

## **CONCEITOS DE PROJETO NO DESENVOLVIMENTO MOBILE**

FONTES, Jonas Henrique Muniz<sup>1</sup>; JUNIOR, Gilberto Ferreira de Sousa<sup>1</sup>; LUCAS, Luiz Claudio Dias<sup>1</sup>; NOGUTI, Ricardo Akio<sup>1</sup>; REIS, José Cláudio de Sousa<sup>2</sup>

(1) Acadêmicos do curso de Ciência da Computação, UNIFENAS, Campus Alfenas.

(2) Professor do curso de Ciência da Computação, UNIFENAS, Campus Alfenas.

### **RESUMO:**

Os conceitos de projeto foram criados pela engenharia de software para um importante finalidade, a elaboração de software com qualidade. Este trabalho tem como objetivo demonstrar a aplicação de alguns conceitos de projeto no desenvolvimento de softwares web e mobile, com intuito de obter – se aplicações com qualidade. Foi utilizado para o desenvolvimento do trabalho o ambiente de desenvolvimento Visual Studio e a linguagem C# para o desenvolvimento web e Windows Phone, o ambiente de desenvolvimento Android Studio e a linguagem Java para o desenvolvimento Android e o banco de dados SQL Server em conjunto do conceito de Entity Framework. O trabalho foi realizado desenvolvendo-se softwares inicialmente sem utilização de qualquer conceito de projeto. Posteriormente foi aplicado o conceito de refatoração em todas as aplicações, incorporando ao código o conceito de independência funcional, encapsulamento de informações, modularidade e o padrão singleton. Conclui-se que com a utilização dos conceitos obteve-se aplicações mais concisas, com maior facilidade de manutenção e com maior qualidade. O trabalho permitiu demonstrar as vantagens obtidas com a aplicação dos conceitos de projeto para o desenvolvimento mobile.

**PALAVRAS CHAVE:** conceitos de projeto, engenharia de software, desenvolvimento mobile.

### **ABSTRACT:**

The concepts of project were created by software engineering for an important finality, the elaboration of software with quality. This work has the goal of showing some project concepts application in the web and mobile software development, with intention of having quality applications. For the development of this project was used IDE Visual Studio with C# programation language for Windows Phone and WebService, Android Studio Environment using Java for Android and SQL Server database with Entity Framework object relational mapper tool. The project was accomplished developing software initially without utilization of any project concept. After was applied the

refactoring concept in all the applications, incorporating to the code the functional independency concept, data encapsulation, modularity and Singleton standard. We can conclude that with the use of concepts, we got more concise applications, more facility of maintenance and higher quality. The project allowed to show the advantages acquired with the project concepts application for the mobile development.

KEYWORDS: concepts of project, software engineering, mobile development.

## **1. INTRODUÇÃO**

### **1.1. Contextualização**

Ocorre uma gradativa mudança nas preferências dos usuários para acesso a informação, o que antes era através de computadores, hoje está sendo por dispositivos moveis. O que acarreta novas necessidades que devem ser supridas por meio de novos aplicativos.

A engenharia de software busca alcançar software de alta qualidade através de uma abordagem sistemática de conceitos consagrados. Na fase do projeto, a engenharia dispõe de diversos conceitos e padrões para auxiliar nesse desenvolvimento, tais como refatoração, padrões e outros.

### **1.2. Objetivos**

#### **1.2.1. Objetivo Geral**

Verificar a aplicabilidade, vantagens e desvantagens no uso de conceitos de projeto no desenvolvimento de aplicações mobile.

#### **1.2.2. Objetivo Especifico**

Desenvolver um software mobile de alta qualidade, aplicando os conceitos de projeto de software.

### **1.3. Justificativa**

O desenvolvimento mobile é uma necessidade atual, apresentando uma maior procura por aplicativos com qualidade.

### **1.4. Hipótese**

Utilizar-se de conceitos de projeto para melhorar a qualidade do software.

## **2. REFERENCIAL TEÓRICO**

### **2.1. O início do desenvolvimento de software**

#### **2.1.1. História**

Software, segundo Deboni (2003) é o resultado da inteligência humana, onde o conhecimento de um indivíduo pode ser capturado e colocado a serviços de outros. Pressman (1995) o descreve como um conjunto de instruções de computadores que quando executados geram a função e o desempenho desejado.

