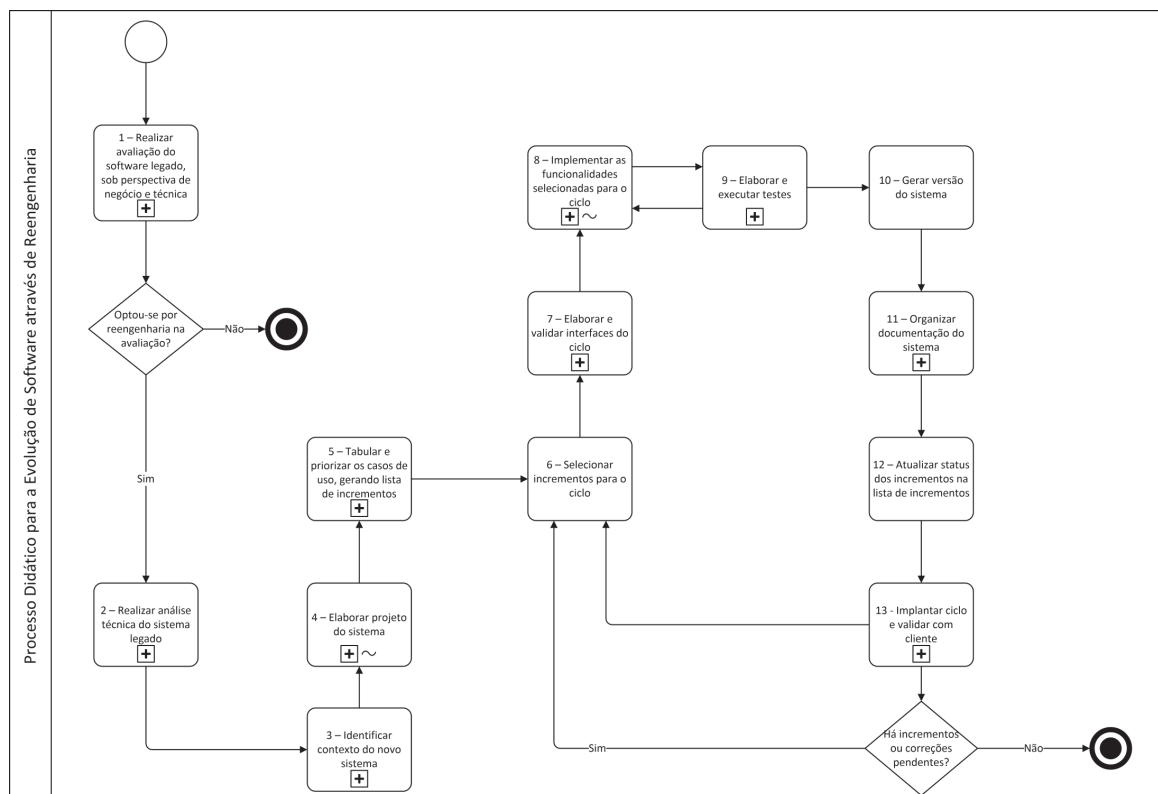


## 4 Definição do processo

A definição do processo de reengenharia envolve a criação de um conjunto de atividades, sendo que o processo inicialmente possui uma abordagem baseada no modelo cascata, para a avaliação do software legado sob perspectiva de negócio e técnica e para a etapa de reengenharia do sistema. A partir daí optou-se por uma abordagem incremental, a partir da elaboração e priorização da lista de incrementos que serão alocados aos ciclos de desenvolvimento seguintes. A utilização desta abordagem incremental justifica-se pela rápida entrega de software funcional ao cliente, o que permite que as funcionalidades sejam aos poucos migradas para o novo sistema, além da realização de *feedback* por parte dos usuários, onde as solicitações de melhorias ocorrem ainda na etapa de desenvolvimento. Adicionalmente, optou-se pela modelagem do sistema evoluído a partir da abordagem orientada a objetos, onde a realidade a ser modelada está mais próxima do modelo, através do conceito de objetos (BEZERRA, 2007). A figura 13 define o modelo de processo de reengenharia proposto por este trabalho.

Figura 13 – Processo de reengenharia.



