UNIVERSIDADE FEDERAL DE SERGIPE

DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RELATÓRIO DO ESTÁGIO

SUPERVISIONADO

JANISSON GOIS DE SOUZA

ITABAIANA – SE

JUNHO - 2011

RELATÓRIO DO ESTÁGIO

SUPERVISIONADO

JANISSON GOIS DE SOUZA

Relatório de estágio apresentado ao Curso de Sistemas de Informação como parte das exigências da disciplina Estágio Supervisionado, sob a orientação pedagógica do Prof. Marcos Barbosa Dósea e orientação técnica da Analista do Centro de Processamento de Dados da Universidade Federal de Sergipe Fernanda Almeida Passos.

ITABAIANA – SE

JUNHO – 2011

UNIVERSIDADE FEDERAL DE SERGIPE

DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RELATÓRIO DO ESTÁGIO

SUPERVISIONADO

JANISSON GOIS DE SOUZA

Aprovado em: \_\_/\_\_/\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Me. Prof. Marcos Barbosa Dosea

Orientador Pedagógico

RESUMO

Os Sistemas de Informação (SI) têm grande importância nas organizações, pois através deles é que facilitamos o gerenciamento das informações necessárias para a execução de um processo de forma mais eficiente. Apesar de sua eficiência, nos deparamos com um fator crucial em grandes e até mesmo em médias organizações, a desintegração desses sistemas. Cada departamento tem seu próprio SI e realiza seu trabalho de forma isolada gerando inconsistências de dados.

Com a Universidade Federal de Sergipe (UFS) não é diferente, uma grande organização e com departamentos cada qual com seu próprio sistema. Para amenizar esse problema, em 2010, a UFS e a Universidade Federal do Rio Grande do Norte (UFRN) firmaram um acordo de cooperação interinstitucional onde UFRN cedeu os seus Sistemas Integrados para que sejam implantados na UFS.

Desde então o Centro de Processamento de Dados (CPD) da UFS vem empenhado em realizar essa implantação, mas os Sistemas da UFRN não são totalmente apropriados à realidade da UFS e, portanto, requerem ajustes bem como a migração dos dados dos sistemas legados para os sistemas novos.

Esse relatório tem como objetivo discutir o trabalho realizado para a implementação de novas funcionalidades, migração de dados e implantação dos módulos Pesquisa e Produção Intelectual do Sistema Integrado de Gestão das Atividades Acadêmicas (SIGAA).

**Palavras-Chave:** Sistemas Integrados de Informação; Implantação; Migração de dados; Sistemas Legados.

SUMÁRIO

[1. INTRODUÇÃO 6](#_Toc295161219)

[1.1. Local do estágio 6](#_Toc295161220)

[1.2. Objetivos do estágio 6](#_Toc295161221)

[1.3. Estrutura do relatório 6](#_Toc295161222)

[2. REFERENCIAL TEÓRICO 7](#_Toc295161223)

[2.1. SCRUM. 7](#_Toc295161224)

[2.1.1. Características. 7](#_Toc295161225)

[2.2. Rational Unified Process – RUP 9](#_Toc295161226)

[2.2.1. Estrutura Estática 9](#_Toc295161227)

[2.2.2. Estrutura Dinâmica 11](#_Toc295161228)

[2.3. Processo de Migração de dados. 11](#_Toc295161229)

[2.4. Arquitetura dos Sistemas da UFRN 13](#_Toc295161230)

[3. FERRAMENTAS UTILIZADAS 15](#_Toc295161231)

[3.1. JBOSS 15](#_Toc295161232)

[3.2. JSF e JPA 15](#_Toc295161233)

[3.3. Hibernate 15](#_Toc295161234)

[3.4. Postgres 15](#_Toc295161235)

[3.5. Eclipse 15](#_Toc295161236)

[3.6. Talend 15](#_Toc295161237)

[4. ATIVIDADES DO ESTÁGIO 16](#_Toc295161238)

[4.1. Migração de Dados 16](#_Toc295161239)

[4.2. Desenvolvimento 16](#_Toc295161240)

[4.3. Implantação 16](#_Toc295161241)

[5. CONCLUSÃO 17](#_Toc295161242)

[6. REFERÊNCIAS BIBLIOGRÁFICAS 18](#_Toc295161243)

1. INTRODUÇÃO

Este relatório apresenta a vivência do estagiário em um ambiente de produção real onde serão conciliados, de forma pratica, o conhecimento aprendido durante o curso.

