

Levantamento e Análise de Requisitos



Prof. Jeferson Souza, MSc. (thejefecomp)

jeferson.souza@udesc.br



UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

JOINVILLE
CENTRO DE CIÊNCIAS
TECNOLÓGICAS

O que é um Requisito?

Requisito

Um requisito pode ser definido como uma necessidade, algo que é desejado e esperado. Exemplo: prestar atenção no professor é um requisito da disciplina de Programação Orientada a Objetos (:-D).

Para que serve o Levantamento de Requisitos?

Finalidade

Descobrir e entender as reais necessidades do cliente, ou seja, o que o cliente realmente quer.

Para que serve o Levantamento de Requisitos?

Finalidade

Descobrir e entender as reais necessidades do cliente, ou seja, o que o cliente realmente quer.

Ahh professor, isso é fácil....

Nãoooooooooooo! O levantamento de requisitos está longe de ser uma tarefa fácil...

Para que serve o Levantamento de Requisitos?

Finalidade

Descobrir e entender as reais necessidades do cliente, ou seja, o que o cliente realmente quer.

Ahh professor, isso é fácil....

Nãoooooooooooo! O levantamento de requisitos está longe de ser uma tarefa fácil...

Por que?

Nem sempre o cliente sabe as suas reais necessidades, ou seja, não sabe muito bem o que quer.

O que esperar do Levantamento de Requisitos?

- ▶ Compreender quais são as reais necessidades do cliente (o que o cliente realmente quer);
- ▶ Compreender o negócio para qual a solução (produto/software) será desenvolvida;
- ▶ Identificar pessoas que podem auxiliar no processo de especificação e entendimento dos requisitos;
- ▶ Elaborar uma lista com os requisitos que descrevem as necessidades do cliente;

O que esperar do Levantamento de Requisitos? (Continuação)

- ▶ Identificar e remover ambiguidades entre os requisitos;
- ▶ Criar casos de uso para auxiliar a identificação dos principais requisitos;
- ▶ Em alguns casos, pode-se também produzir um protótipo simples para auxiliar a definição e o entendimento dos requisitos.

Principais dificuldades

- ▶ Definição de escopo;
- ▶ Entendimento do problema/necessidade;
- ▶ Mudanças.

Definição de escopo

Problemas de escopo

Durante o levantamento de requisitos os limites da solução (produto/software) não ficam muito bem definidos, ou o cliente especifica detalhes técnicos que mais confundem do que ajudam a definir claramente os objetivos do sistema.

Entendimento do Problema/Necessidade

Entendimento

O cliente tem dificuldade para saber o que realmente quer, um baixo conhecimento do seu próprio negócio, e/ou problemas em comunicar suas necessidades.

Mudanças

Ahhh a passagem do tempo....

Os requisitos podem mudar com o passar do tempo, e então atualizações precisam ser realizadas nos requisitos já definidos anteriormente.

Iniciar o Levantamento de Requisitos

O início

O método mais comum para iniciar o levantamento de requisitos é realizar uma reunião ou entrevista com o cliente.

Iniciar o Levantamento de Requisitos

O início

O método mais comum para iniciar o levantamento de requisitos é realizar uma reunião ou entrevista com o cliente.

Dificuldades?

Sim, o início nunca é fácil!

Iniciar o Levantamento de Requisitos

Iniciar o levantamento de requisitos passa por:

- ▶ Falta e dificuldade de comunicação entre as partes (cliente e equipe técnica);
- ▶ Entendimentos divergentes do mesmo problema/domínio;
- ▶ Nenhuma das partes sabe como e o que perguntar;
- ▶ Expectativas podem ser diferentes (pelo menos no início).

Iniciar o Levantamento de Requisitos

Iniciar o levantamento de requisitos passa por:

- ▶ Falta e dificuldade de comunicação entre as partes (cliente e equipe técnica);
- ▶ Entendimentos divergentes do mesmo problema/domínio;
- ▶ Nenhuma das partes sabe como e o que perguntar;
- ▶ Expectativas podem ser diferentes (pelo menos no início).

Porém...

