



Escola Superior Técnica

**GUIÃO DE ELABORAÇÃO DA DOCUMENTAÇÃO DE UM SOFTWARE
CURSO DE INFORMÁTICA**

DEPARTAMENTO DE INFORMÁTICA

Versão 1.0

**ESTEC
2015**

Este guião de apresentação de como documentar o desenvolvimento de um Software, foi elaborado pela docente do Departamento de Informática da Escola Superior Técnica.

MSc. Cláudia Ivete F. Jovo

Índice do Trabalho

1. Introdução	4
1.1. Justificativa da Escolha do Tema	4
1.2. Delimitação do Problema	4
1.3. Objectivos do Projecto	4
1.3.1. Objectivo Geral	4
1.3.2. Objectivos Específicos	4
1.4. Metodologia de Trabalho e de Desenvolvimento do Software	4
1.5. Cronograma das actividades do Projecto	4
1.6. Organização do Trabalho	5
2. Descrição Geral do Sistema	5
2.1. Descrição do Problema.....	5
2.2. Principais Envolvidos e suas Características	5
2.2.1. Utilizadores do Sistema	5
2.2.2. Desenvolvedores do Sistema	5
2.3. Regras de Negócio ^G	5
3. Requisitos G do Sistema.....	5
3.1. Requisitos Funcionais	6
3.2. Requisitos Não-Funcionais	6
3.3. Protótipo	7
3.3.1. Diagrama de Navegação	8
4. Análise e Design	8
4.1. Arquitectura do Sistema	8
4.2. Modelo do Domínio	8
4.3. Modelo de Dados	9
4.3.1. Modelo Lógico da Base de Dados	9
4.3.2. Criação Física do Modelo de Dados	9
4.3.3. Dicionário de Dados	9
4.4. Ambiente de Desenvolvimento	9
4.5. Sistemas e componentes externos utilizados (sem existirem).....	9

5. Implementação.....	9
6. Testes.....	9
6.1. Plano de Testes.....	10
6.2. Execução do Plano de Testes	10
7. Implantação.....	10
7.1. Manual de Implantação	10
8. Manual do Utilizador	10
9. Considerações Finais e Recomendações	11
10. Bibliografia.....	11

1. Introdução

O objectivo da introdução é apresentar o projecto ao leitor de forma breve. Para tal, deve-se desenvolver um texto, com as seguintes características: impessoalidade, objectividade, clareza, precisão, coerência e concisão.

1.1. Justificativa da Escolha do Tema

Neste item deve-se expor a motivação académica para a elaboração do projecto em questão, detalhando os motivos de ordem teórica ou de ordem prática para a sua realização.

1.2. Delimitação do Problema

Neste item deve ser descrita a delimitação do problema, que define o ponto central do projecto. Isso quer dizer que, dentro de uma ideia geral do projecto, deve-se ressaltar a ideia específica a efectivamente ser desenvolvida. É neste item que a amplitude do projecto tem sua delimitação perfeitamente definida.

1.3. Objectivos do Projecto

Neste item devem ser descritos os objectivos gerais e específicos do projecto como um todo. Independente do que será implementado, este item visa o entendimento global do projecto.

1.3.1. Objectivo Geral

1.3.2. Objectivos Específicos

1.4. Metodologia de Trabalho e de Desenvolvimento do Software

Neste item deve-se descrever o método a ser utilizado para realização do projecto, o tipo de processo de desenvolvimento de Software, a modelagem a ser utilizada (**orientada a objecto**, estruturada, outras).

1.5. Cronograma das actividades do Projecto

Neste item deve ser estimado os esforços necessários em termos de recursos alocados ^G e tempo para a obtenção do sistema. O cronograma detalhado do sistema deve contemplar todas as tarefas e os recursos alocados para cada tarefa, com datas para início e término de cada actividade. A sequência das tarefas e a divisão entre os recursos devem ser realizadas de acordo com o processo de desenvolvimento de Software escolhido para o desenvolvimento do Software, descrito no ponto 1.4.

Para elaboração do cronograma pode-se utilizar uma ferramenta como o *Microsoft Project* ou o *Visio*.

1.6. Organização do Trabalho

Neste item deve-se descrever como o documento estará organizado.

2. Descrição Geral do Sistema

Este capítulo tem como objectivo descrever de forma geral o sistema, o escopo e as principais funções. A descrição geral do sistema deve abranger os itens a seguir.

2.1. Descrição do Problema

Neste item deve ser descrito o problema que será resolvido com o desenvolvimento do sistema. As questões a seguir devem ser respondidas.

