Cadastro Único de Clientes

Especificação de Objetivos e Requisitos

**CADU-EOR-001**

**Goiânia, Agosto 2015**

**Revisões**

|  |  |  |
| --- | --- | --- |
| ***Data*** | ***Descrição*** | ***Autor*** |
|  |  |  |

**Conteúdo**

**Figuras e Tabelas**

Incluir índice de figuras e tabelas mostradas no documento usando o mesmo modelo do índice de conteúdo.

**1. Introdução**

O software objeto desse projeto, denominado Cadastro Unico de Clientes (CADU), tem como objetivo o cadastro e a manutenção dos dados dos clientes/usuários em uma única base de dados, a fim de garantir integridade através da normatização das informações, da Agência Goiana de Regulação, Controle e Fiscalização de Serviços Públicos (AGR).

**1.1. Objetivos**

Este documento deve contemplar todos os casos de uso, requisitos (funcionais e não funcionais) que tem como propósito o auxílio no desenvolvimento do projeto de desenvolvimento do módulo CADU, do Sistema de Módulos de Integração, que está em desenvolvimento para a Agência Goiana de Regulação, Controle e Fiscalização de Serviços Públicos.

**1.2. Público Alvo**

Identificar o público alvo do documento, isto é, os órgãos e pessoas que poderão usá-lo.

**1.3. Organização do documento**

Descrever a organização e o conteúdo de cada seção do documento.

**2. Descrição do problema e do sistema**

Atualmente, os sistemas da AGR funcionam de forma independente, ou seja, não há comunicação das informações entre os softwares. Devido a esse modelo utilizado, vários problemas foram encontrados e levantados como críticos para a eficiência e a eficácia dos serviços prestados, entre eles: descentralização das informações, redundância e falta de normatização dos dados, baixa eficiência das áreas fins, falta de suporte para possíveis manutenções evolutivas, etc. Devido as etapas de trâmite do processo, algumas informações são genéricas para mais de uma área, criando a necessidade de uma única base de dados que permita a consulta por mais de um departamento, de forma que as informações sejam consistentes e íntegras, respeitando as regras do negócio. O projeto do CADU deve sanar todos esses problemas, permitindo a utilização do sistema tanto para fins internos quanto externos.

**2.1. Identificação e missão do Sistema**

Centralizar e disponibilizar de forma segura e integra os dados das Empresas fiscalizadas pela AGR.

**2.2. Domínio do problema e contexto de sua aplicação**

O cadastro de empresas na AGR é feita de forma presencial, após cadastrados em um certo sistema, os dados não são disponibilizados para outro sistema de forma integrada. O problema se agrava pela quantidade de processos e papéis na AGR, comprometendo a eficiência dos servidores ao cadastrar esses dados novamente, em cada área (fiscalização, financeira, jurídica, etc.) e em cada instância de trâmite do processo. O domínio do problema envolve todas as áreas fins da AGR, de forma que, centralizar esses dados para que possam ser utilizados em qualquer área irá agilizar o processo de cadastro e manutenção dos mesmos.

**2.3. Objetivos e Benefícios Esperados do Sistema**

|  |  |
| --- | --- |
| ***Objetivo a ser alcançado*** | ***Benefício adquirido*** |
| *Normatizar os dados dos sistemas.* | *Maior integridade dos dados, a fim de geração de informações consistentes para tomadas de decisão.* |
| *Cadastrar os dados das Empresas/Clientes da AGR em uma única base de dados.* | *Disponibilização dos dados para todas as áreas, a fim de agilizar o processo de cadastro e manutenção dos mesmos.* |
| *Informatização dos processos.* | *Diminuir a rotatividade de clientes presenciais na AGR e garantir uma resposta mais rápida aos serviços prestados.* |
| *Redução de custos* | *Redução de custos com papeis gastos na Agência.* |

**2.4. Características Essenciais do Sistema**

Descrever todas as características que devem ser consideradas para avaliar o sucesso do produto construído. Os detalhes necessários a essa avaliação devem ser especificados. Por exemplo, se a capacidade do sistema ser executado em diferentes plataformas computacionais é especificada como uma característica do sistema, deve ser definido um conjunto de plataformas que serão usadas para avaliar essa característica. O produto final do processo de desenvolvimento será validado com relação à missão, aos objetivos e às características descritas neste documento.