Ambas as partes (cliente e equipe técnica) tem o desejo que o relacionamento, que começa a ser estabelecido, seja bem sucedido.

Iniciar o Levantamento de Requisitos

E então, como começar?

Iniciar o Levantamento de Requisitos

E então, como começar?

Comece com perguntas mais genéricas, tais como:

- ▶ Quais serão os benefícios do software para a sua empresa?

Qual o objetivo dessas perguntas?

Ganhar o entendimento do cliente, dos objetivos gerais, e dos benefícios que a solução deve fornecer.

Iniciar o Levantamento de Requisitos

E então, como começar?

Comece com perguntas mais genéricas, tais como:

- ▶ Quais serão os benefícios do software para a sua empresa?
- ▶ Quem vai usar o software?

Qual o objetivo dessas perguntas?

Ganhar o entendimento do cliente, dos objetivos gerais, e dos benefícios que a solução deve fornecer.

Iniciar o Levantamento de Requisitos

Na sequência, é necessário entender o problema e as expectativas do cliente a respeito do software. Para isso, faça perguntas tais como:

- ▶ Que tipo de saída (resultado) você espera que o software forneça? Um gráfico? Uma tabela?
- ▶ Qual são os principais problemas que o software poderá resolver? Melhorias de processo? Agilidade no acesso a informação?
- ▶ Qual é o ambiente e qual o perfil das pessoas que utilizarão o software?

Iniciar o Levantamento de Requisitos

Por fim, é necessário identificar se as pessoas presentes na reunião são realmente quem devem responder todas as perguntas. Logo, o papel da equipe técnica é conduzir o foco da reunião:

- ▶ Existe mais alguma pessoa que deve ser envolvida no processo?
- ▶ As respostas às minhas perguntas são oficiais, ou ainda precisam ser validadas?
- ▶ Será que chegamos a uma visão geral e conjunta da solução?

Trabalhar em Equipe Com o Cliente

- ▶ Necessidade de quebrar a barreira que coloca o cliente em uma posição isolada, e com uma visão da solução que pode ser diferente da visão a ser desenvolvida;
- ▶ Sessões de perguntas e respostas, juntamente com reuniões similares ao início do projeto não funcionam;
- ▶ É necessário trabalhar em conjunto para refinar os requisitos.

Aplicar a Abordagem FAST

O termo *FAST* vem do inglês *Facilitate Application Specification Techniques*, e descreve a criação de uma equipe em conjunto com o cliente para realizar o levantamento de requisitos de forma eficiente. A abordagem *FAST* auxilia:

- ▶ Identificar o problema de forma eficiente;
- ▶ Definir e propor aspectos da solução;
- ▶ Estabelecer uma negociação dos requisitos e da solução;
- ▶ Definir um conjunto preliminar de requisitos da solução de software a ser implementada.

Principais Características da Abordagem FAST

- ▶ Reuniões em locais neutros (de preferência);
- ▶ Estabelecimento de regras de preparação para os participantes;
- ▶ Cada reunião tem uma agenda proposta que deve seguida, mas ao mesmo tempo deve permitir a exposição de ideias;
- ▶ A existência da figura de um “mediador” que tem o controle da reunião;
- ▶ Ata do que foi discutido e decidido na reunião.

Pré-Requisitos da Abordagem FAST

Antes de iniciar a sequência de reuniões usando a abordagem *FAST*, alguns pré-requisitos devem ser assegurados:

- ▶ Escopo e visão geral da solução bem definidos;
- ▶ Especificação de um documento curto (1 ou 2 páginas) a descrever os objetivos, escopo, e a visão geral da solução (o que será feito);
- ▶ Definição do mediador (cliente, engenheiro de software, analista de negócio, consultor externo, entre outros);
- ▶ Definição de local, data e hora.

Pré-Requisitos da Abordagem FAST

Antes da primeira reunião cada participante deve fazer uma lista com os seguintes itens:

- ▶ Aspectos do ambiente onde a solução será utilizada;
- ▶ O que deve ser produzido pela solução;
- ▶ Que tipo de recurso deve ser utilizado pela solução;
- ▶ Recursos/serviços (processos ou funções) que manipulam os dados ou interagem com a solução;
- ▶ Restrições em termos de custo, tamanho, regras de negócio, etc.