- ✓ Quem é afectado pelo sistema?
- ✓ Qual é o impacto do sistema?
- ✓ Qual seria uma boa solução para o problema?

2.2. Principais Envolvidos e suas Características

2.2.1. Utilizadores do Sistema

Neste item deve ser descrito para que tipo de empresa/instituição se destina o sistema, os tipos de utilizadores que utilizarão o sistema.

Estas informações são importantes para a definição de usabilidade ^G do sistema.

2.2.2. Desenvolvedores do Sistema

Neste item deve ser descrito o tipo de pessoas envolvidas em todo o desenvolvimento do sistema directo ou indirectamente.

Estas informações são importantes para a distribuição de responsabilidades e pontos-focais de desenvolvimento.

2.3. Regras de Negócio ^G

Neste item deve ser descrito as regras de negócio relevantes para o sistema, como por exemplo, restrições de negócio, restrições de desempenho, tolerância a falhas, volume de informação a ser armazenada, estimativa de crescimento de volume, ferramentas de apoio, etc.

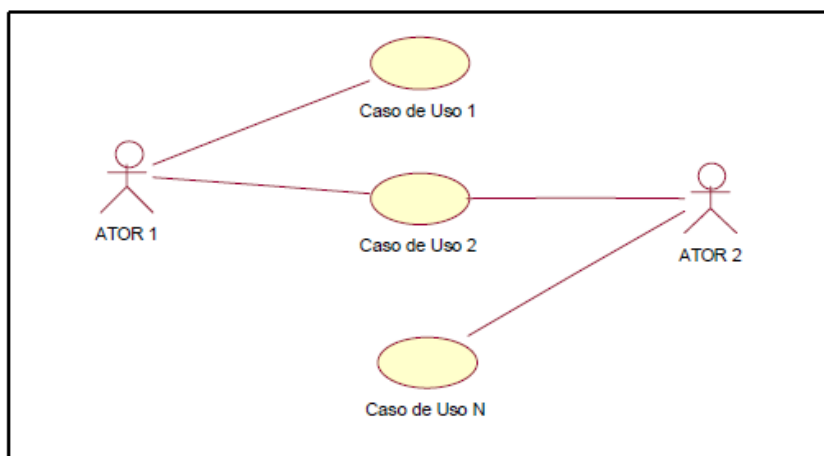
3. Requisitos ^G do Sistema

Este capítulo tem como objectivo descrever os requisitos do sistema. No caso de sistemas que possuam utilizadores reais para o levantamento de requisitos, pode-se utilizar vários métodos de levantamento de requisitos, tais como entrevista, questionários, inquéritos, etc.

3.1. Requisitos Funcionais

Neste item devem ser apresentados os requisitos funcionais que especificam acções que um sistema deve ser capaz de executar, ou seja, as funções do sistema. Os requisitos funcionais geralmente são descritos melhor em diagramas de **caso de uso**, juntamente com o detalhamento dos actores e de cada caso de uso.

A seguir é apresentada a notação básica de um diagrama de caso de uso.