#### **2.1.2. Engenharia de software**

Pressman (2011) relata que a engenharia de software é composta por um processo, que nada mais é do que um conjunto de métodos e um conjunto de ferramentas, onde juntos desenvolvem o softwares de maneira correta.

### **2.2. Os Processos de desenvolvimento de software**

#### **2.2.1. A Importância dos processos**

O objetivo principal da engenharia de software é o desenvolvimento de programas com qualidade, sendo assim são necessários certos cuidados na hora de desenvolvê-los. Destarte, surgiram os processos de software.

#### **2.2.2. Processos gerais**

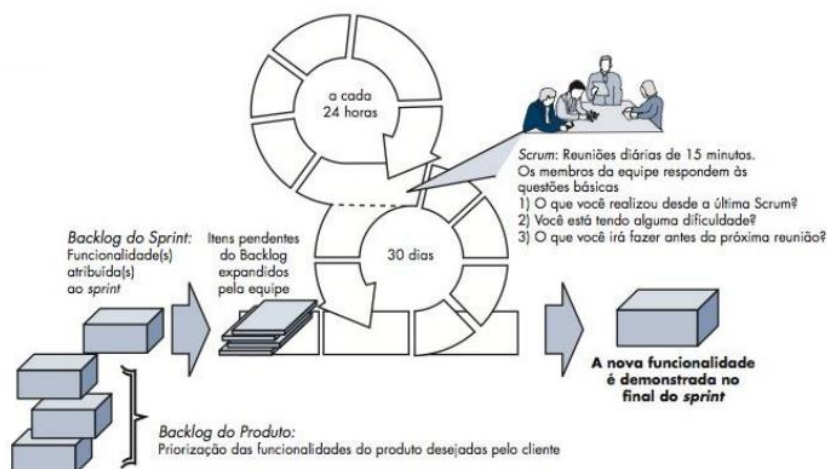
Dentre os diversos modelos de processos gerais, também nomeados de processos prescritivos, os mais utilizados segundo Souza, Barroso e Vinhas (2012) são: o Modelo em Cascata, Modelo em Espiral e o Processo Unificado.

#### **2.2.3. Processos ágeis**

Segundo os estudos de Pressman (2011), engenharia de software ágil defende a satisfação do cliente e a entrega incremental do projeto, onde o mais importante para o cliente é feito primeiro. Engenheiros de softwares, programadores, gerentes, clientes e usuários finais integram uma equipe ágil, que deve se auto-organizar e ser capaz controlar seu próprio destino.

Existem vários processos ágeis, sendo que o mais conhecido é o Scrum conforme ilustra a figura 4.

Figura 1 – Fluxo do processo Scrum



Fonte: (PRESSMAN, 2011, p. 96).

#### 2.2.4. Além dos processos

Junto aos processos de desenvolvimento, devem ser aplicados os conceitos de projeto.

Os conceitos de projeto de software são utilizados para facilitar a manutenção e o entendimento do projeto, onde cada conceito trata de um determinado domínio do problema.

### 2.3. Conceitos de projeto de software

A engenharia de software e os conceitos fundamentais de projetos de software, segundo Pressman (2011), têm sido utilizados há tempos, claro que em constante evolução. Com a evolução o grau de interesse mudou, mas todos os conceitos resistiram ao tempo, possibilitando aos projetistas de software terem uma base pela qual o melhor e mais sofisticado método de projeto possam e devam ser aplicados.

Nos tópicos a seguir serão abordados os seguintes conceitos de projeto: abstração, arquitetura, padrões, separação por interesse, modularidade, encapsulamento de informações, independência funcional, refinamento, aspectos, refatoração, conceitos de projeto orientado a objeto e classes de projeto.

#### 2.3.1. Abstração

Pressman (2011) diz a abstração em níveis surgiu de uma solução modular e consiste em 3 níveis de abstração: um nível mais alto que é uma solução expressa em termos abrangentes, um nível um pouco mais baixo que é uma descrição mais

detalhada da solução e por fim o nível mais baixo que é expressa de tal maneira que pode ser diretamente implementada.

### 2.3.2. Arquitetura

Pressman (2011) afirma que arquitetura de software é onde se estrutura ou ocorre à organização de componentes de um programa, mais conhecidos como módulos, sendo uma referência de como ocorre à organização geral do software.