## 4.1 Detalhamento do processo

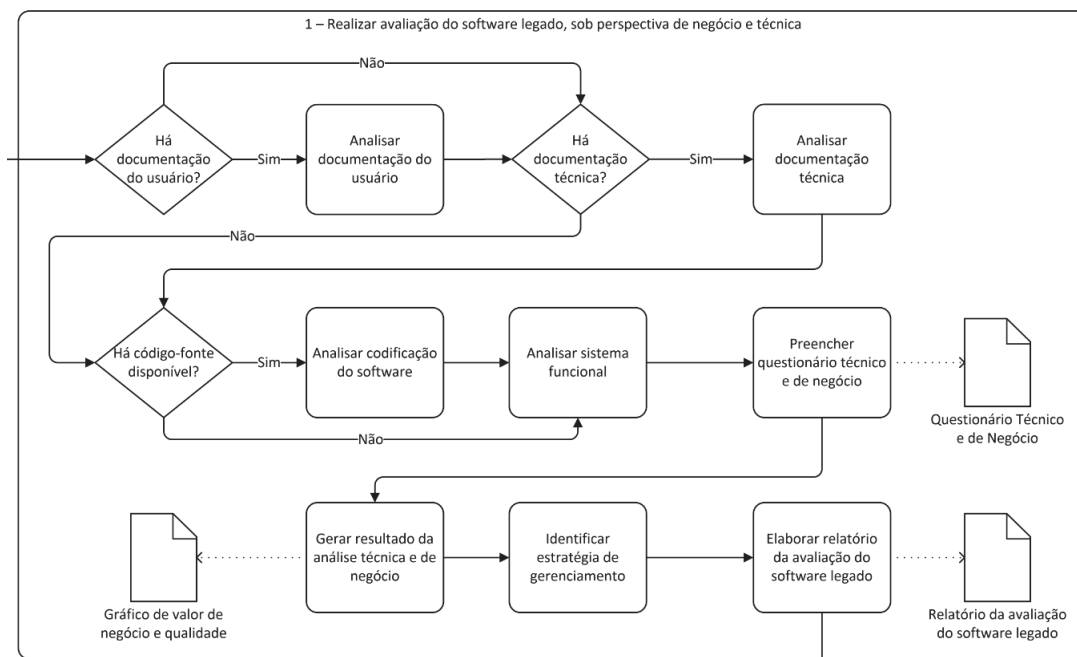
Nesta seção do trabalho cada atividade do processo é descrita de forma detalhada, juntamente com as entradas e saídas, facilitando o entendimento do funcionamento do processo e ilustrando os artefatos elaborados para o suporte do mesmo. Adicionalmente, as técnicas de Evolução de Software são referenciadas em cada atividade as quais estão relacionadas.

### Etapa 1 - Avaliação do software legado

A primeira etapa do processo envolve a avaliação do software legado e a definição da estratégia de manutenção para o sistema.

#### 1 - Realizar avaliação do software legado, sob perspectiva de negócio e técnica

Figura 14 – Subprocesso Expandido 1.



Esta atividade tem por objetivo avaliar o software legado, sob duas perspectivas: a perspectiva de negócio e a perspectiva técnica. Em relação à perspectiva de negócio, avalia-se a real necessidade do sistema para a organização, e sob a perspectiva técnica examina-se a qualidade do software e sua compatibilidade com as tecnologias mais recentes (SOMMERVILLE, 2011).

De acordo com Sommerville (2011), a partir do cruzamento das perspectivas de negócio e técnica, obtém-se quatro grupos distintos de sistemas: (1) Baixa qualidade e

baixo valor de negócio, (2) Baixa qualidade e alto valor de negócio, (3) Alta qualidade e baixo valor de negócio e (4) Alta qualidade e alto valor de negócio. Cada um desses grupos possui correspondência com uma das quatro opções estratégicas para a manutenção do software que estão descritas na [seção 2.4](#): (1) Descartar completamente o sistema, (2) Deixar o sistema inalterado e seguir a manutenção regular, (3) Reestruturar o sistema ou (4) Substituir o sistema por um novo sistema. Abaixo encontra-se a descrição das características do sistema a partir das possíveis combinações de qualidade e valor de negócio, juntamente com a opção estratégica recomendada para cada caso:

1. *Baixa qualidade e baixo valor de negócio.* Devido a essa característica, o uso do software pelo cliente possui pouco retorno, e sua manutenção é cara, o que faz com que sistemas que se enquadram nessa categoria sejam descartados.
2. *Baixa qualidade e alto valor de negócio.* Sistemas que se enquadram nessa categoria são importantes para a organização e não devem ser descartados. A baixa qualidade indica que os custos de manutenção são altos e que o sistema deve ser reestruturado para ampliar a sua qualidade, ou então ser substituído por uma solução de prateleira, caso a mesma esteja disponível.
3. *Alta qualidade e baixo valor de negócio.* Sistemas com essa característica não contribuem de forma efetiva para a organização, porém podem ser submetidos a manutenções regulares se as mudanças necessárias não forem de alto custo, caso contrário o mesmo deve ser descartado.
4. *Alta qualidade e alto valor de negócio.* Sistemas com essas características devem permanecer em operação e a manutenção regular deve ser mantida.