Exemplo de Descrição de Solução

Exemplo [Pressman, 2001]

Nossas pesquisas indicam que o mercado para sistemas de vigilância doméstica está crescendo a uma taxa de 40% ao ano. Portanto, a idéia da empresa é entrar nesse mercado com a criação de um sistema de vigilância doméstica baseado em microprocessador, sistema este a permitir (idealmente) a proteção e o reconhecimento de um conjunto de incidentes indesejáveis tais como: entrada não-autorizada, fogo, alagamento, entre outros. O produto, cujo o nome preliminar é *SafeHome*, usará sensores apropriados para detectar cada incidente indesejado, poderá ser programado pelo próprio dono da propriedade, e telefonará para uma equipe de monitoramento (empresa de segurança) caso algum dos incidentes indevidos ocorra.

Exemplo: Aspectos do Ambiente

- ▶ Detectores de fumaça;
- ▶ Sensores de portas e janelas;
- ▶ Sensores de detecção de movimento;
- ▶ Eventos (um sensor detecta algo e é ativado);
- ▶ Painel de controle;
- ▶ Entre outros [**A Lista deve ser explícita e fechada como nos exemplos a seguir**].

Exemplo: O que Deve Ser Produzido

- ▶ Alerta telefônico;
- ▶ Alarme sonoro;
- ▶ Controle de incêndio;
- ▶ Isolamento de área afetada.

Exemplo: Recursos/Serviços

- ▶ Configuração do alarme;
- ▶ Programação do sistema (liga/desliga sensores, senha de acesso);
- ▶ Monitoramento dos diferentes sensores;
- ▶ Acionamento de portas corta fogo;
- ▶ Chamada telefônica.

Exemplo: Restrições

- ▶ Custo de produção inferior a R\$200;
- ▶ Interface de utilização amigável (fácil de usar e intuitiva);
- ▶ Interagir diretamente com sistema telefônico;
- ▶ Incidente indesejável deve ser reconhecido dentro de 1 segundo;
- ▶ Definir prioridade de eventos (fogo é mais prioritário que abertura de janela).

Aspectos da primeira reunião

- ▶ Discutir a necessidade e a justificativa da nova solução (todos devem concordar nesses pontos);
- ▶ Apresentação das lista de itens (aspectos, resultados esperados, recursos/serviços, restrições) de cada um dos participantes;
- ▶ Criação de uma lista de itens única pelo grupo de participantes;
- ▶ divisão do grupo em pequenos grupos de trabalho, os quais produzirão as especificações do sistema.

Aspectos das reuniões seguintes

- ▶ Apresentação das especificações que forem sendo produzidas sobre a solução;
- ▶ Alteração das especificações (inclusão, atualização, e remoção de itens).

Aspectos das reuniões seguintes

- ▶ Apresentação das especificações que forem sendo produzidas sobre a solução;
- ▶ Alteração das especificações (inclusão, atualização, e remoção de itens).

Importante!

Todos os aspectos da reunião devem ser coordenados pelo mediador.

Classificação dos Requisitos

Os requisitos pode ser classificados em três categorias:

- ▶ Normal;
- ▶ Esperado;
- ▶ Diferencial.

Requisitos Normais

Definição

Descrevem os objetivos que foram definidos juntamente com o cliente durante as reuniões.

Requisitos Normais

Definição

Descrevem os objetivos que foram definidos juntamente com o cliente durante as reuniões.

Importante!

A presença dos requisitos normais já deixa o cliente satisfeito. Exemplo: O sistema deve produzir relatórios com indicadores de custo de produção por hora, dia, e mês.

Requisitos Esperados

Definição

Descrevem os requisitos que são implícitos da solução (produto/-software) a ser desenvolvido. Exemplo: o sistema deve fornecer os resultados em um tempo de, no máximo, 3 segundos.