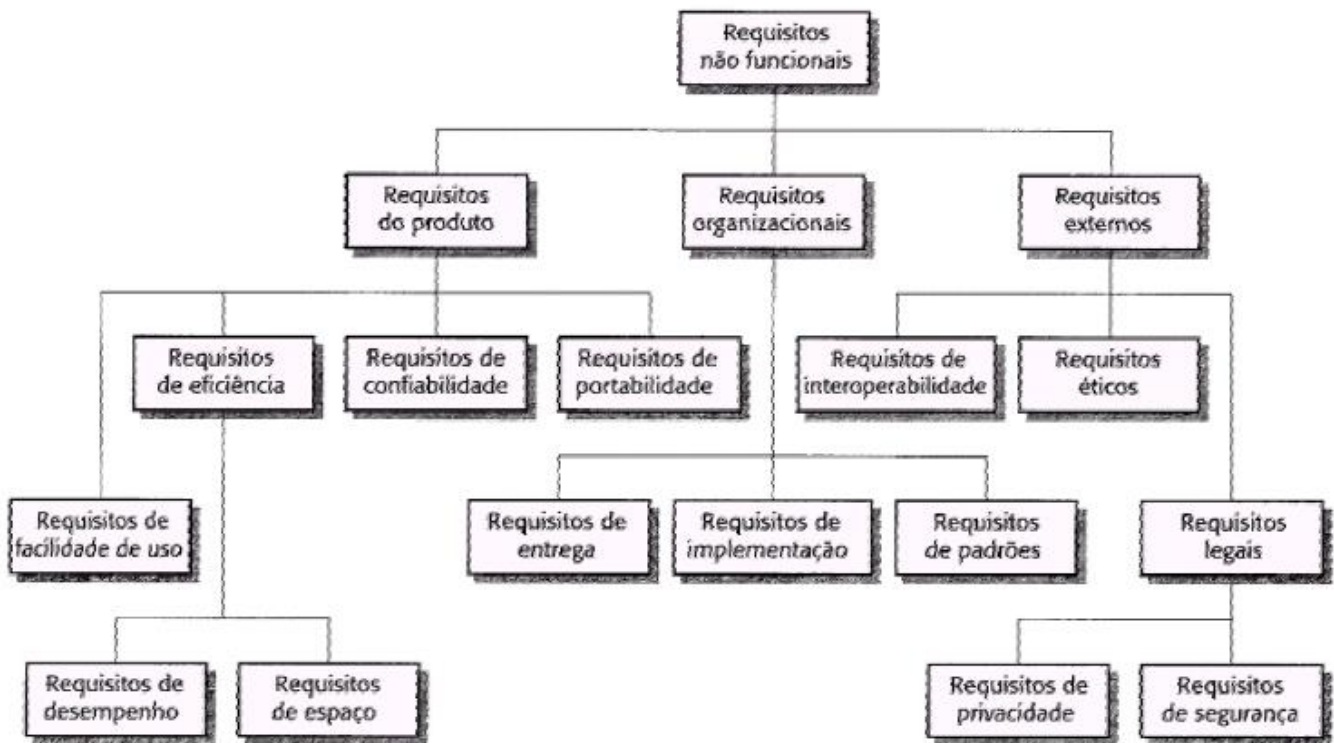
Notação básica do diagrama de caso de uso.

Cada *actor* do diagrama de caso de uso deve ser descrito de forma sucinta (2 linhas) e cada *caso de uso* deve ser especificado. A seguir são apresentados itens básicos para a especificação dos casos de uso do diagrama.

- ✓ Nome do Caso de Uso
- ✓ Breve descrição
- ✓ Actores envolvidos
- ✓ Pré-condições
- ✓ Sequência de Eventos ou Fluxo Principal de Eventos
- ✓ Pós-condições;
- ✓ Excepções ou Fluxo Secundário de Eventos
- ✓ Observações

3.2. Requisitos Não-Funcionais

Neste item devem ser apresentados os requisitos não funcionais, que especificam restrições sobre os serviços ou funções providas pelo sistema. A seguir são apresentados alguns tipos de requisitos não funcionais.



Sommerville, Ian. Engenharia de Software, 6ª edição.

3.3. Protótipo

Neste item deve ser apresentado o protótipo do sistema que consiste na interface preliminar contendo um subconjunto de funcionalidades e telas. O protótipo pode sofrer mudanças na fase da codificação, mas isso em concordância completa dos requisitos previstos para o sistema, de acordo comum com o utilizador. O protótipo é um recurso que deve ser adoptado como estratégia para levantamento, detalhamento, validação de requisitos e modelagem de interface com o utilizador.

Os formulários do sistema podem ser criados na própria linguagem de desenvolvimento ou em qualquer outra ferramenta de desenho. Cada formulário deve possuir uma descrição detalhada do seu funcionamento.

Alguns itens importantes na descrição são:

- Objectivo do formulário;
- Como é que surge (onde é chamada) e que outros formulários podem ser abertos (chamados);
- Regras:
 - Domínio (tamanho de campo, tipo de dados que aceita valor *default*);
 - Tipo de utilizador que pode aceder;
 - Lógica de negócio (campos obrigatórios, validade entre datas, preenchimento anterior de um campo para efectuar uma operação, etc).

A descrição detalhada dos formulários deve constar informações que possam ser consultadas na implementação do sistema, facilitando, agilizando e minimizando erros de implementação e na execução de testes.

3.3.1. Diagrama de Navegação

Neste item deve ser apresentada a sequência de navegação dos formulários.

4. Análise e Design

Este capítulo tem como objectivo analisar e detalhar a solução do sistema de acordo com os requisitos levantados e validados no capítulo 3. Para isso, deve-se ter uma visão geral da arquitectura do sistema e a modelagem da solução do sistema através de diagramas.