* 1. Local do estágio

A realização do estagio foi concebida no Centro de Processamento de Dados (CPD) do Campus Professor Alberto Carvalho da Universidade Federal de Sergipe (UFS), no qual foi disponibilizada toda a infraestrutura necessária para tal atividade. Utilizando a disponibilidade de 20 horas semanais durante o período de 04 de Abril de 2011 a 05 de Agosto de 2011.

* 1. Objetivos do estágio

O objetivo geral do estágio é adequação e implantação dos módulos ‘Produção Intelectual’ e ‘Pesquisa’ do SIGAA seguindo as regras de negócio da instituição.

Objetivos específicos do estagio:

* Preparar ambiente de desenvolvimento;
* Estudo da arquitetura do sistema SIGAA;
* Estudo dos módulos ‘Produção Intelectual’ e ‘Pesquisa’ do SIGAA;
* Estudo das ferramentas utilizadas;
* Migração de dados do sistema legado;
* Implementação de correções e novos requisitos;
* Realização de testes nas correções e nos novos requisitos implementados.
  1. Estrutura do relatório

Esse relatório está dividido da seguinte forma: na Seção 2 é abordado o conhecimento teórico necessário para a atuação do estagiário, na Seção 3 são apresentadas as ferramentas necessárias para a execução do trabalho a ser realizado, a seguir, na Seção 4, são listadas as atividades realizadas, bem como as dificuldades enfrentadas e as soluções encontradas, e por fim, na Seção 5, temos uma breve conclusão sobre o estágio realizado.

1. REFERENCIAL TEÓRICO

Para a realização do trabalho foi necessário o estudo de processos de desenvolvimento de software e de processo de migração de dados. Para o desenvolvimento de software o CPD já possui um processo definido baseado no framework SCRUM. Já para a migração de dados o processo é baseado nas especificações de migração da disciplina de análise e design do Rational Unified Process (RUP) [1], a seguir descreveremos melhor cada um deles.

* 1. SCRUM.

SCRUM é um processo de desenvolvimento ágil, iterativo e incremental de softwares que inicialmente foi proposto como um estilo de gerência de projetos e ganhou atenção pelos bons resultados obtidos em suas equipes. Após essas observações dos resultados, Jeff Sutherland, em 1993, começou a moldar um framework de processo que mais tarde, em 1995, sua documentação foi formalizada por Ken Schwaber.

Apesar de ser bem difundido para desenvolvimento de software, o SCRUM também pode ser aplicado a qualquer projeto que priorize o trabalho em equipe e a entrega do produto em um curto espaço de tempo agregando assim um maior valor ao negócio do cliente o mais breve possível.

* + 1. Características.

Inicialmente é especificado o *Product Backlog* que é uma lista de todas as funcionalidades a serem implementadas no produto, em seguida é realizado o *Sprint Planning Meeting* onde são separadas algumas dessas funcionalidades, através de prioridades definidas pelo *Product Owner (*papel atribuído ao principal interessado no produto) juntamente com equipe, para formar o *Sprint Backlog* e dar inicio um ciclo de desenvolvimento conhecido como *Sprint*.Esse ciclo dura em torno de 2 a 4 semanas (*weeks*) dependendo da forma como foi definido na empresa.