**2.5. Descrição dos interessados do sistema**

Descrever os perfis de cada interessado envolvido com o sistema.

|  |  |  |
| --- | --- | --- |
| ***Interessado(s)*** | ***Descrição*** | ***Necessidades Básicas*** |
| Servidores da AGR | Cadastrar e/ou consultar os dados das Empresas/Clientes no sistema. | O servidor necessita de um sistema que lhe permita fazer o cadastro e consulta dos dados de uma empresa/cliente que é fiscalizada pela AGR. |
| Usuários externos | Cadastrar e consultar os dados da sua empresa. | O usuário necessita de um sistema que lhe permita, através de login e senha, cadastrar os dados de sua empresa, verificar a situação dos dados e de sua situação perante a AGR. |

**2.6. Diagnóstico da Situação Atual**

Atualmente a AGR possui sistemas que operam independentemente os outros, ou seja, sistemas que não tem interoperabilidade entre si, são softwares legados desenvolvidos em uma linguagem já em desuso, que não permitem manutenções evolutivas e são passíveis de erros. O projeto do novo software tem como um dos objetivos substituir esses sistemas, a fim de utilizar nova linguagem de programação que permita manutenções dos diversos tipos, inclusive, para manter a melhor operabilidade e usabilidade dos usuários.

**2.6.1 Áreas de negócio atendidas**

Descrever as áreas de negócio (unidades organizacionais) dos clientes (ou da própria empresa, no caso de sistemas de uso interno), que utilizam o sistema atual. Para cada área de negócio identificada deverão ser indicados quais os processos de negócio realizados e como cada processo é executado (para quê, quando e como). Além disso, devem ser identificadas as informações manipuladas no processo.

**2.6.2 Áreas de negócio não atendidas**

Descrever os processos de negócio que deveriam ou poderiam ser realizados no sistema atual, mas não o são. Identificar que tipo de suporte deveria ser fornecido pelo sistema (o quê deveria ser feito, quando e como) e que tipo de informações seriam necessárias para esse fim.

**2.6.3 Integração com outros sistemas da empresa**

O projeto do Cadastro Único, tem como objetivo disponibilizar em uma base única os dados das empresas/clientes da AGR. Esse sistema estará sendo utilizado como suporte para o desenvolvimento de outros módulos, tais como, Módulo de Dívida Ativa, Módulo de Fiscalização de Transportes, Módulo Jurídico, portanto, os outros softwares que venham ser desenvolvidos e/ou comprados pela AGR devem se integrar com o CADU.

**2.6.4 Integração com outros sistemas de terceiros**

Ainda não existe nenhuma integração do Sistema CADU com sistemas de terceiros.

**2.6.5 Pontos negativos**

|  |  |  |
| --- | --- | --- |
| ***Risco*** | ***Descrição*** | ***Mitigação*** |
| Quantidade de acesso | Alta quantidade de acesso (interno/externo) causando lentidão no sistema. | Utilização de tecnologias (servidores, banda, banco de dados) que permitam uma alta quantidade de transações. |
| Baixa aceitação do usuário | Baixa aceitação devido ao conhecimento do sistema legado existente. | Treinamento no novo sistema. |
| Responsividade | Sistema não é responsivo, inviabilizando o uso em diversos dispositivos. | Utilização de uma interface/tecnologia que permita a responsividade. |
|  |  |  |

**2.6.6 Pontos positivos**

|  |  |
| --- | --- |
| ***Benefício*** | ***Descrição*** |
| Eficiência | O novo sistema trará eficiência no uso dos dados compartilhados em outros módulos. |
| Eficácia | Os dados serão integros o que irá permitir futura criação de relatórios que irão auxiliar na tomada de decisões. |
| Novas tecnologias | A utilização de técnicas e novas tecnologias, por serem bastante difundidas, vão permitir futuras manutenções e melhorias no sistema. Integração com sistemas de terceiros também são relacionados ao benefício. |
| Centralização das informações | As informações estarão disponíveis a qualquer sistema relacionado para que sejam feitas consultas/manutenção das informações. |
|  |  |

**2.6.7 Sugestões de melhorias**

|  |  |  |
| --- | --- | --- |
| ***Melhoria*** | ***Criticidade*** | ***Descrição*** |
| Responsividade |  | Implementar interface responsiva no sistema. |
| Acesso externo a usuários |  | Implementar/planejar estrutura para acesso de usuários externos a AGR. |
|  |  |  |