### 2.3.3. Padrões

Segundo Pressman (2011), padrões são utilizados para resolver problemas que ocorrem em certos contextos específicos, ou seja, é um conhecimento já existente, testado e aprovado. É a essência de uma solução mais simples e sem erros.

### 2.3.4. Separação por interesses (por afinidades)

Pressman (2011) afirma que a separação por interesses facilita muito a resolução de qualquer problema. Por mais complexo que problema seja, se ocorrer a divisão em partes menores bem planejadas e separa-los em blocos do mesmo interesse, se torna mais fácil a administração e solução do problema.

### 2.3.5. Modularidade

A modularidade já vem sendo utilizada a muito tempo e foi um dos primeiros conceitos a ser criado.

Ocorrendo a divisão do software indefinidamente, os módulos ficam pequenos com facilidade de desenvolvimento com mínimo de esforço. Porém, há outros fatores que precisam de atenção. Referindo-se à FIG. 5, deve-se encontrar o ponto de equilíbrio, a posição M da imagem.

Figura 2 – Modularidade e Custo do Software



Fonte: (PRESSMAN, 2011, p. 96).

### **2.3.6. Encapsulamento de Informações**

O encapsulamento, segundo Pressman (2011), implica que as informações (algoritmos e dados) que não interessam a outros módulos sejam inacessíveis pelos mesmos, disponibilizando apenas os itens que lhe interessam. Enquanto as abstrações ajudam a definir as entidades que constituem o software, o encapsulamento define e impõe restrições de acesso. Com o encapsulamento, possíveis erros tem menor probabilidade de se propagar pelo software.

### **2.3.7. Independência Funcional – IF**

De acordo com Pressman (2011), Independência funcional é o resultado da modularidade, dos conceitos de abstração e encapsulamento de informações. É possível através da utilização de módulos com uma única função e sem interação excessiva com outros módulos, o que define o acoplamento e a coesão, que são os critérios qualitativos para a IF.

### **2.3.8. Refinamento**

Segundo Pressman (2011), refinamento é um processo de elaboração ou uma estratégia de projeto, onde se inicia em um alto nível de abstração, descrevendo funções ou informações conceituais, mas não informando o funcionamento interno das mesmas. A abstração vai aos poucos diminuindo o nível, melhorando o que está sendo feito e cada vez fornecendo mais detalhes. Quanto mais se refina, mais revela detalhes na medida em que o projeto avança.

No refinamento a evolução do projeto é gradual, não sendo necessário fazer tudo de uma vez.

### **2.3.9. Aspectos**

Em conformidade com Pressman (2011) a identificação dos aspectos é importante para que o projeto possa acomodá-los à medida que ocorre o refinamento e a modularização, e é implementado como um módulo separado ao invés de ter fragmentos do software emaranhados ou espalhados.

### **2.3.10. Refatoração**

Segundo Fowler (2004) a refatoração é conseguir melhorar o projeto mesmo após ele ter sido construído, e consegue-se isto com passos simples como movendo um campo de uma classe para outra, tirando um pouco de código de um método para transforma-lo em outro método, descer ou subir uma parte de código pela hierarquia de classes e outros. Estes simples passos podem melhorar radicalmente todo o projeto e

o sistema como um todo. Portanto, para FOWLER “Refatoração é o processo de alteração de um sistema de software de modo que o comportamento externo do código não mude, mas que a estrutura interna seja melhorada.”.

#### **2.3.11. Conceitos de Projeto Orientado a Objeto**

Deboni (2003) afirma que as vantagens que ocorrem através do uso da orientação a objeto são entre outras, a facilidade de integração das fases durante o processo de desenvolvimento, favorecimento no teste do sistema devido ao encapsulamento, redução dos custos e prazos no desenvolvimento, possibilidade de abordar problemas mais complexos, além de facilitar descrição do comportamento do mundo real.

#### **2.3.12. Classes de Projeto**

Pressman (2011) assegura que as classes de projetos facilitam a criação de soluções e cada uma descreve algum elemento do domínio do problema, concentrando-se em seus aspectos.

### **3. MATERIAL E MÉTODOS**

O trabalho foi realizado de forma progressiva em 4 passos. No primeiro passo foi realizado um levantamento bibliográfico a fim de obter conhecimento referente aos conceitos consagrados de projetos existentes na engenharia de software e conhecer os processos de desenvolvimento, procurando obter informações dos quais mais se adequaria para o desenvolvimento de aplicações mobile.