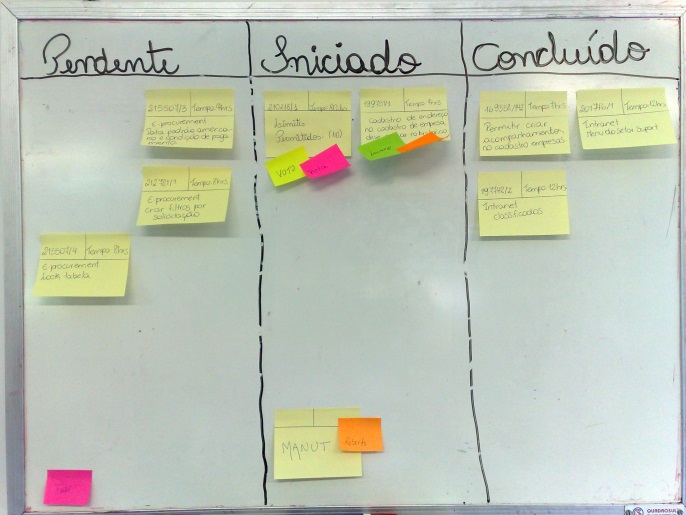
Durante o Sprint o *Scrum Master* (papel atribuído ao gerente da equipe) realiza, juntamente com a equipe, a *Daily Scrum Meeting* que são reuniões diárias rápidas realizadas de pé para o acompanhamento de cada membro da equipe onde serão discutidas e priorizadas as tarefas a serem realizadas naquele dia e também discutir possíveis problemas ou dificuldades, assim todos os membros poderão contribuir para uma possível solução dessas dificuldades e ataca-las o mais cedo possível.

No fim de cada *Sprint* além da entrega são feitas as *Sprint Review Meeting* para apresentar as funcionalidades desenvolvidas durante aquele *Sprint* e também é feita uma *Sprint Retrospective* onde é avaliado o andamento do *Sprint* e identificado o que foi bom que pode ser melhorado para o próximo *Sprint*. Feito isso se inicia todo o ciclo novamente a partir da definição do *Sprint Backlog* como mostrado na figura 1.

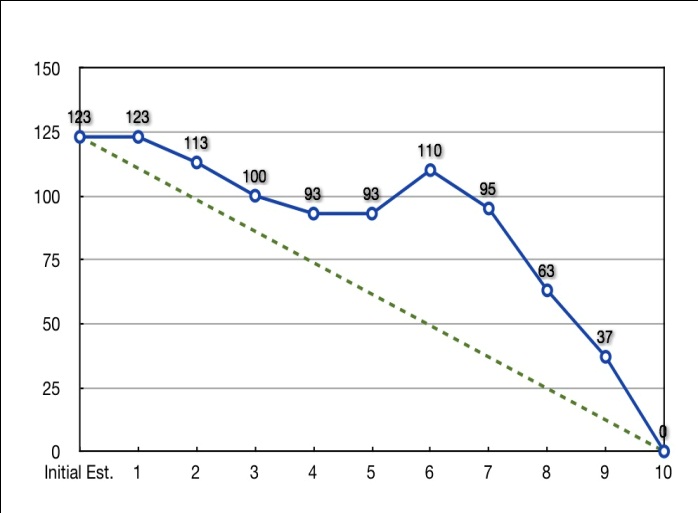


**Figura 1:** Demonstração gráfica do processo SCRUM.

Para realizar o acompanhamento do *Sprint* são utilizados o *Scum Board* que é o quadro onde são registrados o andamento das tarefas, como mostrado na Figura 2, e o *Burndown Chart* que é um gráfico que representa quantos pontos ainda faltam até o fim do Sprint.



**Figura 2:** Exemplo de um Scrum Board

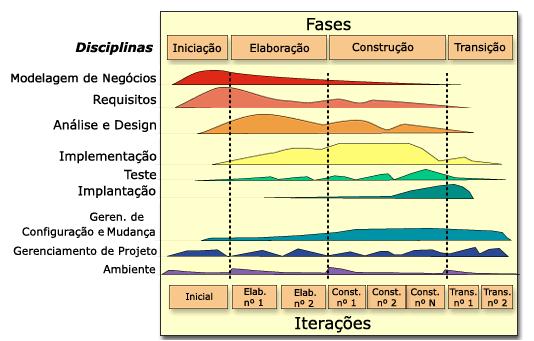


**Figura 3:** Exemplo de um Burndown Chart

* 1. Rational Unified Process – RUP

O RUP é um processo de engenharia de software que oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. É um processo iterativo incremental, baseado em componentes que realizam interfaces os quais podem intercomunicar entre si, dirigido por casos de usos e centrado na arquitetura, pois a mesma é projetada e desenvolvida em paralelo ao desenvolvimento dos casos de uso.