Requisitos Esperados

Definição

Descrevem os requisitos que são implícitos da solução (produto/-software) a ser desenvolvido. Exemplo: o sistema deve fornecer os resultados em um tempo de, no máximo, 3 segundos.

Importante!

A presença dos requisitos esperados é fundamental para assegurar os requisitos normais, e o bom funcionamento da solução.

Requisitos Diferenciais

Definição

Descrevem características que vão além da expectativa do cliente, e deixam o mesmo ainda mais satisfeito. Exemplo: o sistema deve fornecer uma planejamento de custos baseado no histórico de utilização do cliente.

Requisitos Diferenciais

Definição

Descrevem características que vão além da expectativa do cliente, e deixam o mesmo ainda mais satisfeito. Exemplo: o sistema deve fornecer uma planejamento de custos baseado no histórico de utilização do cliente.

Importante!

A presença dos requisitos diferenciais permite um aumento do valor agregado da solução.

Unified Modelling Language - UML

O que é UML?

A Unified Modelling Language (UML) é uma linguagem de modelagem utilizada na concepção de sistemas de software. A UML possui muitos diagramas para auxiliar a análise e modelagem de sistemas de software, e para auxiliar o levantamento de requisitos será utilizado o **Diagrama de Casos de Uso**.

Introdução aos Casos de Uso

O que são casos de uso?

Um caso de uso é uma representação de um cenário, o qual descreve como o sistema a ser implementado será usado em uma situação específica. Exemplo: Sacar dinheiro em um sistema bancário.

Introdução aos Casos de Uso

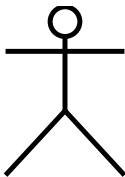
O que são casos de uso?

Um caso de uso é uma representação de um cenário, o qual descreve como o sistema a ser implementado será usado em uma situação específica. Exemplo: Sacar dinheiro em um sistema bancário.

Qual a importância dos casos de uso?

Compreender em maior detalhes como o sistema será usado, para então identificar com maior clareza quais são os requisitos normais, esperados, e diferenciais. Além disso, casos de uso também podem apoiar a priorização de análise e implementação dos requisitos levantados.

Diagrama de Casos de Uso

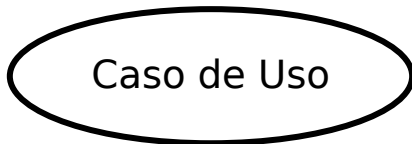


Ator

Ator

Atores são entidades que interagem com o sistema. Essas entidades podem ser pessoas (utilizadores) ou outros sistemas. Cada ator representa um único papel na utilização do sistema. Exemplos de atores: Cliente, Administrator, Gestor, Balconista, Sistema de Pagamento, etc.

Diagrama de Casos de Uso



Caso de Uso

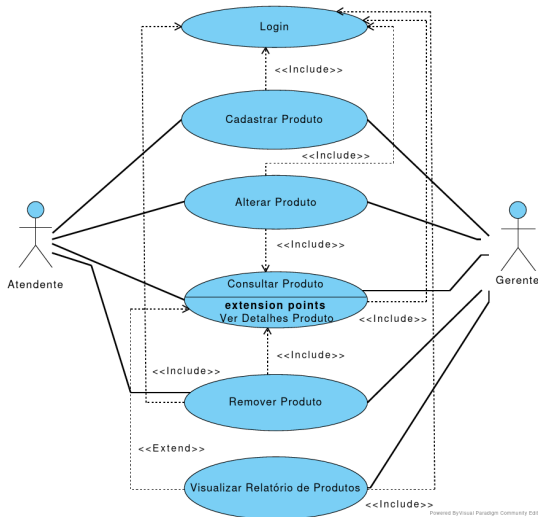
Casos de Uso representam o que um dado sistema deve fazer. Entretanto, um caso de uso não representa apenas o que o sistema deve fazer, mas sim o comportamento do sistema quando um dado ator utiliza-o.

Diagrama de Casos de Uso

Diagrama de Casos de Uso

Um diagrama de casos de uso descreve o comportamento do sistema em situações específicas. Logo, o diagrama representa a associação entre atores e casos de uso, assim como (caso existam) as relações entre diferentes casos de uso.