**3. Casos de Uso**

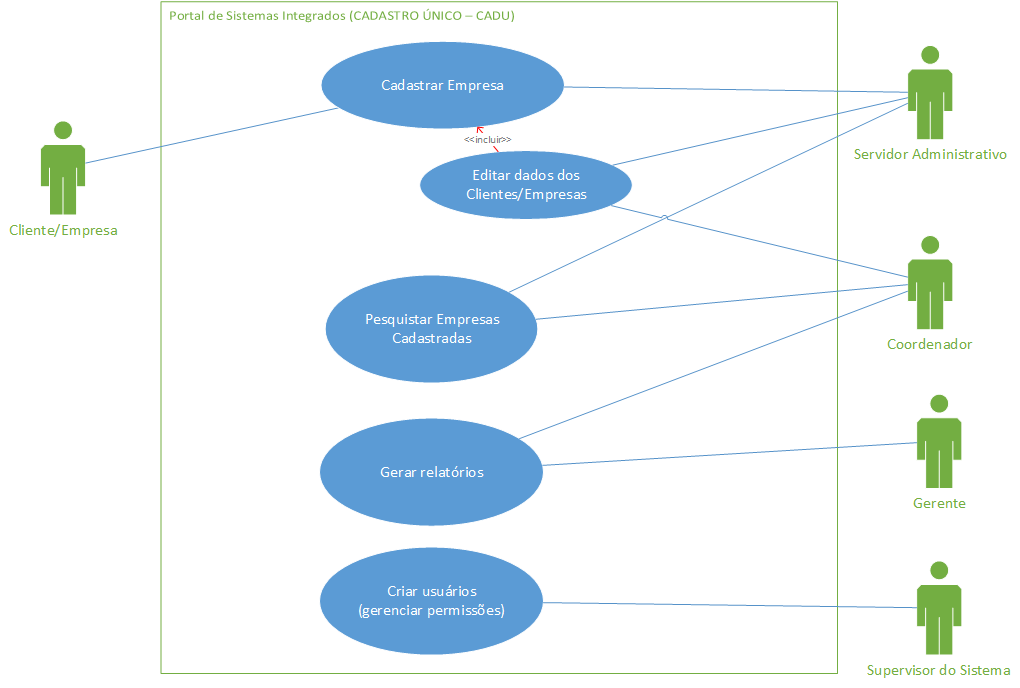
Nesta seção serão identificados os casos de uso do software que será desenvolvido, os atores neles envolvidos e os requisitos funcionais e não funcionais neles tratados.

**3.1. Atores**

1. Servidores administrativos
2. Coordenadores
3. Gerentes
4. Supervisores
5. Clientes/Empresas

**3.2. Diagramas de Caso de Uso e Lista de casos de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Descrição** | **Atores** | **Categoria** |
| CSU1 | Cadastrar dados das Empresas | Servidores Administrativos | Primário |
| CSU2 | Desbloquear Empresas/Usuários | Servidores Administrativos | Primário |
| CSU3 | Cadastrar os dados de sua Empresa | Clientes/Empresas | Primário |
| CSU4 | Criar relatórios gerenciais | Coordenadores/ Gerentes | Secundário |
| CSU5 | Gerenciar usuários no sistema | Supervisores | Primário |



**3.3 Descrição do Caso de Uso**

Identificar e descrever cada caso de uso listado anteriormente. Os casos de uso deverão ser descritos preferencialmente no formato de alto nível, mas dependendo da sua importância ou da sua complexidade eles poderão ser descritos no formato essencial expandido. Casos de uso essenciais são aqueles que não referenciam aspectos de soluções tecnológicas adotadas ao contrário dos casos de uso em formato concreto que referenciam as estruturas de interface da solução projetada para o sistema e outros aspectos tecnológicos do sistema.

**4. Requisitos e restrições funcionais (RFUN)**

Elaborar uma lista das funções que o software deve prover. Considerar não apenas as funções intrínsecas (essenciais) do software, mas também as de suporte, tais como administração da utilização do software, cópias de segurança, auditoria, e controle de acesso. Indicar para cada função uma referência, descrição, categoria, prioridade e casos de uso em que ela é tratada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ref.** | **Função** | **Categoria** | **Prioridade** | **Casos de Uso** |
| RFUN1 | Registrar venda de um produto | Evidente | Alta | CSUx, CSUy |
| RFUN2 | Atualizar o estoque do produto  vendido | Oculta | Média | CSUz |

**5. Requisitos e restrições não funcionais**

Elaborar uma lista de todos os requisitos não funcionais. Considerar requisitos de informação, de interface, de projeto, de arquitetura de software, de plataforma de hardware, de plataforma de software, de plataforma de comunicação, de desempenho, de disponibilidade, de segurança, de manutenibilidade, de portabilidade e de documentação. A lista poderá ser dividida por tipo de requisito, mas é importante que os requisitos tenham uma identificação única para que possam ser referenciados sem ambigüidades no futuro.