O RUP possui duas dimensões, uma é a estrutura estática que é representada pelas nove disciplinas que agrupam as atividades de maneira lógica e a outra uma estrutura dinâmica composta de quatro fases que representa o tempo de execução e mostra os aspectos do ciclo de vida do processo à medida que se desenvolve, como mostrado na Figura 4.



**Figura 4:** Gráfico do esforço realizado por cada disciplina durante cada fase do RUP

* + 1. Estrutura Estática

A estrutura estática é composta por nove disciplinas que utilizam quatro elementos primários de modelagem, são eles:

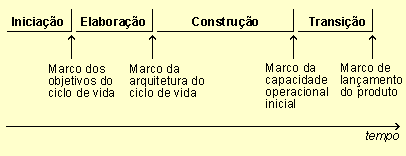
* **Papéis:** Descrevem o comportamento e as responsabilidades de um indivíduo ou grupo de indivíduos de uma equipe.
* **Atividades:** São ações realizadas pelos papéis que atualizam ou geram artefatos.
* **Artefatos:** São informações produzidas, modificadas ou utilizadas ao longo do desenvolvimento de software e são utilizados como entradas para as atividades que produzirão outros como saída.
* **Workflows:** Define a sequência de atividades que produzem resultados observáveis na forma de artefatos.

As disciplinas descrevem todas as atividades que devem ser realizadas, por quais papéis, seguindo um workflow para produzir um determinado conjunto de artefatos. São elas:

* **Modelagem de negócios:** Analisar e entender a estrutura e a dinâmica da organização onde o sistema será implantado para Identificar as possibilidades de melhoria e derivar os requisitos de sistema necessários.
* **Requisitos:** Delimitar o sistema estabelecendo uma concordância com os clientes sobre o que se deve fazer, oferecer aos desenvolvedores do sistema uma compreensão melhor dos requisitos e possibilitar uma base para uma estimativa de custo e tempo de desenvolvimento.
* **Análise & Design:** Transformar os requisitos em um design para desenvolver uma arquitetura do sistema.
* **Implementação:** Definir a organização do código, implementa classes e objetos, realiza testes unitários e integra os resultados produzidos ao sistema executável.
* **Teste:** Localizar e documentar defeitos na qualidade do software e validar o design e as funções do software conforme foram projetadas.
* **Implantação:** Testar o produto em um ambiente mais próximo do real, antes de ser finalmente entregue ao cliente.
* **Gerenciamento de configurações e mudanças:** Descreve os métodos, processos e ferramentas utilizados para o gerenciamento de configuração e mudança durante o processo de desenvolvimento de software.
* **Gerenciamento de Projetos:** Descreve apenas algumas práticas da gerencia de projetos, como por exemplo, gerenciamento de riscos, planejamento do projeto iterativo e monitoramento do progresso de um projeto através de métricas.
* **Ambiente:** Descreve as diretrizes para o suporte do projeto em relação ao ambiente (processos e ferramentas) para a execução do trabalho.
  + 1. Estrutura Dinâmica

Composta por quatro fases, a estrutura dinâmica possibilita um melhor gerenciamento durante os ciclos de desenvolvimento, pois cada fase possui um marco bem definido com entregas de artefatos que servem de entradas para a próxima fase. As fases são:

* **Iniciação:** é a fase responsável por Formular o escopo do projeto, planejar e preparar um caso de negócio, sintetizar uma possível arquitetura e preparar o ambiente para o projeto.
* **Elaboração:** é a fase onde são definidas e validadas as baselines da arquitetura, refinada a visão do projeto, criado os planos de iteração, a refinação da arquitetura e a seleção dos componentes a serem utilizados.
* **Construção:** é a fase onde são gerenciados os recursos, otimizados os controles e processos, desenvolvido os componentes e os testes dos critérios de avaliação e avaliado os releases do produto de acordo com os critérios estabelecidos no documento de visão.
* **Transição:** é a fase responsável por executar os planos de implantação, finalizar o material de suporte, testar o produto, criar uma release do produto, obter feedback dos usuários, ajustar o produto e disponibilizar o produto para os usuários finais.



**Figura 5:** Gráfico dos marcos de cada fase do RUP

* 1. Processo de Migração de dados.