Exemplo de Diagrama de Casos de Uso



Exemplo de Descrição de Caso de Uso

Nome: Cadastrar Produto

Descrição: Este caso de uso permite o cadastro de produtos no sistema de controle de estoques. Diferentes atores (atendente e gerente) podem cadastrar produtos no sistema.

Pré-Condições: O utilizador deve estar logado no sistema.

Pós-Condições: Caso todas as informações estejam corretas, o novo produto é armazenado na base de dados do sistema. Caso contrário, o produto não é armazenado na base de dados.

Exemplo de Descrição de Caso de Uso

Fluxo normal:

1. A qualquer momento antes de salvar o novo produto na base de dados do sistema, o utilizador (Atendente ou Gerente) pode cancelar o cadastro do produto;
2. Utilizador (Atendente ou Gerente) seleciona tipo de produto a ser cadastrado;
3. Utilizador (Atendente ou Gerente) preenche as informações do produto a ser cadastrado (código, descrição, quantidade, etc..);
4. Utilizador (Atendente ou Gerente) salva o produto na base de dados do sistema.

Exemplo de Descrição de Casos de Uso

Fluxos alternativos:

Condição 2: Utilizador (Atendente ou Gerente) tenta cadastrar produto já existente na base de dados do sistema.

1. Ao tentar cadastrar o produto na base de dados do sistema, o utilizador (Atendente ou Gerente) recebe um alerta do sistema indicando que o produto a ser salvo já existe na base de dados;
2. Após emitir o alerta para o utilizador (Atendente ou Gerente) o sistema indica os campos que devem ser únicos, ou seja, não podem pertencer a mais nenhum produto. O utilizador (Atendente ou Gerente) então decide o que fazer;
3. Se o utilizador (Atendente ou Gerente) alterar as informações para cadastrar um novo produto, o produto é salvo na base de dados do sistema. Caso contrário, O utilizador (Atendente ou Gerente) cancela o cadastro.

Exemplo de Descrição de Casos de Uso

Fluxos alternativos:

Condição 3: ...

Condição 4: ...

Condição 5: ...

Importante!

A descrição de um caso de uso não deve ser muito extensa. Portanto, não é preciso especificar uma lista exaustiva de fluxos alternativos. Concentre-se nos fluxos mais importantes.

Exercício de Fixação

Exercício

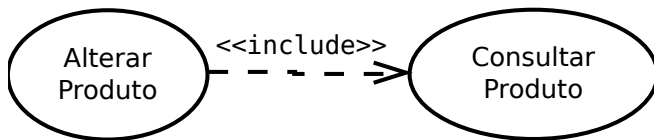
Escreva a descrição do caso de uso **Remover Produto**.

Inclusão de Casos de Uso

Inclusão (<<include>>)

Às vezes se faz necessária a utilização de um caso de uso dentro de outro, principalmente quando um determinado comportamento é comum a partes diferentes do sistema. Quando o comportamento de um caso de uso é parte obrigatória de outros casos de uso, utiliza-se a inclusão (<<include>>).

Exemplo de Inclusão de Casos de Uso



Detalhes

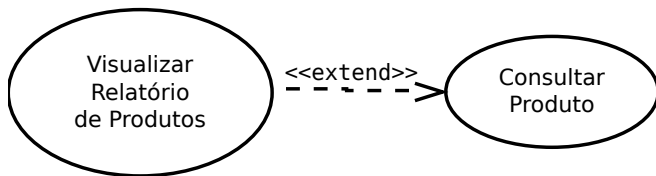
No exemplo acima, o caso de uso **Consultar Produto** é incluído como parte obrigatória do comportamento do caso de uso **Alterar Produto**. O motivo é simples: é necessário consultar as informações de um produto antes de poder realizar qualquer alteração. Essa inclusão é realizada através de um ponto de inclusão (Inclusion point), a especificar exatamente onde o comportamento do caso de uso incluído deve ser executado.

Extensão de Casos de Uso

Extensão (<<extend>>)

Diferente da inclusão que é obrigatória, a extensão (<<extend>>) de casos de uso permite a inclusão de um caso dentro do outro como parte opcional.