É importante salientar que o processo somente prosseguirá caso a opção escolhida seja a reengenharia como estratégia de manutenção do sistema legado.

#### Entradas

- Documentação técnica do sistema legado (caso ela exista);
- Documentação de usuário do sistema legado (caso ela exista);
- Software legado (executável ou códigos-fonte, caso estejam disponíveis);

#### Atividades

A partir da análise das informações presentes na documentação técnica e de usuário e do sistema legado, avalia-se o software, a partir de um conjunto de questões previamente definidas em função da perspectiva técnica e da perspectiva de negócio. Tais

questões foram elaboradas a partir da proposta de gerenciamento de sistemas legados de [Sommerville \(2011\)](#) e estão organizadas em dois conjuntos, um de questões relativas ao valor de negócio e outro em função da qualidade do sistema. Cada conjunto de questões é composto pelo número sequencial da questão, a descrição da questão, o avaliador da questão (responsável ou o grupo de responsáveis por respondê-la) e pela resposta (sim ou não), de forma que a resposta seja quantificável. As tabelas abaixo representam as questões, sob a perspectiva de valor de negócio ([Tabela 1](#)) e sob a perspectiva de qualidade ([Tabela 2](#)).

Tabela 1 – Questões para avaliação sob perspectiva de valor de negócio

Nº	Questão	Avaliador	Resposta
1	Considerando o contexto onde o software está inserido, o sistema é usado com frequência?	Usuário	
2	Os processos de negócio do sistema estão atualizados?	Gerente	
3	O sistema é confiável (possui pouquíssimos defeitos ou falhas)?	Usuário, gerente	
4	As saídas do sistema são utilizadas? Há importância para o usuário?	Usuário, gerente	

As questões são respondidas com o apoio de uma planilha eletrônica. De acordo com as respostas das questões, a planilha representa em um gráfico a situação do sistema, em função da perspectiva de valor de negócio e da perspectiva de qualidade. Para cada possível localização do sistema em um dos quadrantes do gráfico, há uma estratégia recomendada para a manutenção desse sistema. Tal gráfico foi construído a partir de uma adaptação do gráfico de avaliação de sistemas legados proposto por [Sommerville \(2011\)](#), que classifica o sistema em função da perspectiva de valor de negócio e qualidade. A figura 15 ilustra o gráfico gerado pela planilha.

Adicionalmente, a partir da análise da documentação técnica e de usuário do sistema e da codificação, deve-se identificar as seguintes informações adicionais:

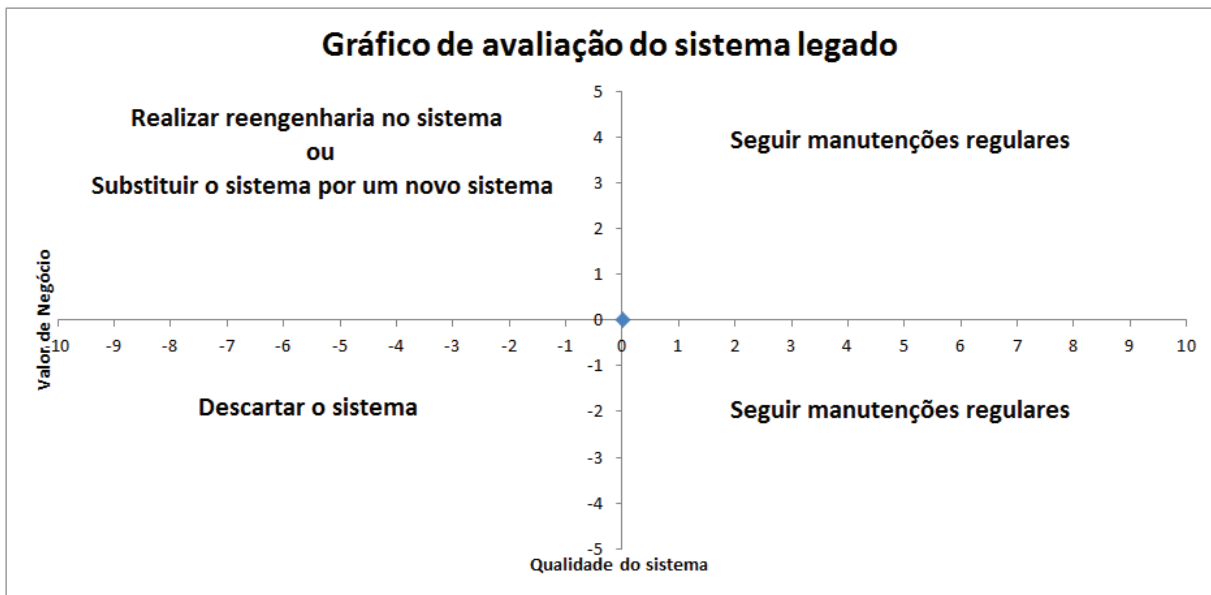
- Identificar a linguagem de programação em que o software foi desenvolvido;
- Identificar o propósito do sistema (a que se destina?);
- Identificar o paradigma do software (estruturado ou orientado a objetos?);