A migração de dados aparentemente consiste em uma tarefa simples de cópias dos dados de um banco para outro, mas na verdade não é só isso. Ela consiste em um processo que envolve algumas etapas de extrema importância.

As etapas básicas [1] são definir o escopo da migração, entender os dados de origem e traçar um perfil de dados, definir o mapeamento entre as origens de dados e o novo banco de dados e por fim identificar a migração automática e a migração manual. Em seguida são descritas mais detalhadamente essas etapas.

A etapa de ‘Definir o escopo da migração’ consiste em identificar as origens dos dados, seus locais, os sistemas que o utilizam, os dados que não são mantidos na base de origem e como inseri-los no novo sistema e se os dados históricos realmente devem ir para a base nova.

‘Entender os dados de Origem e traçar um perfil de dados’ é a etapa onde devemos coletar as informações através de metadados que descrevam as origens dos dados, logo após analisa-se as origens dos dados montando um mapa das origens dos dados e por fim determina o perfil dos atributos da origem de dados, onde será montada a sua semântica.

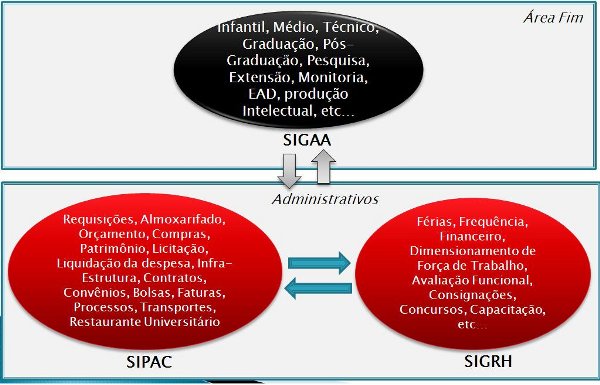
‘Definir o mapeamento entre as origens de dados e o novo banco de dados’, nessa etapa faz-se a correlação entre as origens de dados e o modelo físico do novo banco. Uma concepção errada é fazer essa correlação com o modelo lógico já que as características físicas são de extrema importância para o sucesso dessa migração.

A última etapa da migração é ‘Identificar a migração automática e a migração manual’ no qual são analisados os perfis de dados e os mapeamentos a fim de classificar quais podem ser executados por procedimentos automáticos e quais devem ser executados manualmente.

Durante a execução das etapas de migração de dados podemos nos deparar com vários problemas, os mais comuns são:

* **Bases diferentes:** Em um processo de migração podemos ter diferentes Sistemas de Gerenciamento de Banco de Dados (SGBD), bem como outras formas de armazenamento como, por exemplo, arquivos texto, planilha, DBF;
* **Base desnormalizada:** Sistemas Legados normalmente são mal estruturados e concebidos e, portanto, pode haver várias ocorrências de “falhas” como, por exemplo, mistura de assuntos e redundância desnecessária.
* **Duplicidades de dados**: quando não há validação de dados pela aplicação podem ocorrer de dados idênticos serem inseridos, causando assim inconsistência.
* **Tipos de dados diferentes**: Fator muito comum entre bases diferentes são tipos diferentes, onde são necessárias várias conversões para que os dados mantenham seu valor original;
* **Sistemas legados sem documentação**: Um dos principais causadores de atrasos nas migrações é a falta de documentação, pois, muitos sistemas tem que usar engenharia reversa para poder descobrir quais as correspondências corretas.
* **Tempo de transferência e organização de dados:** todo processo de migração requer tempo e disponibilidade dos servidores, portanto o mais indicado é a realização do mesmo em horário não comercial para causar menos impacto à organização.
  1. Arquitetura dos Sistemas da UFRN

Antes de iniciarmos na arquitetura propriamente dita, vamos mostrar os sistemas institucionais da UFRN e como eles interagem entre si. São eles: o Sistema Integrado de Gestão das Atividades Acadêmicas (SIGAA) que é o sistema responsável por toda a parte acadêmica que é a área fim da universidade, o Sistema Integrado de Patrimônio, Administração e Contatos que é o sistema responsável pelas atividades básicas administrativas, como por exemplo, solicitação de material, orçamentos, contratos, etc. e o Sistema Integrado de Gestão de Recursos Humanos que é o sistema responsável pelo controle das informações dos recursos humanos, como por exemplo, férias, concursos, financeiro, capacitação, etc.



