



SGB_ProjetoDeSoftware_(Design)

Versão 0.1

Histórico de Revisões

Nome	Alterações	Data	Versão
Gabriel da Silva Vieira	Criação do Documento	31/10/2012	0.1
Luã Silvério e Silva Rossi	Revisão, formatação do documento	14/11/2012	0.2

Sumário

1	Introdução	3
1.1	Objetivo	3
1.2	Declaração do Escopo	3
1.3	Contextualização do Software	4
1.4	Principais Restrições	4
2	Esquema de Iterações	4
3	Ferramentas e Notação	5
4	Levantamento de Requisitos	5
4.1	Papel – Especificador de Requisitos	5
4.2	Atividades	6
4.3	Artefatos	6
5	Design dos Dados	6
5.1	Papel – Designer de Banco de Dados	6
5.2	Atividade	6
5.3	Artefato	6
6	Projeto Arquitetural e de Componentes	6
6.1	Papel – Designer	6
6.2	Atividades	7
6.3	Artefatos	7
7	Design de Interfaces de Usuários	7
7.1	Papel - Designer de Interface de Usuário	7
7.2	Atividades	8
7.3	Artefatos	8
8	Teste do Software	8
8.1	Papel - Designer de Teste	8
8.2	Atividades	9
8.3	Artefatos	9

1 Introdução

Este documento apresenta o modelo de processo de desenvolvimento de software para o projeto intitulado “Fábrica de Software do INF – segundo semestre de 2012”. Como principal referência na definição deste documento destaca-se o *RUP (Rational Unified Process)*, que para este projeto foi customizado. Este documento atua na parte específica da Análise e Design do software. Na parte introdutória deste documento apresenta-se o objetivo do projeto, a declaração de escopo do projeto, uma contextualização do sistema a ser desenvolvido e as principais restrições. Na parte seguinte, o esquema de iterações é apresentado. Por fim, apresenta-se os papéis, atividades e artefatos da Análise e Design do Projeto.

Este documento atuará na orientação das etapas de desenvolvimento do sistema proposto através do desenho do banco de dados, das interfaces e da arquitetura do software, e da implementação do sistema e testes. Adota-se aqui o modelo híbrido (iterativo e incremental) para o desenvolvimento do sistema.

1.1 Objetivo

O objetivo do projeto é desenvolver o Sistema de Gestão Bibliográfica (SGB) para auxiliar a Direção e o Corpo Docente do INF a gerenciar as referências bibliográficas adotadas nos cursos do INF. O sistema deverá integrar informações bibliográficas contidas nos planos de ensino preenchidos pelos professores a partir do software SIPA, e nas informações de bibliografias existentes na Biblioteca Central (BC) da UFG a partir do software SOPHIA.

1.2 Declaração do Escopo

O Sistema de Gestão Bibliográfica (SGB) oferecerá serviços à Direção e o Corpo Docente do INF. Demais unidades acadêmicas não serão contempladas com esse projeto, mas isso não impede que futuramente o sistema possa ser utilizada por outras unidades acadêmicas.

A ideia básica é que os livros que constam nos planos de ensino dos professores devem existir em quantidade específica na BC. Os livros que não constam em quantidade suficiente na BC são utilizados para constituir uma lista para aquisição por parte da UFG. É esperado que o SGB faça a cotação via web para determinar o preço médio de cada livro. Além disso, o sistema deve sugerir a quantidade de exemplares que precisa ser adquirida para atender às normas do Ministério da Educação.

Adota-se neste projeto o seguinte mecanismo de classificação dos requisitos do sistema:

1. **Requisitos Essenciais:** certeza que podem ser completados.
2. **Requisitos Desejáveis:** espera-se que sejam completados, mas não é garantido.
3. **Requisitos Futuros:** há serias dúvidas sobre a interpretação e/ou implementação desses requisitos.

O ciclo de vida do projeto será dividido em quatro fases sequenciais: concepção, planejamento, construção e encerramento. Em especial, a fase de construção será iterativa, com iterações de duas semanas de duração, sendo que, ao final de cada

interação haverá um marco de projeto. O processo executado será aderente ao nível F do MPS.BR.

Este projeto esforça-se para equilibrar os desejos do cliente com a realidade do tempo para o desenvolvimento do sistema.

1.3 Contextualização do Software

O Sistema de Gestão Bibliográfica (SGB) atuará como meio no processo de levantamento e aquisição de livros através da cotação de exemplares e sugestão de compra. Atenderá os cursos de nível de graduação oferecidos pelo Instituto de Informática (INF/UFG) provendo os serviços necessários aos usuários do sistema, com base nas resoluções da UFG e na legislação vigente. Atuarão de forma direta no uso do sistema:

1. **Coordenadores de Cursos;**
2. **Professores de Disciplinas** através do registro de bibliografias sugeridas.