Todas essas informações levantadas nessa atividade deverão ser organizadas em um relatório, onde deverão estar presentes as questões do questionário técnico e de negócio preenchidas, o gráfico indicando a estratégia compatível com as respostas dos questionários, as informações adicionais e a decisão de estratégia de manutenção do software

Tabela 2 – Questões para avaliação sob perspectiva de qualidade

Nº	Questão	Avaliador	Resposta
1	O fornecedor do sistema ainda existe e dá suporte para o mesmo?	Gerente	
2	O sistema é compatível com o hardware e sistema operacional mais recentes?	Área técnica	
3	O sistema foi submetido a várias modificações durante o seu uso?	Área técnica, gerente	
4	O desempenho do sistema é adequado?	Usuário, gerente	
5	O software opera independentemente de manutenção de hardware legado?	Área técnica	
6	O software opera independentemente de manutenção de softwares de apoio pagos?	Área técnica	
7	O sistema possui custo baixo de manutenção?	Área técnica	
8	O sistema é compatível com diversos sistemas operacionais (windows, linux, etc.)?	Área técnica	
9	O código-fonte do sistema legado é de fácil compreensão?	Área técnica	
10	Há documentação do software legado disponível?	Área técnica	
11	Os dados são armazenados em um banco de dados?	Área técnica	
12	O banco de dados está normalizado?	Área técnica	
13	É adotado um padrão para nomenclatura das tabelas e campos do Banco de Dados?	Área técnica	
14	São realizadas validações nos campos (Ex.: Datas, CPF, e-mail, etc.)?	Área técnica, usuário	
15	O sistema exige que campos obrigatórios sejam preenchidos?	Área técnica, usuário	
16	A linguagem de programação do sistema legado ainda é utilizada no mercado de trabalho?	Área técnica	
17	Existem registros de execução de testes do sistema?	Área técnica	
18	A interface do sistema é implementada em modo gráfico?	Área técnica	
19	Existem profissionais com habilidades necessárias para o suporte da aplicação?	Área técnica	
20	O sistema redimensiona as informações ao maximizar a tela?	Área técnica	
21	Ao solicitar o armazenamento de dados de um formulário, o usuário é sempre informado se os mesmos foram gravados com sucesso ou se ocorreu uma falha?	Área técnica, usuário	
22	O sistema possui um padrão de interfaces? Esse padrão é sempre seguido?	Área técnica	
23	O sistema permite a navegação através do teclado?	Área técnica	

Figura 15 – Gráfico de avaliação do sistema legado.



legado escolhida pela equipe ((1) Descartar completamente o sistema, (2) Deixar o sistema inalterado e seguir a manutenção regular, (3) Reestruturar o sistema ou (4) Substituir o sistema por um novo sistema).

#### Saídas

- Relatório da avaliação do software legado.

## Etapa 2 - Reengenharia

É nesta etapa que ocorre as atividades que compreendem a reengenharia do sistema legado. É composta por 3 grupos de macro atividades: o de análise do software legado, o de análise e projeto do novo software e o de implementação e implantação do novo sistema.

Em relação a macro atividade de análise do software legado, a mesma baseia-se no conceito da engenharia reversa, cujo objetivo é extrair informações relacionadas a estrutura do sistema legado, que facilitam a documentação do sistema (SOMMERVILLE, 2007). Dentro desta macro atividade há a atividade de análise técnica do software legado, que tem por objetivo o levantamento de informações relacionadas à estrutura do sistema.