**Figura 6:** Área de abrangência dos sistemas institucionais da UFRN

A arquitetura utiliza uma abordagem em camadas com o intuito de separar as responsabilidades dos componentes do sistema. A disposição das camadas é feita através de quatro camadas, sendo elas: Apresentação, Aplicação, Domínio/Negócio e Infraestrutura/Acesso a dados.

As camadas são organizadas em formato de pilha, como mostrado na Figura 6, onde uma camada inferior não pode ser dependente da camada superior limitando-se a apenas às chamadas de métodos e somente as camadas adjuntas podem se comunicam entre si.

**Figura 7:** Disposição das camadas da arquitetura.

A camada de apresentação é responsável pela interface onde serão exibidas as informações para o usuário final e é feita a manipulação as requisições.

A camada de aplicação

A camada de negócio

A camada de persistência (Acesso a dados) é o meio pelo qual a aplicação interage diretamente com o banco de dados. A modelagem dessa camada se deu através do padrão *Data Access Object* (DAO) possibilitando o mapeamento objeto relacional. Aqui também podemos encontrar um pool de conexões, que é gerenciado pelo servidor de aplicação.

1. FERRAMENTAS UTILIZADAS

Para a realização do estágio foi necessária à utilização de ferramentas de terceiros que nos auxiliaram no desenvolvimento e na migração de dados, são elas: JBOSS [3], JSF [4], JPA [5], Hibernate [6], Postgres [7], Eclipse [8] e Talend [9]. A seguir serão mostradas algumas características de cada uma delas.

* 1. JBOSS
  2. JSF e JPA
  3. Hibernate

É um framework para mapeamento objeto-relacional, ou seja, através dele conseguimos um nível de abstração maior em relação ao banco de dados relacional e a aplicação Orientada a Objetos (OO), tornando-a menos dependente a um Sistema de Gerenciamento de Banco de Dados (SGBD) específico, diminuindo a complexidade e facilitando o desenvolvimento de forma OO.

Além de executar consultas utilizando SQL (Structured Query Language) comum, o Hibernate também possui uma linguagem específica, baseada no SQL, totalmente orientada a objetos chamada HQL (Hibernate Query Language), onde as consultas podem ser montadas utilizando os objetos de domínio.

* 1. Postgres
  2. Eclipse
  3. Talend

1. ATIVIDADES DO ESTÁGIO

Começa aqui...

* 1. Migração de Dados
  2. Desenvolvimento
  3. Implantação

1. CONCLUSÃO

Começa aqui...

1. REFERÊNCIAS BIBLIOGRÁFICAS

x

|  |  |
| --- | --- |
| 1. | PROCESSO de migração de dados baseado na disciplina de Análise e Design do RUP, 2011. Disponivel em: <http://rjserver23.brq.com/rupport/rup\_legacy\_evol\_plugin/tasks/design\_database\_migration,%7BAFE73E86-AD34-4A5C-8DF3-FCD28DBE2F22%7D.html>. Acesso em: 02 jun. 2011. |
| 2. | JBOSS - Web Server. Disponivel em: <http://www.jboss.org/>. Acesso em: 20 maio 2011. |
| 3. | JSF - Java Server Faces Technology. Disponivel em: <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>. Acesso em: 12 maio 2011. |
| 4. | JPA - Java Persistent API Technology. Disponivel em: <http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>. Acesso em: 14 maio 2011. |
| 5. | HIBERNATE - Relational Persistence for Java. Disponivel em: <http://www.hibernate.org/>. Acesso em: 28 abr. 2011. |
| 6. | POSTGRES Database. Disponivel em: <http://www.postgresql.org.br/>. Acesso em: 12 maio 2011. |
| 7. | ECLIPSE - IDE. Disponivel em: <http://www.eclipse.org/org/>. Acesso em: 22 maio 2011. |
| 8. | TALEND Open Studio. Disponivel em: <http://www.talendforge.org>. Acesso em: 1 Junho 2010. |
| 9. | MENDES, A. **Arquitetura de Software:** desenvolvimento orientado para arquitetura. Rio de Janeiro: Campus, 2002. |

x