O sistema irá considerar as informações do **Plano de Ensino** registradas no SIPA (sistema já em uso por coordenadorias de cursos da UFG). Irá **exportar planilhas** para os formatos suportados pelo sistema da Biblioteca Central/UFG, o **software SOPHIA**, e pelo sistema de aquisição da UFG, o **software SOLICITE**. Irá **constituir** uma **lista de livros** que precisam ser adquiridos pela UFG. Irá **sugerir** a quantidade de livros que precisa ser adquirida em função da quantidade exigida pelo MEC e quantidade existente na BC. Irá **preencher o preço médio** do exemplar a partir de pesquisa de preços realizada na web. Irá **manter registro** das cotações realizadas ao longo do tempo. Irá fazer a **conversão** para reais de livros cotados em dólar. Irá **considerar** a configuração de **valor de frete** na relação de exemplares. Irá **sugerir** uma lista com livros e quantidades de modo a maximizar o atendimento das necessidades de aquisição. E por fim, irá **manter registro** das listas para aquisição construídas ao longo do tempo.

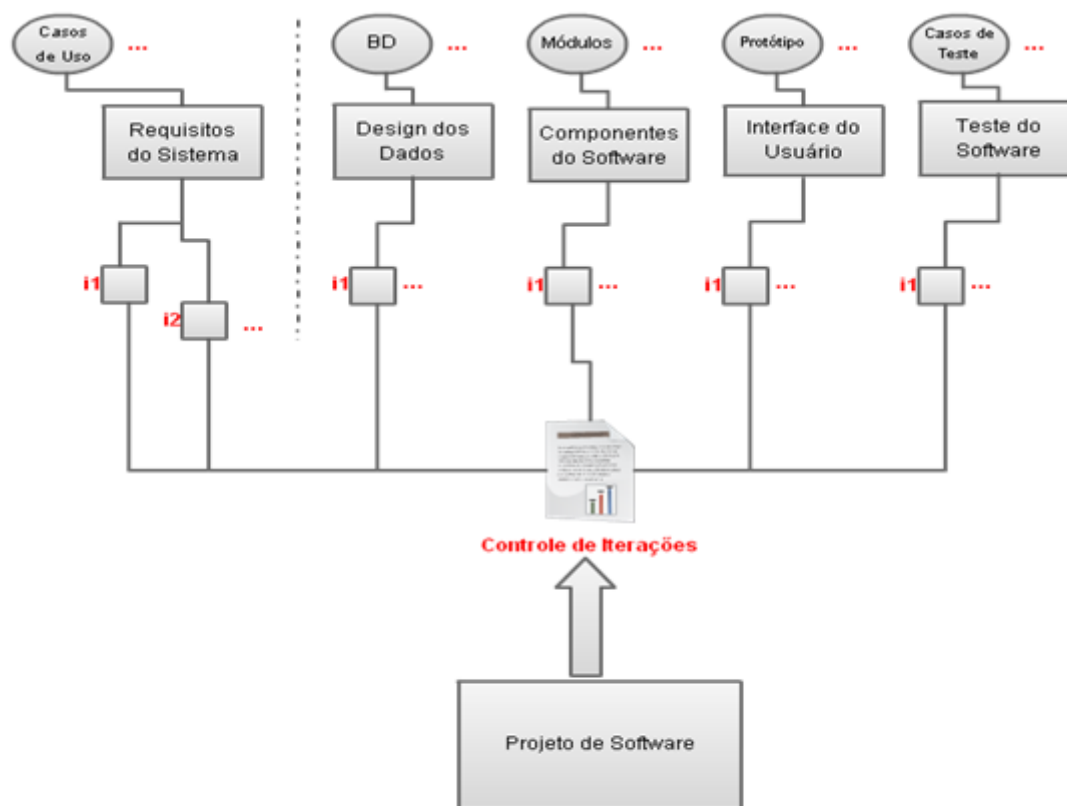
1.4 Principais Restrições

1. Interoperabilidade com outros sistemas – SIPA, SOPHIA, SOLICITE, *e-commerce*.
2. Disponibilidade do Fornecedor de Requisitos.

2 Esquema de Iterações

Adota-se o modelo iterativo híbrido com referência ao RUP (*Rational Unified Process*). A abordagem definida toma como princípio a *priorização do software*. Neste contexto, a priorização do software consiste em modelar e implementar primeiramente as partes mais complexas do sistema. Para atender o modelo iterativo cada etapa da Análise e Design irá priorizar os componentes definidos pela equipe de gestão. O conjunto das partes de cada etapa (Levantamento de Requisitos, Design dos Dados, Arquitetura e Componentes do Software, Interface do Usuário, e Teste de software) constituirão os elementos das iterações. O acompanhamento das iterações será feito por meio de *baselines* e *releases*.