No segundo passo foram selecionadas as tecnologias a serem usadas, elegendo o ambiente de programação Android Studio e a linguagem de programação Java para o desenvolvimento de aplicações para a plataforma Android, e o ambiente de programação Microsoft Visual Studio 2013 junto com a linguagem de programação C#, para o desenvolvimento de Windows Phone. Ambos utilizando a programação orientada a objeto.

No terceiro passo, os softwares foram desenvolvidos utilizando boas práticas e programação orientada a objetos, porém sem quaisquer utilizações dos conceitos de projeto. Em segundo instante, foram desenvolvidos novas versões dos softwares aplicando o conceito de projeto que mais se adequou a cada um.

No quarto e último passo, foram realizados os testes e comparativos nos softwares a fim de definir as diferenças entre os softwares da primeira e segunda versão.

## 4. DESENVOLVIMENTO

Foram desenvolvidos os seguintes aplicativos, o Diluição de Ácido Peracético, o Cálculo de Enfermagem e o EasyWayMenu. Tais aplicativos foram desenvolvidos para a plataforma Android e Windows Phone.

O aplicativo Diluição do Ácido Peracético possibilita o cálculo da diluição a partir do fornecimento do volume de ácido a preparar, a concentração em porcentagem e concentração do ácido comercial.

Já o aplicativo Calculo de Enfermagem auxiliam os enfermeiros e técnicos em enfermagem a realizarem o cálculo de gotejamento. Com o mesmo, pode se fazer também o cálculo de solvente e soluto.

O EasyWayMenu é uma solução para comércios no ramo de restaurante e lanchonete, de forma a informatizar os pedidos e entregas de seus produtos. Basicamente esta solução é composta de um WebSite, um WebService e Aplicativos Mobile.

O Website, em primeiro instante, permite o acesso do administrador do sistema, onde é possível ações administrativas, como cadastrar os produtos, acompanhar os pedidos, controlar o estoque e outros.

O WebService é responsável por acessar todo conteúdo administrado pelo WebSite e disponibilizar para os aplicativos moveis, e ainda receber e administrar tudo que provem dos aplicativos.

O aplicativo possibilita ao usuário/cliente cadastrar seus dados, visualizar todos os produtos disponíveis, comprar os produtos, visualizar todos pedidos realizados e visualizar os status dos pedidos.

### 4. Aplicação dos Conceitos

#### 4.1 Encapsulamento de Informações

No WebService, assim como no WebSite, a utilização deste conceito possibilitou que o método login se tornasse mais claro e efetivo.

```
EasyWayMenu_dbEntities EWEntity = new EasyWayMenu_dbEntities();

    [WebMethod]
    public bool Login(String login, String Senha)
    {
        try
        {
            return EWEntity.Pessoa.Any(p => p.Login.Contains(login) &&
p.Senha.Contains(Senha));
        }
    }
}
```



```
    }  
    catch (Exception)  
    {  
        return false;  
    }  
}
```

#### 4.2 Refatoração

No WebService, este conceito possibilitou a resolução de um problema de atomicidade na função InserirCarrinhodeCompra.

#### 4.3 Modularidade

O EasyWayMenu Website está dividido em basicamente 3 módulos, onde as páginas são a Interface, a BLL ou camada de negócio é onde ocorre as validações do sistema, EasyWayModelo.edmx ou modelo da entity framework é a camada que faz acesso ao banco de dados.

#### 4.4 Aplicação do Padrão Singleton

Na primeira versão do EasyWayMenu Android, este padrão foi utilizado para minimizar o número de instancias da classe conexão.

#### 4.5 Independência Funcional

Este conceito influenciou o uso da Entity Framework e cada resultado de Stored Procedure e cada tabela no banco se tornou uma classe encapsulada que é altamente coesa e pouco acoplada.

### 5. RESULTADO E DISCUSSÃO

Durante o desenvolvimento do projeto, a Microsoft liberou a atualização “update4” para o Visual Studio 2013, o que removeu o “service reference” utilizado para referenciar-se a um webservice em Windows Phone, assim não permitindo o acesso ao webservice utilizado. A solução encontrada foi a utilização de uma aplicação Windows Phone Silverlight 8.1 que permite a programação similar a versão 8.