**5.1. Requisitos e restrições de informação (RINF)**

Elaborar uma lista de todas as necessidades de informação que o software não pode deixar de atender. Esta lista deverá ser classificada em informações cadastrais e informações gerenciais. Por exemplo, para um software de vendas existem, entre outras, as seguintes necessidades de informação:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.** | **Tipo** | **Descrição** | **Casos de Uso** |
| RINF1 | Cadastral | Vendedor: matrícula, nome, documentos  de identificação, salário, endereço e  telefone. | CSUx |
| RINF2 | Cadastral | Produto: código, descrição, unidade de  armazenamento, preço. | CUSy, CUSz |
| RINF3 | Gerencial | Relação dos dez vendedores que mais  venderam um determinado produto em  um determinado período, contendo a  matrícula e o nome do vendedor, o nome  do produto, a quantidade vendida, o valor  total da venda e a comissão paga. | CSUw |

Estes requisitos de informação são importantes para verificar a qualidade da modelagem de dados que for feita.

**5.2 Requisitos e restrições de interface Homem-Computador (RHIC)**

Definir todos os aspectos de Interface Homem Computador (IHC) incluindo: conteúdo de informações, fatores ergonômicos, dispositivos de interação, formato de apresentação, tipo de diálogo, e mecanismos de ajuda alocados a cada perfil/grupo/tarefa de usuário. Descrever, em particular, os requisitos de usabilidade para cada perfil/grupo/tarefa de usuário. Por exemplo, pode-se definir como requisito que as opções de *menu* do sistema tenham teclas de atalho associadas. É recomendável definir diagramas de interface (telas e relatórios) para as funções previstas para o software. Devem ser estabelecidas, no diagrama, as áreas da janela ou do relatório destinadas a cada tipo de informação. Existem diversos padrões para definição de interface de usuário. Exemplos destes padrões predefinidos são *Motif* e *Windows*. Esses padrões de interface definem desde diretivas para diagramação até os tipos de objetos de interface que podem ser adotados, com seus respectivos atributos e valores default.

No caso de se adotar algum padrão predefinido de interface, pode-se apenas referenciá- lo nesta seção, já que a diagramação de todas as janelas e relatórios deve seguir este padrão. Caso contrário, para cada área funcional identificada na diagramação, deve-se especificar a sua finalidade, suas dimensões e seu posicionamento relativo na janela ou relatório. Restrições sobre o tamanho e posição relativos entre as diversas áreas do diagrama também devem ser especificadas. Cada área do diagrama pode ser recursivamente subdivida em áreas menores. Nestes casos devem ser especificadas as mesmas informações definidas para as áreas principais, ou seja, a identificação, a finalidade, o tamanho e o posicionamento relativo.

Exemplos típicos de áreas funcionais de janelas incluem: área de mensagens; área de comandos; área de respostas a comandos; área de desenho; área de menus; e área de identificação da Janela. As áreas típicas de composição de um relatório são: cabeçalhos, corpo, linhas de detalhe, linhas de totalização, e rodapé.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RIHC1 | Para facilitar a usabilidade na transação de venda, pede-se que a tela de vendas tenha uma fonte (tipo e  tamanho de letra) que permita uma fácil visualização a uma distância de 2 metros do monitor porque desta  forma o cliente poderá visualizar as informações da venda da sua posição. | Todos |

**5.3 Requisitos de Interface Externa (RIEX)**

Identificar e descrever as interfaces com outros softwares/sistemas que o software deverá prover. Por exemplo, um software comercial deve gerar informações para o Sistema de Arrecadação da Secretaria da Fazenda Estadual. O formato dessas informações e o protocolo de envio são definidos pela própria secretaria, e atender essas definições é um requisito do software.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RIEX1 | O software do sistema de vendas deverá gerar um arquivo SINTEGRA conforme Legislação do Convênio ICMS 57/95 atualizado até 69/02, incluindo as alterações posteriores, para que seja enviado para a Secretaria da Fazenda trimestralmente. | CSUn |