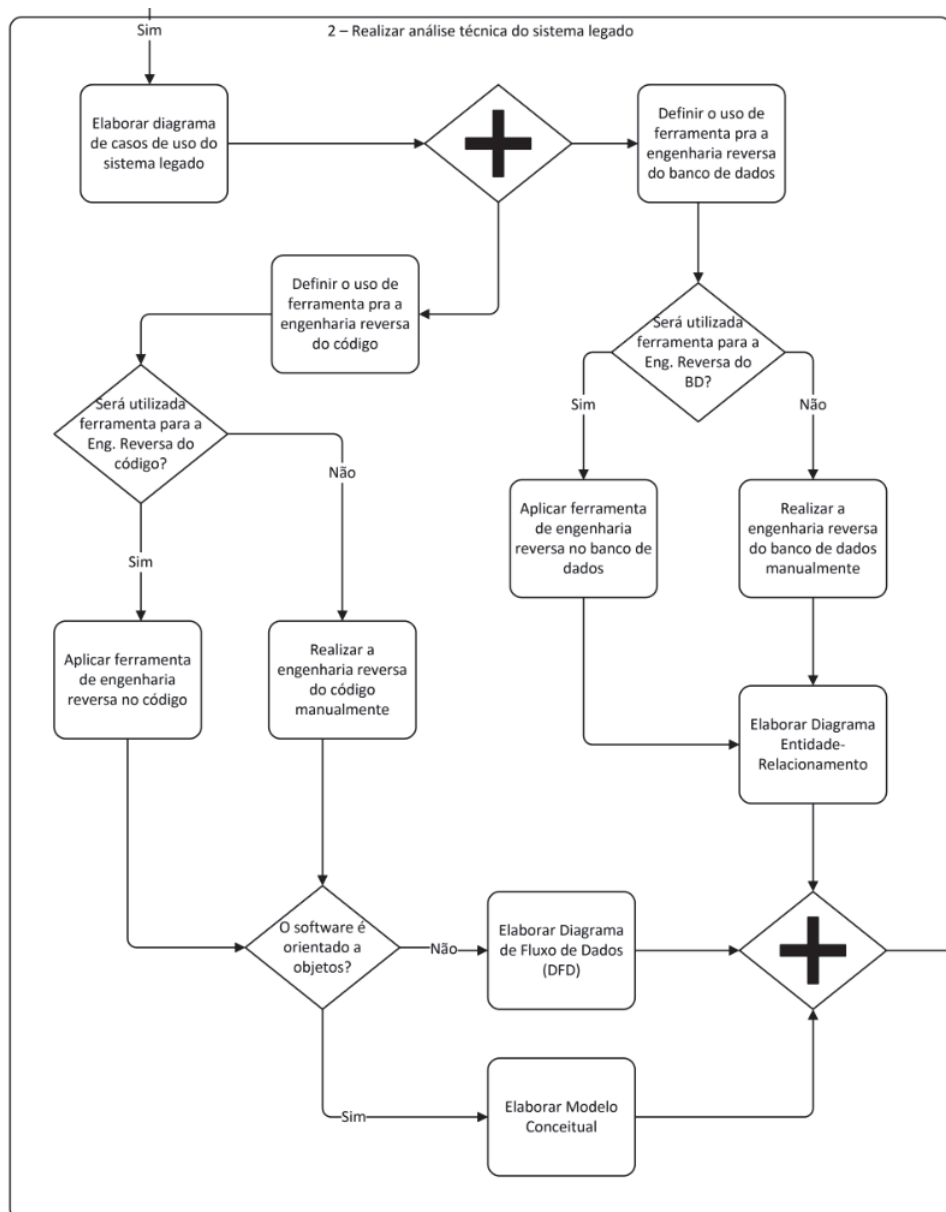
A macro atividade de análise e projeto do novo software tem por objetivo o planejamento e a identificação do contexto do novo sistema, a elaboração do projeto do novo sistema e a elaboração da lista de incrementos, através da tabulação e priorização dos

casos de uso do novo sistema. [Pfleeger \(2004\)](#) define em seu processo de reengenharia que é necessário que a especificação e o projeto do sistema sejam atualizados.

A terceira macro atividade envolve o desenvolvimento das funcionalidades do novo sistema a partir da construção através de uma abordagem incremental.

## 2 - Realizar análise técnica do software legado

Figura 16 – Subprocesso Expandido 2.



A análise técnica tem por objetivo levantar informações relacionadas a estrutura do sistema legado, de forma mais aprofundada, envolvendo a elaboração do diagrama de casos de uso do sistema e a obtenção de diagramas através de engenharia reversa.

A engenharia reversa é realizada a partir do uso de ferramentas ou de forma manual, caso não existam ferramentas para a linguagem de programação do sistema legado. As informações são organizadas de forma que seja possível a sua manipulação, e a partir de sua representação pode-se identificar inconsistências e violações de padrões de projeto (PFLEEGER, 2004).