No decorrer do desenvolvimento do webservice para o acesso a dados por parte do aplicativo EasyWayMenu para Android e Windows Phone, detectou-se um problema de comunicação das versões android superiores a 2.3.3 com o webservice WCF, utilizado para o aplicativo Windows Phone devido a sua adaptabilidade. A solução encontrada durante o trabalho foi a utilização de um webservice ASMX exclusivo para o android.

Com o uso do padrão Singleton, foi possível garantir que um objeto de uma classe não seja instanciado mais de uma vez, o que evita desperdício de memória e possibilita um baixo acoplamento.

Com o conceito de Refatoração, foi possível melhorar o conteúdo interno da função, resolvendo o problema de atomicidade e ainda deixando mais claro e objetivo o código.

Com a utilização do conceito de encapsulamento de informações, foi possível facilitar a manutenção do código, pois com o uso da orientação objeto diminuiu bastante o número de linhas de código.

A independência funcional possibilitou que classes fiquem menos acopladas, possibilitando que problemas não interrompam o sistema.

A utilização da modularidade permitiu uma melhor compreensão do sistema, pois dividiu o mesmo em módulos.

## **6. CONCLUSÃO**

Após o término deste trabalho, foi concluído que o sucesso na aplicação dos conceitos da engenharia de software se repetiu tanto no desenvolvimento web quanto no desenvolvimento mobile.

O uso dos conceitos trouxe consigo eficiência no sistema, eliminação de erros clássicos como atomicidade, facilidade de manutenção, código limpo, coeso e uma facilidade de entendimento do sistema como um todo, resultando em mais qualidade e melhor desempenho.

## **REFERÊNCIAS**

CYBIS, Walter de Abreu, Universidade Federal de Santa Catarina, Engenharia de Software – INE 5322. Disponível em: <[http://www.inf.ufsc.br/~cybis/ine5322/Aula5\\_%20Modularidade\\_e\\_reutilizacao.pdf](http://www.inf.ufsc.br/~cybis/ine5322/Aula5_%20Modularidade_e_reutilizacao.pdf)> Acesso em: 15 outubro 2014.

DEBONI, José E.Z. Modelagem orientada a objetos com a UML. São Paulo, Futura, 2003.

FABRÍZZIO, Alphonsus A. M. N. Soares, Universidade Federal de Goiás. Padrões de Projeto. Disponível em: <<http://www.inf.ufg.br/~fabrizio/web/java/aula6>> Acesso em: 15 outubro 2014.

FOWLER, Martin, Refatoração: aperfeiçoando o projeto de código existente. Porto Alegre. Bookman Editora, 2004.

GOULART, Reane Franco, Refinamento. Disponível em:  
<<http://profareane.files.wordpress.com/2013/03/aula-2-conceitos-de-projeto.pdf>>  
Acesso em: 17 outubro 2014.

LEITE, Julio Cesar Sampaio do Prado, Níveis de Abstração. 2007 Disponível em:  
<<http://jcspl.net/2007/03/14/niveis-de-abstracao/>>  
Acesso em: 17 de outubro 2014.

MAYNARD, Jeff. PROGRAMAÇÃO MODULAR – Modularidade. 1. Ed. Rio de Janeiro. Editora S.A, 1972.

PRESSMAN, Roger S. Engenharia de software – Uma Abordagem Profissional. 3. Ed. Porto Alegre. MAKRON Books do Brasil Editora Ltda, 1995.

PRESSMAN, Roger S. Engenharia de software – Uma Abordagem Profissional. 7. Ed. Porto Alegre. AMGH Editora Ltda, 2011.

SHAW, Mary, GARLAN, David; Software Architecture. Perspectives on an Emerging Discipline. Prentice Hall, 1996.

SOMMERVILLE, Ian. Engenharia de Software. 8 Ed. São Paulo. Pearson, 2007.

SONNINO, Bruno, Microsoft MVP (Most Valuable Professional). Programação Orientada a Aspectos. Disponível em: <<http://msdn.microsoft.com/pt-br/magazine/dn574804.aspx>> Acesso em: 12 outubro 2014.

SOUZA, Ana C.S; BARROSO, Graciele C; VINHAS, Hingridi M. Refatoração: Aperfeiçoando um código existente. 2012. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade José do Rosário Vellano, Alfenas.