**Michael Fernandes Rodrigues**

**ERRATA**

**(TCC I)**

**Análise Orientada a Objetos**

A análise orientada a objetos consiste na criação de objetos que representam o problema a ser resolvido, desta forma as classes se relacionam e interagem umas com as outras utilizando os métodos, atributos e os mecanismos de comunicação destes objetos. Para se fazer a análise orientada a objetos é necessário fazer uma descrição das características dos objetos utilizados para descrever o problema.

“A análise enfatiza uma investigação do problema e dos requisitos, em vez de uma solução, por exemplo, se desejamos um novo sistema online de comercialização, como ele será usado? Quais são as suas funções?”. Larman (2007,p34).

De acordo com Larman, a Analise é um termo de significado amplo melhor qualificado como analise de requisitos (uma investigação dos requisitos) analise orientada a objetos (uma investigação dos objetos do domínio).

Os princípios básicos da analise orientada a objetos são: A função é descrita, o comportamento é representado, o domínio da informação é modelado, os modelos de dados funcionais e comportamentais são separados a fim de expor maiores detalhes, os modelos iniciais representam a essência do problema, e os últimos modelos expõem detalhes da implementação.

Segundo Larman, O projeto enfatiza uma solução conceitual (em software ou hardware) que satisfaça os requisitos e não sua implementação. Uma descrição de um esquema de banco de dados e objetos de software é um bom exemplo. Idéias de projeto excluem freqüentemente detalhes de baixo nível ou “óbvios” – óbvios para os consumidores visados. Em ultima instancia, projetos podem ser implementados e a implementação (como por exemplo, o código) expressa o verdadeiro e completo projeto realizado.

**Projeto Orientado a Objetos**

O projeto orientado a objetos transforma o modelo criado na analise orientada a objetos em um modelo que se apresenta como um documento para a construção do projeto de software. O projeto orientado a objetos necessita de uma arquitetura de software com multicamadas, da especificação de subsistemas que exerçam funções críticas e que forneçam uma infra-estrutura de apoio ao projeto. A descrição dos objetos que formam os blocos do sistema, assim como, a descrição dos mecanismos de comunicação dos objetos que possibilitam a transição das informações entre as camadas também são essenciais para um projeto orientado a objetos.

A arquitetura Orientada a Objetos visa maximizar o reuso dos objetos isso com objetivo de aumentar a qualidade e diminuir o tempo de desenvolvimento do produto. Para isso os arquitetos necessitam da documentação gerada pelo projeto orientado a objetos estes artefatos são essências para a continuação do projeto.

Segundo Larman, o projeto orientado a objetos se preocupa com a definição de objetos de software e suas responsabilidades e colaborações. Uma notação comum para ilustrar essas colaborações e o diagrama de seqüência (uma espécie do diagrama de interação da UML). Ele mostra o fluxo de mensagens entre os objetos de software e, assim, a invocação de métodos.

Um projeto orientado a objetos é dividido em quatro partes, cada uma dessas camadas é responsável por áreas especificas do projeto.

A Camada de subsistemas é a camada responsável por conter a representação de cada um dos subsistemas, essa camada permite ao software responder aos requisitos definidos pelos clientes e implementar a infra-estrutura para o suporte técnico dos requisitos.

A Camada de classes e objetos esta camada tem a função de conter a representação de cada um dos objetos desta forma as hierarquias das classes são definidas nesta camada, com isso é possível a construir o software utilizando generalizações e especializações cada vez mais precisas.

A Camada de mensagens contém os detalhes do projeto orientado a objetos, estes detalhes possibilitam que cada objeto se comunique entre seus colaboradores. Sendo assim a camada de mensagens estabeleça as interfaces internas e externas do software.

Camada de responsabilidades é responsável por manter as estruturas de dados e os algoritmos dos atributos e métodos e todos os objetos do projeto orientado a objetos.

**Programação Orientada a Objetos**

De acordo com Milane de padronização para expressar instruções para um computador. Conjunto de regras semânticas e sintáticas usadas para definir um programa de computador. Ela permite que permita que o programador sobre quais dados o computador via atuar.

Segundo Milane a orientação a objeto, também é conhecida como POO (Programação Orientada a Objeto) é um modelo de desenvolvimento de software, que organiza as informações utilizadas como parte de informações maiores conhecidas como objetos.

O objetivo da programação orientada a objetos é melhor identificar os conjuntos de objetos responsáveis por detalhar um produto de software. Este princípio de programação consiste em relacionamentos e trocas de mensagens entre os objetos. Na programação orientada a objetos um conjunto de classes é implementado. Essas classes são responsáveis por definir os comportamentos dos objetos isso através de métodos e atributos.

**IRUP**

Segundo (Fowler, 2005 p39) e uma estrutura de processos fornecidos pela (Rational Unified Process) fornece um vocabulário e uma vaga estrutura sobre processos define o que e como fazer ele e basicamente uns processos interativos.

Assim com mencionado no site www.wthreex.com, O Rational Unified Process® (também chamado de processo RUP®) é um processo de engenharia de software.  Ele oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento.  Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.

