usados para proporcionar vantagens a essas abordagens bem estabelecidas. Entre as vantagens, os autores apontam: melhor

compreensão dos problemas da arquitetura do sistema trazido pela separação de preocupações e a possibilidade de flexibilidade e personalização fornecidos pelo modelo conceitual com diferentes modelos de apresentação ou navegação. Embora acreditemos que estas abordagens podem fornecer suporte razoável para modelagem da arquitetura de informação, o aspecto funcional é normalmente fraco ou mesmo ausente nestas abordagens. A utilização da arquitetura MVC nos permitirá:

- A) Em *Model* - o modelo de informação que trata do armazenamento de dados e processamento de informações. Ele vai gerenciar o comportamento dos dados no domínio do aplicativo;
- B) Na *View* - lidará com a forma como a informação é apresentada visualmente, o que é a interface do sistema;
- C) No *Controller* - proporcionará interação com o usuário, ou seja, vai controlar os modelos de informação.

A partir do exposto até aqui, acreditamos que a utilização da arquitetura MVC está totalmente alinhada com os objetivos deste projeto, ou seja, haverá ganho de eficiência nos sistemas utilizados pela associação. Entretanto, esta hipótese só poderá ser confirmada a partir da utilização prática do MVC no desenvolvimento dos sistemas necessários para o correto funcionamento da associação.

6 METODOLOGIA

Na visão de Silva e Menezes (2001), a função da metodologia é mostrar e ajudar a refletir e instigar a olhar o mundo com um olhar curioso, indagador e criativo. A elaboração de um projeto de pesquisa, seja ela de qualquer natureza, necessita estar baseada em planejamento cuidadoso, reflexões conceituais sólidas e alicerçados em conhecimentos já existentes, para que seus resultados sejam satisfatórios.

Assim, em relação aos objetivos, nossa pesquisa será descritiva. Em relação à coleta de dados, esta pesquisa será bibliográfica, aplicada (experimental) e quantitativa.

De acordo com Cervo *et al* (2007, p.61), “a pesquisa descritiva procura descobrir, com a maior precisão possível, a frequência com que um fenômeno ocorre, sua relação e conexão com os outros, sua natureza e suas características”. Uma vez que nossa pesquisa tem por objetivo verificar ganho de eficiência de uso de um sistema, a partir da frequência de seu uso em consultas, alterações, validações, etc, então isto caracteriza nossa pesquisa como descritiva.

Em relação à coleta de dados, nossa pesquisa é bibliográfica porque “abrange a leitura, análise, interpretação de livros, artigos científicos, etc.” (SILVA e MENEZES, 2001). Uma das principais técnicas da coleta de dados será a pesquisa bibliográfica que é utilizada em quase todas as pesquisas. Todo material recolhido teve que ser submetido a uma triagem. Fizemos uma leitura atenta e sistemática que se faz acompanhar de anotações que, eventualmente, poderão servir à fundamentação teórica do estudo. De posse de todas as informações consideradas relevantes e necessárias passamos então para a fase de formulação de hipóteses e de construção das questões que poderiam ser abordadas junto às pessoas responsáveis pela associação.

A coleta de dados será também quantitativa, com a aplicação de questionários com uma lista de perguntas fixas do tipo de auto-preenchimento. Deste modo, será possível a obtenção de dados que possibilitem conhecer a situação dos sistemas da associação e que permitam a construção de uma linha de base para o processo de avaliação

Por último, a pesquisa será aplicada (experimental), que caracteriza-se por manipular diretamente as variáveis relacionadas com o objeto de estudo (CERVO *et al*, 2007, p.63). Como pretendemos com a pesquisa quantitativa levantar parâmetros de utilização do sistema por meio de questionários aplicados aos usuários, então nossa pesquisa caracteriza-se como aplicada (experimental).

Após a finalização da coleta de dados, deverá ser realizada uma verificação crítica para detectar possíveis falhas ou erros, evitando informações confusas ou incompletas que possam prejudicar o resultado da pesquisa. A análise e a interpretação dos dados são atividades distintas, porém estreitamente relacionadas. A análise procura evidenciar as relações existentes entre o fenômeno estudado e outros fatores. A interpretação dos

dados busca dar um significado mais amplo às respostas, associando-as a outros conhecimentos (SILVA e MENEZES, 2001). As mesmas autoras ainda afirmam que a interpretação significa, em geral, a exposição do verdadeiro significado do material apresentado, em relação aos objetivos propostos. Esta fase da pesquisa é fundamental para o sucesso da mesma, já que é nesse momento que definimos os caminhos e os instrumentos mais adequados ao que queremos estudar. Portanto, a partir da análise dos dados coletados, procuraremos sintetizar os resultados obtidos para evidenciar o resultado alcançado com o estudo e com isso indicar as vantagens da arquitetura MVC, apontando a relação entre os fatos verificados e a teoria, procurando com isso dar uma contribuição para o meio acadêmico e empresarial.

Referências

ALEXANDER, Christopher. **Model-View-Controller**: A design pattern for software. 2004. Disponível em: <http://ist.berkeley.edu/as-ag/pub/pdf/mvc-seminar.pdf>. Acesso em: 16 set. 2011.

CASTELLANO, M; PASTORE, N; ARCIERI F; SUMMO, V & GRECIS ,G. Bellone de. **A Model-View-Controller architecture for knowledge discovery**. 2004. Disponível em: <http://library.witpress.com/pages/PaperInfo.asp?PaperID=14252>. Acesso em: 16 set. 2011.

CERVO, Amado; BERVIAN, Pedro A.; SILVA, Roberto da. **Metodologia Científica**. 6ª ed. São Paulo: Pearson Prentice-Hall, 2007.

DEACON, John. **Model-View-Controller (MVC) architecture**. 2009. Disponível em: <http://www.jdl.co.uk/briefings/MVC.pdf>. Acesso em: 16 set. 2011.

HEINEMAN, George T. **An instance-oriented approach to constructing product lines from layers**. 2004. Disponível em: <http://web.cs.wpi.edu/~heineman/serg/acdk.pdf>. Acesso em: 16 set. 2011.

KRASNER, Glenn E; POPE, Stephen T. **A description of the Model-View-Controller user interface paradigm in the Smalltalk-80 System**. 1988. Disponível em: www.create.ucsb.edu/~stp/PostScript/mvc.pdf. Acesso em: 16 set. 2011.

LOWE, David; HENDERSON-SELLERS, Brian and GU, Alice. **Web extensions to UML: using the MVC triad**. 2002. Disponível em: <http://dl.acm.org/citation.cfm?id=725910>. Acesso em: 16 set. 2011.

MORSE, Scot F.; ANDERSON, Charles L. **Introducing application design and software engineering principles in introductory CS courses: Model-View-Controller Java application framework**. 2004. Disponível em: http://www.wou.edu/~andersc/pubs/CCSC-NW_2004.pdf. Acesso em: 16 set. 2011.

SILVA, Cassandra Ribeiro de O. e. **Metodologia e organização do projeto de pesquisa: (guia prático)**. 2004. Disponível em: <http://www.ufop.br/demet/metodologia.pdf>. Acesso em: 24 out. 2011.

SILVA, Edna Lúcia da; MENEZES, Eстера Muszkat. **Metodologia da pesquisa e elaboração de dissertação.** 2001. Disponível em: <http://projetos.inf.ufsc.br/arquivos/Methodologia da Pesquisa 3a edicao.pdf>. Acesso em: 24 out. 2011