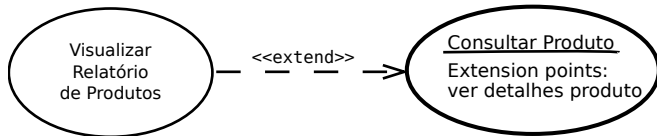
Exemplo de Extensão de Casos de Uso



Detalhes

No exemplo acima, o caso de uso **Consultar Produto** é incluído como parte opcional do comportamento do caso de uso **Alterar Produto**. O motivo é simples: a consulta de um determinado produto só é necessária caso o utilizador (Gerente) deseje ver mais detalhes do produto. O uso de extensão de casos de uso é indicado por pontos de extensão (Extension points).

Exemplo de Pontos de Extensão



Pontos de Extensão (Extension Points)

Os Pontos de extensão indicam em que momento e condição a extensão de caso de uso deve ser executada. No exemplo acima, o caso de uso **Consultar Produto** é executado assim que a condição ver detalhes do produto é disparada.

Diferenças Entre <<include>> e <<extend>>

	<<include>>	<<extend>>
O caso de uso é opcional?	Não	Sim
O caso de uso base completa sem o caso de uso que é incluído/estendido?	Não	Sim
A execução do caso de uso é condicionada?	Não	Sim
O caso de uso muda o comportamento do caso de uso base?	Não	Sim

Análise de Requisitos

A análise de requisitos consiste em:

- ▶ Verificar a consistência dos requisitos que foram levantados;
- ▶ Verificar se os requisitos levantados não são redundantes;
- ▶ Verificar se não faltam requisitos para suprir o que o cliente/usuário espera do sistema;
- ▶ Extrair/Especificar os requisitos funcionais e não funcionais do software a ser desenvolvido;

Análise de Requisitos (Continuação)

- ▶ Organizar os requisitos em categorias;
- ▶ Identificar o relacionamento entre os requisitos;
- ▶ Identificar a importância de cada um dos requisitos funcionais e não funcionais (Priorização).

Perguntas Importantes na Análise de Requisitos

Perguntas que devem ser feitas durante a análise:

- ▶ Cada um dos requisitos levantados é consistente com o objetivo do software a ser desenvolvido?
- ▶ O requisito levantado é realmente necessário? Ou pode ser classificado como algo adicional/diferencial?
- ▶ Existe ambiguidade nos requisitos levantados?
- ▶ O escopo de todos os requisitos levantados está bem definido?

Perguntas Importantes na Análise de Requisitos (Continuação)

- ▶ Existem conflitos entre os requisitos levantados?
- ▶ Cada um dos requisitos levantados é tecnicamente viável, e portanto pode ser implementado?
- ▶ Cada um dos requisitos levantados pode ser testado após ser implementado?

Especificação de Requisitos

Especificação de Requisitos

A especificação dos requisitos consiste na análise dos requisitos levantados para realizar uma tradução e classificação em requisitos funcionais e requisitos não funcionais do software a ser implementado.

Requisitos Funcionais e Não-Funcionais

Requisitos Funcionais

Requisitos funcionais são requisitos a descrever elementos que fazem parte da(s) funcionalidade(s) oferecida(s) pelo software.

Requisitos Não-Funcionais

Requisitos não-funcionais são requisitos a descrever elementos que não fazem parte da(s) funcionalidade(s) oferecida(s) pelo software, porém são importantes no suporte do oferecimento dessa(s) funcionalidade(s).

Definição de Nomenclatura

É importante definir uma nomenclatura a ser utilizada na produção de um documento formal, o qual deve incluir a especificação dos requisitos funcionais e não-funcionais do software a ser desenvolvido.

Nomenclatura (Exemplo)

- ▶ **RF** - Para definir um requisito funcional. Exemplo: **RF01** - Tela de Cadastro de Produto;
- ▶ **RNF** - Para definir um requisito não-funcional. Exemplo: **RNF01** - O sistema deve produzir o relatório de lista de produtos em, no máximo, 3 segundos.