4.1. Arquitectura do Sistema

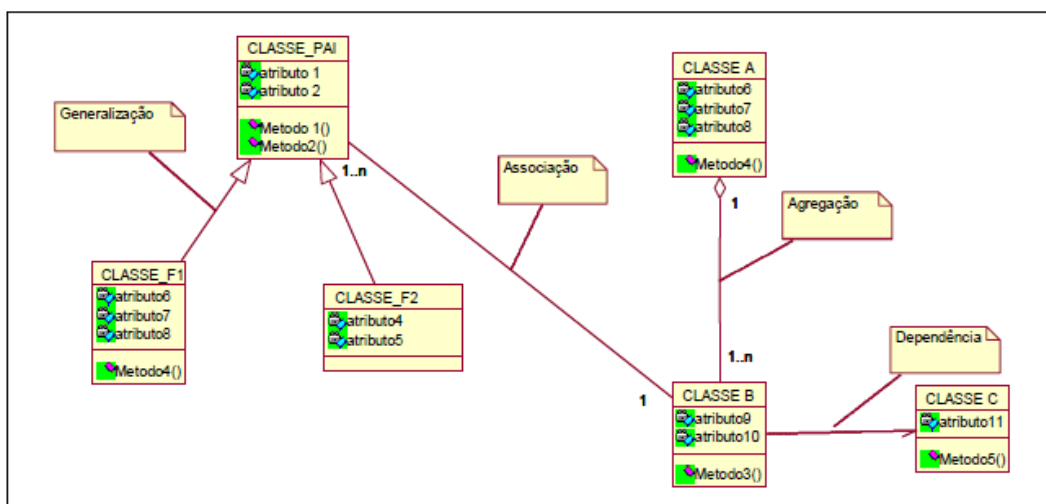
Neste item deve ser apresentada a arquitectura de infra-estrutura do sistema, demonstrando o tipo de arquitectura que será utilizada (por exemplo, cliente/servidor de n-camadas), bem como o mínimo de conexões ao Software.

4.2. Modelo do Domínio

Neste item deve ser apresentado o modelo do domínio, que representa o modelo conceitual do diagrama de classes.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e o relacionamento entre elas.

A seguir é apresentada a notação básica de um diagrama de classes



Notação básica do diagrama de classes.

4.3. Modelo de Dados

4.3.1. Modelo Lógico da Base de Dados

Neste item deve ser apresentado o modelo lógico da base de dados, que pode ser o modelo entidade-relacionamento. No modelo entidade-relacionamento o modelo lógico deve passar por todas as regras de normalização.

Como base para geração do modelo lógico pode-se utilizar o diagrama de classes.

4.3.2. Criação Física do Modelo de Dados

Neste item deve ser realizada a criação física da base de dados, ou seja, a criação de *scripts*.

4.3.3. Dicionário de Dados

Neste item deve ser criado o dicionário de dados da base de dados, com o objectivo de documentar todas as tabelas, atributos, *stored procedures* ^G.

4.4. Ambiente de Desenvolvimento

Neste item devem ser apresentados os Softwares de desenvolvimento (linguagem de programação, base de dados, ferramentas, etc.), equipamentos de Hardware e de rede que sejam essenciais para o desenvolvimento do Software.

4.5. Sistemas e componentes externos utilizados (sem existirem)

Neste item devem ser descritos os sistemas e componentes externos que serão utilizados no funcionamento do Software. Por exemplo, sistemas que serão integrados ao sistema desenvolvido, componentes comprados ou livres.

5. Implementação

Este capítulo tem como objectivo a implementação das classes em termos de componentes, ou seja, toda a implementação deve ser realizada de acordo com as definições das fases anteriores e todos os recursos da programação orientada a objectos que a linguagem escolhida oferece.

Geralmente ferramentas CASE geram automaticamente pseudocódigos fontes (dependendo da linguagem utilizada) baseados no diagrama de classes.

Algumas boas práticas de programação devem ser seguidas para um maior entendimento do código.

6. Testes

Este capítulo tem como objectivo identificar defeitos no sistema, validar as funções do sistema, verificar se os requisitos foram implementados de forma adequada e avaliar a qualidade do Software.

6.1. Plano de Testes

Neste item deve ser criado o plano de testes do Software, permitindo a validação do sistema por parte do desenvolvedor, através da verificação dos requisitos do sistema desenvolvido. Inicialmente, identificam-se os requisitos técnicos e funcionais do Software, e listam-se todas as situações que podem ocorrer com o Software. Deve-se realizar testes de consistência de campos, funcionalidades, desempenho, etc. O Plano de Testes do Sistema deverá conter, no mínimo.

- N° do Teste;
- Descrição do Teste;
- Resultado Esperado.

Por conter todos os testes do Software, este plano poderá ser um anexo na documentação do Software.

Alguns tipos de testes a serem realizados são: teste de funcionalidades, teste de usabilidade, teste de desempenho, teste de segurança e controle de acesso, teste de tolerância a falhas e recuperação, etc..