A figura abaixo ilustra o esquema de iterações.



[online diagramming & design] createely.com

Figura 1: Esquema de Iterações

3 Ferramentas e Notação

Astah (<http://astah.net/>) - para a criação de modelos.

UML (Unified Modeling Language) - notação a ser utilizada.

4 Levantamento de Requisitos

4.1 Papel – Especificador de Requisitos

O papel especificador de requisitos detalha a especificação de uma parte da funcionalidade do sistema, descrevendo o aspecto Requisitos de um ou de vários casos de uso e outros requisitos de software de apoio. O especificador de requisitos também pode ser responsável por um pacote de casos de uso e por manter a integridade desse pacote. É recomendável que o especificador de requisitos responsável por um pacote de casos de uso também seja responsável pelos casos de uso e atores contidos no pacote.

Figura 2: Especificador de Requisitos

4.2 Atividades

1. Detalhar casos de uso.
2. Detalhar os requisitos do software.

4.3 Artefatos

1. Casos de uso.
2. Pacotes de Casos de Uso.
3. Especificação de requisitos de software.

5 Design dos Dados

5.1 Papel – Designer de Banco de Dados

O papel designer de banco de dados define tabelas, índices, visões, restrições, triggers, procedimentos armazenados, parâmetros de armazenamento ou tablespaces e outras construções específicas de um banco de dados necessárias para armazenar, recuperar e excluir objetos persistentes. Essas informações são mantidas no Artefato: Modelo de Dados.

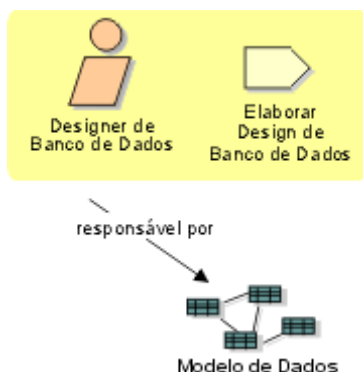


Figura 3: Designer de Banco de Dados

5.2 Atividade

- Elaborar Design de Bando de Dados

5.3 Artefato

- Modelo de Dados

6 Projeto Arquitetural e de Componentes

6.1 Papel – Designer

O papel designer define as responsabilidades, as operações, os atributos e os relacionamentos de uma ou de várias classes e determina como eles serão ajustados

para o ambiente de implementação. Além disso, o papel designer pode ser responsável por um ou mais pacotes de design ou subsistemas de design, incluindo todas as classes pertencentes aos pacotes ou subsistemas.

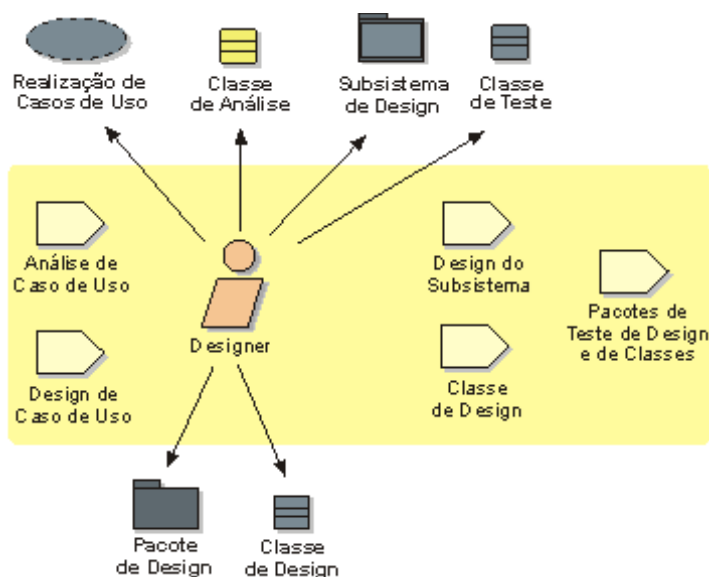


Figura 4: Designer

6.2 Atividades

1. Análise de casos de uso.
2. Design de casos de uso.
3. Design do subsistema.
4. Classe de Design.
5. Pacotes de Teste de Design e de Classes.

6.3 Artefatos

1. Subsistema de design.
2. Classe de teste.
3. Pacote de Design.
4. Classe de Design.

7 Design de Interfaces de Usuários

7.1 Papel - Designer de Interface de Usuário

O designer de interface de usuário lidera e coordena a construção do protótipo e o design da interface do usuário da seguinte forma:

- capturando os requisitos da interface do usuário, incluindo requisitos de usabilidade
- construindo protótipos de interface do usuário
- incluindo outros envolvidos da interface de usuário, como usuários, nas revisões de usabilidade e nas sessões de teste de uso
- revisando e fornecendo o feedback apropriado sobre a implementação final da interface do usuário, se criada por outros desenvolvedores, ou seja, designers e implementadores.