**5.4 Requisitos e Restrições de Projeto (RPRO)**

Nesta seção serão especificados todos os requisitos e restrições associados a condução do projeto de desenvolvimento e que podem limitar ou definir ações que serão executadas.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RPRO1 | O cliente solicitou que o módulo de contabilidade fosse entregue até o dia 10 do último mês do ano corrente (10/12/05), para testes em ambiente real. | CSUi, CSUj, CSUk |

**5.5 Requisitos e restrições de arquitetura de software (RARQ)**

Se o software tiver de ser desenvolvido em uma arquitetura específica, então essa arquitetura deverá ser descrita.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RARQ1 | O software deverá ser desenvolvido com uma arquitetura de camadas que permita isolar as funcionalidades ligadas ao negócio das  funcionalidades relacionadas com a interface homem-computador. | Todos |

**5.6 Requisitos e restrições de plataforma de hardware (RPHW)**

Identificar e descrever requisitos e restrições relacionadas com a plataforma de hardware que será utilizada pelo software:

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RPHW1 | O software deverá ser capaz de rodar em um Servidor com processador Intel xSeries (IBM) |  |

**5.7 Requisitos e restrições de plataforma de software (RPSW)**

Se o software tiver que ser executado em plataformas de software específicas, essas plataformas de software deverão ser definidas:

Sistema Operacional: identificar e descrever o sistema operacional em que o software deverá ser executado;

Softwares Básicos: identificar SGBD, linguagem de programação, ferramentas CASE e outros. Se houver mais de uma plataforma de software, deve-se especificar qual a plataforma principal e em que situações as outras plataformas podem ser utilizadas.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RPSW1 | O software deverá ser desenvolvido com a ferramenta CASE XYZ gerando código Java. A justificativa para esta restrição é que esta é plataforma-padrão adotada  pela empresa. | Todos |

**5.8 Requisitos e restrições de desempenho (RDES)**

Identificar e descrever os requisitos e restrições de desempenho do software.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RDES1 | O ambiente onde o software rodará deverá permitir pelo menos três usuários acessando o banco de dados sem queda de velocidade. | Todos |
| RDES2 | O tempo de resposta máximo permitido para transações on-line é de 5 segundos | CSUx,... |
| RDES3 | O software deverá ser capaz de atender até dez transações simultâneas da função “Registrar Venda”. | CSUz |

**5.9 Requisitos e restrições de disponibilidade (RDIS)**

Especificar os requisitos de disponibilidade necessários para o software de uma forma

global:

Período de disponibilidade: horário comercial, 24 horas por dia, etc.

Período máximo para recuperação do software em caso de falha. Devem ser definidos os tipos de falha e a tolerância aceitável para cada tipo de falha. Os tipos de falha podem ser definidos em função dos requisitos funcionais e de dados, mas não se restringem a estes. Por exemplo: a função “Registrar Venda” deve ter um tempo para recuperação de falha de no máximo uma hora (o que significa que esta função não poderá ficar mais do que uma hora indisponível para o usuário em nenhuma circunstância).

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RDIS1 | O software deverá estar disponível 24 horas por dia | Todos |

**5.10 Requisitos e restrições de segurança (RSEG)**

Especificar os requisitos de segurança necessários para controle de acesso ao software. Definir a necessidade de:

Verificação de senha;

Criptografia de dados;

Registro das operações efetuadas;

Habilitação de funções por perfil de usuário;

Acesso seletivo aos dados e funções.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RSEG1 | O software deverá solicitar autorização do supervisor para  excluir uma venda já registrada. | CSUv |
| RSEG2 | Todos os dados em arquivos de exportação deverão ser  criptografados. | CSUx |
| RSEG3 | Para qualquer atualização efetuada, deverá ser registrado o usuário que realizou a operação, bem como a data e hora. | CSUn,  CSUm,... |
| RSEG4 | Todo usuário do software deverá ser associado a um perfil que define as funcionalidades que poderão ser utilizadas por ele. | Todos |

|  |
| --- |
|  |

**5.11 Requisitos e restrições de manutenibilidade (RMAN)**

Especificar os requisitos que visam facilitar a manutenção posterior do software, tais

como:

Requisitos de reutilização (exemplo: uso de implementação orientada a objetos; bibliotecas de classes e padrões de projeto);

Requisitos de modularização (exemplo: valores para métricas de acoplamento entre módulos; máximo de pontos de função por módulo);

Requisitos de configuração (exemplo: regras para controle de versões);

Requisitos de documentação (exemplo: documentação de programa)

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RMAN1 | O projeto das responsabilidades de cada classe de objetos deverá seguir os padrões GRASP sugeridas no livro Utilizando UML e Padrões de Craig Larman [1]. | Todos |
| RMAN2 | Todo programa deve estar documentado de acordo com as orientações contidas na Norma de Documentação de Programas da empresa [2] | Todos |

**5.12 Requisitos e restrições de portabilidade (RPOR)**

Identificar as diversas plataformas de software e hardware com as quais o software deve ser compatível. Devem ser consideradas tanto plataformas de desenvolvimento como plataformas de produção. Outros exemplos de requisitos de portabilidade são:

Percentual de componentes que podem ter código dependente da plataforma hospedeira;

Percentual de código que pode ser dependente da plataforma hospedeira; Uso de uma linguagem reconhecidamente portável.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RPOR1 | O software deverá ser capaz de rodar tanto em computadores Desktop com Windows 98 ou superior ou com Linux. Deverá ser capaz de rodar também em Palmtops com Windows CE. | CSUx, CSUy |
| RPOR2 | O software deverá permitir a utilização dos seguintes Bancos de dados: SQL Server, MySql ou Oracle | Todos |

**5.13 Requisitos de documentação (RDOC)**

Especificar os requisitos de documentação do produto de software que será desenvolvido.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RDOC1 | Junto com o script de instalação do produto deve ser colocado um manual de instalação do mesmo | Não se aplica |
| RDOC2 | O software deve prover ajuda on-line sensível ao contexto em todas as suas telas | Todos |

**6. Requisitos Futuros (RFUT)**

Descrever os requisitos que poderão ser especificados em uma nova versão do produto.

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Descrição** | **Casos de Uso** |
| RFU1 | Em um futuro próximo o software de atendimento de clientes deverá ser integrado com o software do sistema de faturamento para que o atendente possa  identificar o perfil de negócios do cliente | CSUx |

**7. Referências cruzadas complementares**

Nesta seção são colocadas algumas referências cruzadas que podem ajudar o rastreamento futuro dos requisitos. Estes mapeamentos podem ser feitos em forma de matrizes de rastreabilidade como mostram os exemplos a seguir:

|  |  |
| --- | --- |
| **Requisitos Funcionais** | **Requisitos não funcionais vinculados** |
| *Colocar a identificação do*  *requisito funcional* | *Colocar a identificação do requisito não funcional vinculado* |

|  |  |
| --- | --- |
| **Caso de Uso** | **Requisitos funcionais vinculados** |
| *Colocar a identificação do*  *caso de uso* | *Colocar a identificação do requisito vinculado* |

|  |  |
| --- | --- |
| **Requisito Funcional** | **Origem do Requisito** |
| *Colocar a identificação do*  *requisito funcional* | *Colocar a origem do requisito. Pode ser uma entrevista, um questionário, ou outra técnica qualquer de elicitação de requisito aplicada.* |

|  |  |
| --- | --- |
| **Requisito Não Funcional** | **Origem do Requisito** |
| *Colocar a identificação do requisito não funcional* | *Colocar a origem do requisito. Pode ser uma entrevista, um questionário, ou outra técnica qualquer de elicitação de requisito aplicada.* |

|  |  |
| --- | --- |
| **Caso de Uso** | **Origem do caso de uso** |
| *Colocar a identificação do*  *requisito funcional* | *Colocar a origem do caso de uso. Pode ser uma entrevista, um questionário, ou outra técnica qualquer de elicitação de requisito aplicada.* |

**8. Aprovação Formal**

O EOR deve ser datado e assinado de acordo com a definição contida no MPD. O documento aprovado se torna uma *baseline* do projeto que passa a ser controlada pela Gerência de Configuração de Software.

**10. Anexos**

Anexar ao EOR documentos considerados importantes para sua compreensão. Por exemplo, documentos e leis usadas na especificação dos requisitos podem ser colocados nos anexos para facilitar o acesso do leitor às fontes dos requisitos do software. Cada anexo deverá receber uma folha de rosto indicando seu conteúdo.