### Entradas

- Documentação do sistema legado (caso ela exista);
- Software legado (executável ou códigos-fonte, caso estejam disponíveis);
- Informações relativas ao contexto do software legado, presentes no relatório da avaliação do software legado(saída da atividade 1);

### Atividades

- Elaborar o diagrama de casos de uso do sistema legado;
- Definir se a atividade de engenharia reversa do código e do banco de dados será apoiada por alguma ferramenta ou se será realizada de forma manual;
- Realizar a engenharia reversa do código e do banco de dados;
- A partir das informações obtidas com a atividade de engenharia reversa, elaborar o modelo conceitual do software legado<sup>1</sup> ou diagrama de fluxo de dados do software legado<sup>2</sup>;
- A partir das informações obtidas com a engenharia reversa do banco de dados, elaborar o modelo entidade-relacionamento do software legado (caso o software armazene as informações em banco de dados);

### Saídas

A atividade gera os seguintes artefatos:

- Diagrama de casos de uso do sistema, identificando os atores do sistema e como se relacionam com as funcionalidades;

<sup>1</sup> O modelo conceitual representa as informações que o sistema gerencia e que o usuário identifica, representada em notação UML (WAZLAWICK, 2011). Caso seja identificado anteriormente que o sistema legado foi modelado utilizando-se a abordagem orientada a objetos, deve-se representar os atributos, conceitos e associações do sistema através do modelo conceitual.

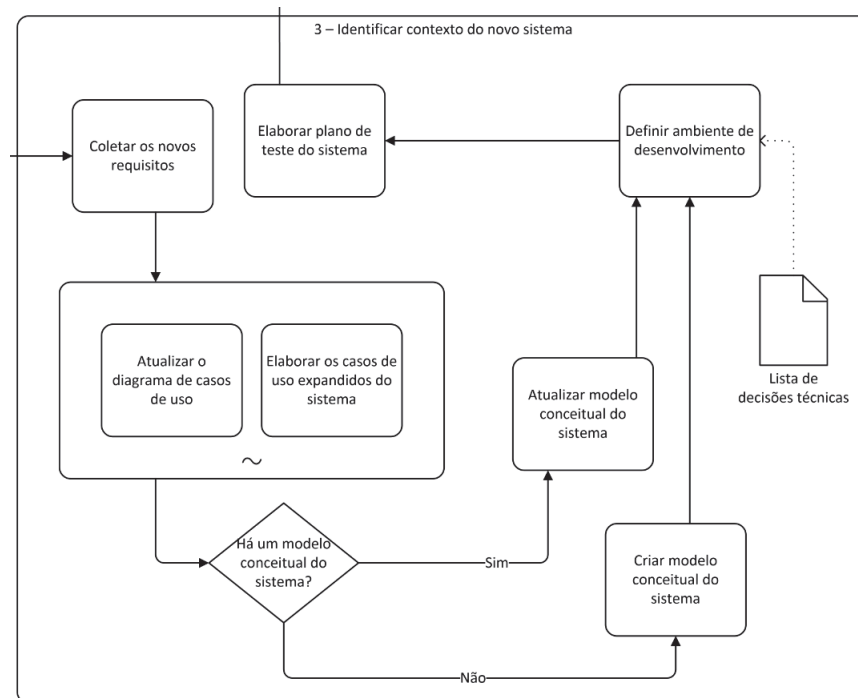
<sup>2</sup> Através do diagrama de fluxo de dados do sistema, obtém-se uma visão dos fluxos e requisitos de um sistema, através da visão entrada-processo-saída, onde os objetos de dados entram no sistema, sofrem transformação através de processamento e saem do software (PRESSMAN, 2011). O uso do diagrama de fluxo de dados é mais frequente em sistemas que possuem paradigma estruturado, como forma de representação das transições e processamento dos dados do sistema.



- Identificação da forma com que os dados são armazenados no sistema (arquivos, banco de dados, etc.), juntamente com o modelo entidade-relacionamento do sistema legado (caso o software armazene as informações em banco de dados);
- Modelo conceitual ou Diagrama de Fluxo de Dados do Software legado.

### 3 - Identificar contexto do novo sistema

Figura 17 – Subprocesso Expandido 3.



A etapa a seguir envolve a identificação do contexto do novo sistema, elaborado a partir das informações obtidas através da análise do sistema legado e das mudanças solicitadas pelo cliente (caso elas existam).