6.2. Execução do Plano de Testes

Neste item devem ser registados os testes realizados no Software tendo como base o Plano de Testes do Software.

O registo dos testes deve conter a identificação do sistema, o nome do realizador dos testes e a configuração do ambiente onde foi realizado o teste. Além disso, para cada teste, deve-se ter os seguintes dados:

- N° do teste;
- Resultado Obtido.

7. Implantação

Este capítulo tem como objectivo apresentar informações relevantes para a implantação e funcionamento do Software.

7.1. Manual de Implantação

Neste item deve ser elaborado o manual de instalação. Este manual deve conter a descrição passo a passo de como deve ser realizada a instalação do sistema.

8. Manual do Utilizador

Este capítulo tem como objectivo a elaboração de um manual do utilizador. Este manual deve conter a descrição passo a passo de como utilizar o Software.

9. Considerações Finais e Recomendações

Este capítulo tem como objectivo apresentar e demonstrar a aplicabilidade dos resultados obtidos, suas limitações, inovação, possíveis integrações com outros projectos e melhoramento do Software em projectos futuros.

10. Bibliografia

Neste item devem-se apresentar todas as obras (livros, artigos, Internet, revistas, etc...) utilizadas na elaboração da documentação e na implementação do projecto.

Glossário

[1] **Componente:** representa uma parte física da implementação de um sistema, que inclui código de Software, com o objectivo de criar código de Software coeso para sua reutilização e facilidade de manutenção.

[2] **Ferramenta CASE (Computer Aided Software Engineering):** é uma ferramenta que auxilia no processo de desenvolvimento de Software, ajudando a garantir a qualidade do projecto e facilitando a criação de modelos, documentos.

[3] **Padrões de Projecto (design patterns):** são soluções simples para problemas específicos no projecto de Software orientado a objectos. Padrões de projecto capturam soluções que foram desenvolvidas e aperfeiçoadas ao longo do tempo.

[4] **Recursos alocados:** pessoas que irão trabalhar no projecto.

[5] **Regras de negócio:** declarações e regras da política ou condição que deve ser satisfeita no âmbito do negócio.

[6] **Requisito:** um requisito descreve uma condição ou capacidade à qual um sistema deve se adaptar, sejam necessidades dos usuários, um padrão ou uma especificação.

[7] **Stored Procedures:** é uma rotina escrita através de comandos SQL, que tem como objectivo encapsular o processo de negócio e sua reutilização. As stored procedures ficam armazenadas no gerenciador de base de dados.





[8] **Usabilidade:** é a qualidade da interface homem-máquina, que permite que o utilizador realize com eficiência e conforto as actividades a que o sistema se destina.







Material que podem consultar caso queiram ter mais subsídios para o projecto

- [1] FOWLER, M.; SCOTT.K. *UML Essencial*. Editora Bookman, 3ª edição.
- [2] BOOCH, G.; RUMBAUGH, J.; JACOBSON. I. *UML – Guia do Usuário*. Editora Campus, 2000.
- [3] BOGGS, W.; BOGGS, M. *Mastering UML com Rational Rose 2002*. Editora Alta Books, 2002.
- [4] QUATRANI. T. *Modelagem Visual com Rational Rose 2000 e UML*. Editora Ciência Moderna, 2001.
- [5] PRESSMAN, R. *Software Engineering. A Practitioner's Approach*. 5ª edição, 2003. McGrawHill.
- [6] SOMERVILLE, I. *Engenharia de Software*. Addison Wesley, 6ª edição.
- [7] YOURDON, E *Análise Estruturada Moderna*. Editora Campus, 3ª edição.
- [8] Ferramenta *Rational Unified Process*.

Ferramentas CASE que podem ser utilizadas para o desenvolvimento do projecto

Rational Suite Enterprise

-  *Rational Rose* para modelagem de sistemas orientados a objectos;
-  *Rational Unified Process* - RUP para utilização do processo de desenvolvimento e para consulta de artefactos, definições, etc;
-  *Requisite Pro* para criação e gerenciamento de requisitos.
-  Além de outras ferramentas da Suíte...

-  *System Architect* para modelagem de sistemas orientados a objectos e estruturados.
-  *Microsoft Project* para elaboração de cronograma, panejamento e gerenciamento do projecto.
-  *Erwin* para elaboração do modelo lógico e físico do banco de dados;
-  *ArgoUML* para modelagem de sistemas orientados a objectos.
-  *Dome* para modelagem de sistemas orientados a objectos.
-  Outras....