O RUP é inicialmente dividido em quatro fases: iniciação, elaboração,construção e transição. E também e composto de nove disciplinas presentes em quase todas as fases citadas essas disciplinas são: modelagem de negócios, requisitos, analise e designer, implementação, teste, implantação, gerencia de configuração e mudança gerenciamento de projeto e por fim Ambiente.

**UML**

O UML é uma linguagem de modelagem unificada, ou seja, é uma linguagem visual para especificar, construir e documentar os artefatos de um sistema.

“O padrão UML indica que certos elementos normalmente são desenhados em determinados tipos de diagramas, mas isso não é uma regra.” Fowler (2005, p 33).

De acordo com Larman, a palavra visual na definição é um ponto chave – UML é a notação diagramática padrão, de fato, para desenhar ou apresentar figuras (com algum texto) relacionadas a software – principalmente software OO.

Segundo Larmam a UML define vários perfis UML que especificam subconjuntos da notação para áreas de assunto comum, tais como diagramação de Enterprise JavaBeans (com o perfil UML EJB).

“A UML nasceu da unificação de muitas linguagens gráficas de modelagem orientadas ao objeto que floresceram ao final dos anos oitenta, início dos noventa. Desde sua aparição em 1997, ela fez com que essa torre de Babel fosse resolvida...” Fowler (2005 p 25).

O diagrama de classe é responsável por descrever os tipos de objetos utilizados no sistema e os possíveis relacionamentos entre eles. O digrama de classe também tem a função de mostrar as propriedades e as operações de uma classe assim as suas restrições na maneira em que os objeto se comunicam.

Diagramas de seqüência são os diagramas responsáveis por informar o comportamento de um determinado cenário. Nesses diagramas também é mostrado diversos exemplos de mensagens que trafegam as informações entre os objetos de um caso de uso especifico.

A divisão de componentes reutilizáveis e suas dependências internas de um sistema são de responsabilidade dos diagramas de componentes.

Outro diagrama muito utilizado no processo de modelagem dos sistemas é o diagrama de atividades. Os diagramas de atividades são utilizados para esclarecer os fluxos de controle ou de atividades em operações complexas em um caso de uso. Geralmente os digramas de Atividades são confundidos com fluxogramas, apesar de serem semelhantes à diferença principal é a possibilidade que os diagramas de atividades têm de suportar fluxos paralelos.

Os diagramas de casos de uso são técnicas de auxilio na captura de requisitos funcionais de um sistema. Sua função é descrever as interações entre os usuários e o sistema.

**Mecanismos de segurança da Linguagem de Programação e SGDB**.

O termo captcha vêem do inglês “Completely Automated Public Turing test to tell Computers and Humans Apart” e pode ser traduzido como “público completamente automatizado para diferenciação entre computadores e humanos”. O captcha tem a função de impedir que pessoas utilizem ferramentas para acessar de forma abusiva paginas web ou interfaces de sistemas. Os captchas também são conhecidos com anti-robôs já que dependendo do nível de complexidade do captcha torna-se impossível ser interpretado automaticamente por um computador. O captcha é usado na maioria das vezes em telas de autenticação de sistemas corporativos, blogs, site de download entre outros. Tudo isso no intuito de restringir o acesso das informações somente a humanos.

A linguagem de programação PHP possui alguns métodos de criptografia entre eles os que mais se destacam são: o MD5, o SHA1 e o Base64.

MD5 é a sigla de Message-Digest Algorithm 5 foi desenvolvido em 1991 por Ronald Rivest, o MD5 é um algoritmo de hash de 128 bits. O MD5 é utilizado por software para proteger senhas, verificar a integridade de arquivos entre milhares de aplicações. Uma aplicação bem conhecida é o md5sum que consiste em criar uma string hash de um arquivo.

O SHA ou Algoritmo de Hash Seguro é uma família de algoritmos, o algoritmo mais usado desta família é o SHA1. O SHA1 é utilizado em uma gama de aplicações e protocolos de segurança. Os protocolos TLS, SSL e SSH são exemplos de protocolos que utilizam a criptografia do SHA1, mas mesmo sendo tão reconhecido o SHA1 possui algumas vulnerabilidades comprovadas da mesma forma que o MD5.

O base64 se comparado ao MD5 e ao SHA1 é o mais simples, este método de codificação de dados e voltado para a transferência de informações pela internet, o seu funcionamento é constituído de 64 caracteres que são “de A até Z”, “de a até z” e “de 0 a 9” e o mesmo pode ser facilmente descriptografado.

O recurso register\_globals do PHP oferece grande risco a integridade do sistema. O register\_globals possibilita a criação de variáveis do PHP a partir do envio da requisição ao servidor que hospeda a aplicação, por exemplo:

Acessando a “http://dominio.com.br/index.php?function=kill” com o register\_globals habilitado, automaticamente o PHP irá instanciar uma nova variável com o nome de “function” contendo o valor “kill”.

Este Recurso do PHP já considerado obsoleto já que não través benefício algum pelo contrario representa um grande risco de segurança.

O SQL Injection é uma técnica de invasão de sistemas geralmente aplicada contra sistemas web, mas pode ser utilizada em qualquer linguagem de programação. O SQL Injection ocorre quando o atacando malicioso consegue inserir um fragmento de SQL ou ate mesmo uma instrução inteira dentro de uma consulta isso através das entradas de dados vulneráveis no sistema.