#### Entradas

- Informações presentes na análise do software legado, produzidas na atividade 2;
- Solicitações de mudança;

#### Atividades

- Coletar os novos requisitos (caso eles existam);
- Atualizar o diagrama de casos de uso do sistema (inclusão de novos casos de uso e/ou alterações nos casos de uso já existentes);

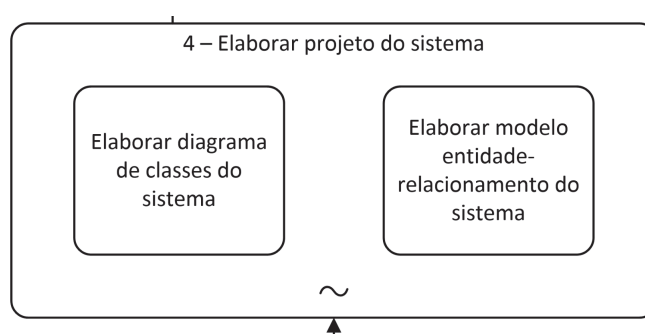
- Elaborar os casos de uso expandidos do sistema;
- Atualização do modelo conceitual (caso o sistema legado seja de paradigma orientado a objetos) ou elaboração do modelo conceitual do sistema (caso o sistema legado seja de paradigma estruturado);
- Definir o ambiente de desenvolvimento (escolha de linguagem de programação, padrões de projeto, ferramentas a serem utilizadas, banco de dados, forma de migração das informações do sistema antigo para o novo sistema, IDE de desenvolvimento, etc.) e registrá-lo em uma lista das decisões técnicas tomadas pela equipe;
- Definir do plano de teste (definição estratégia de teste do novo sistema, seleção das funcionalidades a serem testadas e funcionalidades que não serão testadas, escolha de ferramentas para o teste, definição de artefatos gerados pelos testes);

#### Saídas

- Diagrama de casos de uso do novo sistema;
- Casos de uso expandidos do novo sistema;
- Modelo conceitual do novo sistema;
- Lista de decisões técnicas (linguagem de programação escolhida, padrões de projeto, ferramentas a serem utilizadas, banco de dados, forma de migração das informações do sistema antigo para o novo sistema, IDE de desenvolvimento, tempo de duração de cada ciclo de desenvolvimento, etc.);
- Plano de teste;

## 4 - Elaborar projeto do sistema

Figura 18 – Subprocesso Expandido 4.



A partir das informações obtidas na etapa de identificação do contexto do novo sistema, é necessária a realização do projeto do novo sistema, que será elaborado inicialmente de forma ampla e será modificado e complementado de acordo com a necessidade durante os ciclos de desenvolvimento.

#### Entradas

- Diagrama de casos de uso do novo sistema (saída da atividade 3);
- Casos de uso expandidos do novo sistema (saída da atividade 3);
- Modelo conceitual do novo sistema (saída da atividade 3);
- Lista de decisões técnicas (saída da atividade 3);
- Plano de teste (saída da atividade 3);
- Modelo entidade-relacionamento do sistema legado (caso o sistema legado utilize tecnologia de banco de dados para o armazenamento das informações - saída da atividade 2);

#### Atividades

O projeto do sistema envolve a elaboração do diagrama de classes, que representa a estrutura do sistema, e do modelo entidade-relacionamento, sendo que este deve também passar pela reengenharia de dados ([SOMMERVILLE, 2007](#)), de forma a refletir as mudanças do software, corrigindo erros, redundâncias, etc.

#### Saídas

- Diagrama de classes do sistema;
- Modelo entidade-relacionamento do sistema;

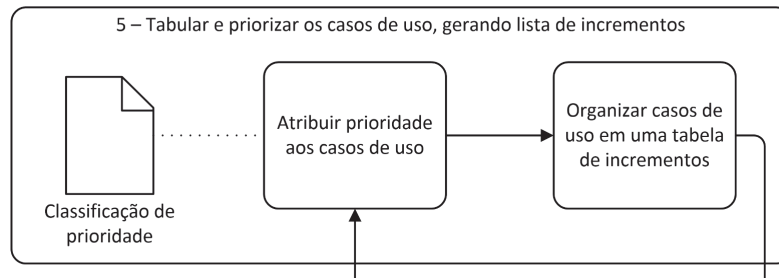
### 5 - Tabular e priorizar os casos de uso, gerando lista de incrementos

Esta atividade tem por objetivo definir uma lista de incrementos a serem realizados, a partir da descrição dos casos de uso do novo sistema.

#### Entradas

- Diagrama de casos de uso do novo sistema (saída da atividade 3);
- Casos de uso expandidos do novo sistema (saída da atividade 3);

Figura 19 – Subprocesso Expandido 5.