Exemplos de Requisitos Funcionais e Não-Funcionais

Exemplo de Requisito Funcional

RF01 - Tela de Cadastro de Produto

Descrição: A tela de cadastro de produto fornece uma interface gráfica a permitir que, um utilizador do sistema, cadastre um novo produto na base de dados. Para tal, a tela de cadastro de produto possui os seguintes elementos:

Código do produto: O código do produto deve possuir denominação de texto simples “Código do Produto”, além de uma caixa de texto para informar o código do produto a ser cadastrado. A caixa de texto deve permitir a visualização de 10 caracteres. A caixa de texto deve permitir somente a entrada de códigos com o seguinte formato: **PRD-XXXXXXX**. O termo **XXXXXX** tem um tamanho fixo de sete (7) caracteres, e pode ser composto por: (a) somente letras maiúsculas de A a Z; (b) somente números de zero (0) a nove (9); ou (c) uma combinação de letras maiúsculas de A a Z e números de zero (0) a nove (9). Em qualquer um dos casos (de (a) a (c)) letras e números podem se repetir. Exemplos de códigos válidos: **PRD-ABCDEFGH**, **PRD-0001234**, **PRD-AZ4576D**.

E assim sucessivamente ...

Exemplos de Requisitos Funcionais e Não-Funcionais

Exemplo de Requisito Não-Funcional

RNF01 - O sistema deve produzir o relatório de lista de produtos em, no máximo, 3 segundos

Descrição: Após o utilizador entrar na opção de menu (Relatórios → Lista de Produtos) o sistema deverá realizar o processo de produção do relatório citado em, no máximo, 3 segundos. Em caso de falha na produção do relatório de Lista de Produtos, o sistema deve notificar o utilizador com uma mensagem de erro, a indicar o motivo pelo qual o relatório não pode ser produzido. A mensagem não deve conter elementos técnicos, e utilizar uma linguagem que esteja no domínio do utilizador do sistema. Além disso, a notificação de erro ao usuário deve também ocorrer em, no máximo, 3 segundos.

Exercício

O Desafio da Petshop

Sua equipe de desenvolvimento foi contratada para desenvolver um sistema para gerir uma petshop. Você ficou responsável pelo levantamento e análise dos requisitos do sistema. Para completar essa tarefa você deve realizar o levantamento de cinco requisitos, cuja a análise resulte na obtenção de, ao menos, quatro requisitos funcionais e um requisito não-funcional. Durante a fase de levantamento de requisitos, você também possui a missão de especificar os casos de uso que descrevem o cadastro de clientes, o cadastro de animais associados aos seus clientes, a consulta de informações referentes a um dado animal (ex: última vacina realizada, última lavagem, última tosagem, etc), e um relatório dos animais que são atendidos pelo estabelecimento. Portanto, mãos na massa para completar o desafio :-D.



Bibliografia



Booch G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., and Houston, K. A. *“Object-Oriented Analysis and Design With Applications”*. 3rd edition. Addison-Wesley, 2007.

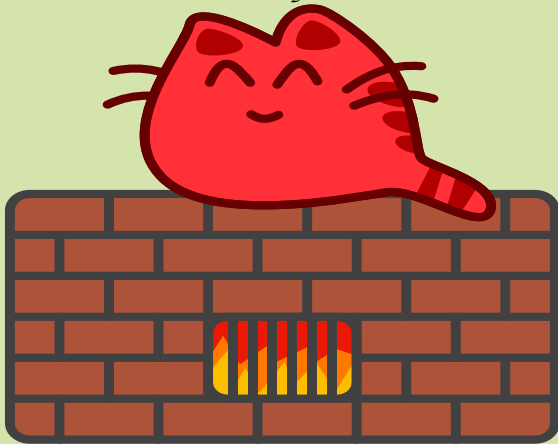


OMG. *“OMG Unified Modeling Language (OMG UML)”*, version 2.5.1. 2017.



Pressman, R. *“Software Engineering: A Practitioner's Approach”*. 4th edition. McGraw-Hill, 2001.

That's it folks!



Thank you for your attention!