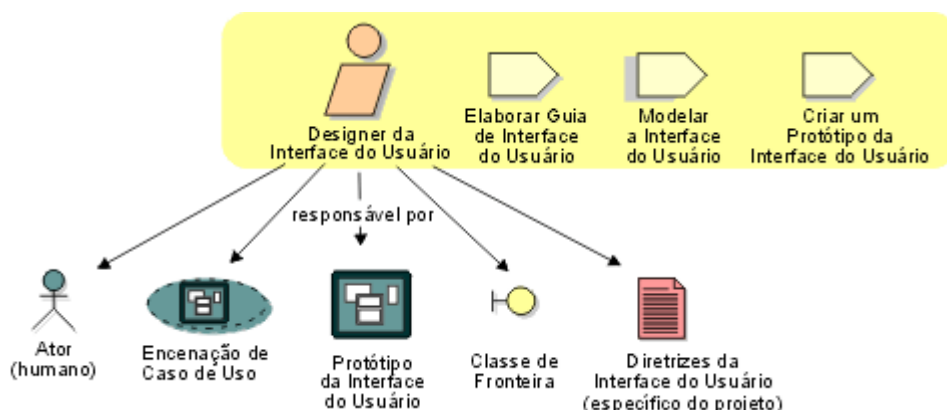


Figura 5: Designer de Interface de Usuário

7.2 Atividades

1. Elaborar guia de interface do usuário.
2. Modelar a interface do usuário.
3. Criar protótipos da interface do usuário.

7.3 Artefatos

1. Encenação de casos de uso.
2. Protótipo da interface do usuário.
3. Diretrizes da interface do usuário.

8 Teste do Software

8.1 Papel - Designer de Teste

O papel Designer de Teste é responsável por definir a abordagem de teste e assegurar sua correta implementação. O papel envolve identificar as técnicas, ferramentas e diretrizes apropriadas para implementar os testes necessários e dar orientação sobre os correspondentes requisitos de recursos para o esforço de teste. Este papel é responsável por:

- Identificar e descrever as técnicas de teste apropriadas
- Identificar as ferramentas de suporte apropriadas
- Definir e manter a Arquitetura de Automatização de Teste
- Especificar e verificar as Configurações do Ambiente de Teste Obrigatórias
- Verificar e avaliar a Abordagem de Teste

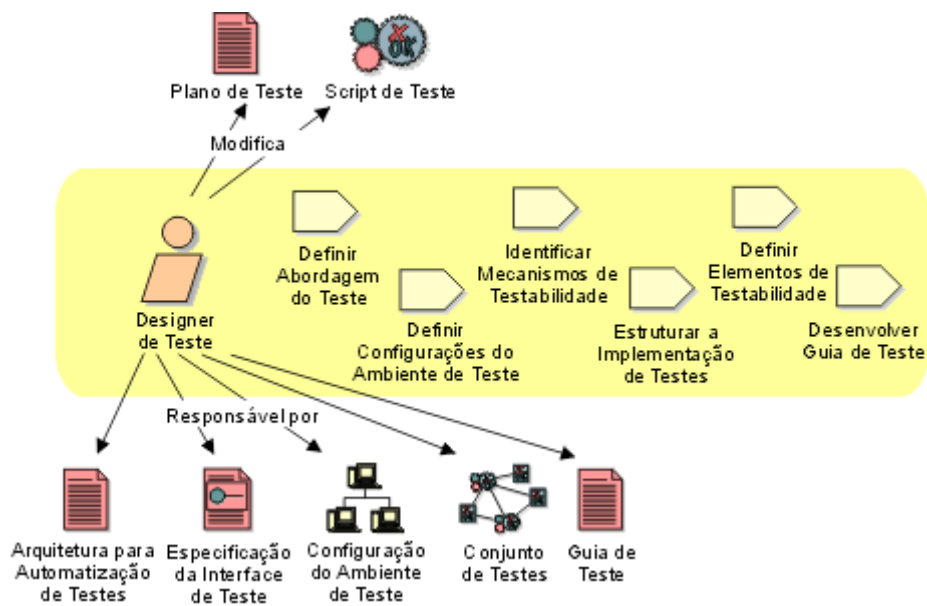


Figura 6: Designer de Teste

8.2 Atividades

1. Definir a abordagem de teste.
2. Definir configurações do ambiente de teste.
3. Identificar mecanismos de testabilidade.
4. Estruturar a implementação dos testes.
5. Definir elementos de testabilidade.
6. Desenvolver guias de teste.

8.3 Artefatos

1. Plano de teste.
2. Script de teste.
3. Conjunto de testes.
4. Guia de testes.