- Diagrama de classes do sistema (saída da atividade 4);
- Modelo entidade-relacionamento do sistema (saída da atividade 4);

#### Atividades

A partir das informações das expansões dos casos de uso do sistema e do projeto, a equipe de desenvolvimento do sistema atribui a cada caso de uso sua prioridade, seguindo o método descrito por [Bezerra \(2007\)](#), que relaciona o risco de desenvolvimento e a prioridade do usuário:

1. *Risco alto e prioridade alta*, são casos de uso críticos e devem ser implementados o quanto antes;
2. *Risco alto e prioridade baixa*, são casos de uso de alto risco, porém sua necessidade de implementação é baixa e deve-se negociar com o cliente a necessidade de implementação;
3. *Risco baixo e prioridade alta*, possuem risco baixo, porém devem ser priorizados após os de alto risco;
4. *Risco baixo e prioridade baixa*, são os casos de uso de menor importância e os passíveis de corte em caso de atrasos no desenvolvimento.

Os casos de uso são então organizados em uma tabela de incrementos, que serão posteriormente selecionados para a realização dos ciclos, conforme ilustrado na tabela 3. A tabela é composta pelo número sequencial do incremento, a descrição do incremento, a prioridade de implementação e o status atual do incremento. Nesta tabela também serão registradas as correções necessárias após a conclusão de um ciclo, a partir da verificação do cliente.

Tabela 3 – Modelo de lista de incrementos.

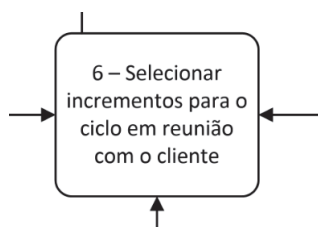
Nº	Descrição	Prioridade	Status
01	Cadastro de Clientes	03	Em aberto
02	Emissão de Nota Fiscal	01	Realizado no ciclo 1
03	Cadastro de Produtos	03	Em implementação

## Saídas

- Lista de incrementos;

## 6 - Selecionar incrementos para o ciclo

Figura 20 – Atividade 6.



Esta atividade tem por objetivo selecionar os incrementos que serão realizados durante o ciclo de desenvolvimento, através de uma reunião com o cliente.

## Entradas

- Diagrama de casos de uso do novo sistema (saída da atividade 3);
- Casos de uso expandidos do novo sistema (saída da atividade 3);
- Diagrama de classes do sistema (saída da atividade 4);
- Modelo entidade-relacionamento do sistema (saída da atividade 4);
- Lista de incrementos (saída da atividade 5);

## Atividades

A seleção de incrementos para o ciclo é realizada através de uma reunião entre o cliente e a equipe de desenvolvimento. A definição de quais incrementos serão selecionados para o ciclo é determinada pela equipe, juntamente com o cliente e levando em consideração:

- O tempo de duração do ciclo de desenvolvimento;

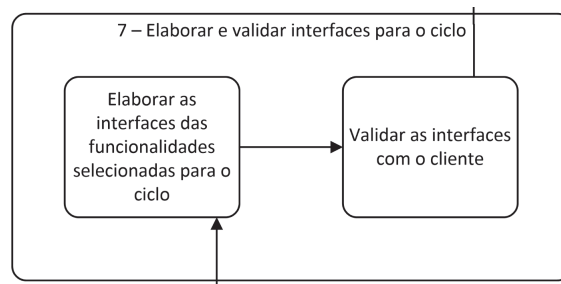
- A priorização dos incrementos;
- A dependência entre os incrementos;
- As necessidades do cliente;
- Os incrementos de correção (caso existam) e sua prioridade;

#### Saídas

- Definição de incrementos a serem realizados no ciclo de desenvolvimento;

### 7 - Elaborar e validar interfaces do ciclo

Figura 21 – Subprocesso Expandido 7.



Nesta etapa do processo são elaboradas as interfaces das funcionalidades selecionadas para o ciclo, observando mudanças solicitadas pelo cliente e também recomendações para a elaboração de interfaces.

#### Entradas

- Interfaces anteriores do sistema;
- Diagrama de casos de uso do novo sistema (saída da atividade 3);
- Casos de uso expandidos do novo sistema (saída da atividade 3);
- Normas e recomendações para a elaboração de interfaces;
- Incrementos selecionados para o ciclo de desenvolvimento (saída da atividade 6);

#### Atividades

A elaboração das interfaces deve utilizar como referência as interfaces antigas, procurando manter a identidade anterior do sistema e realizando as modificações necessárias, evitando-se assim a perda de identidade ao sistema antigo.

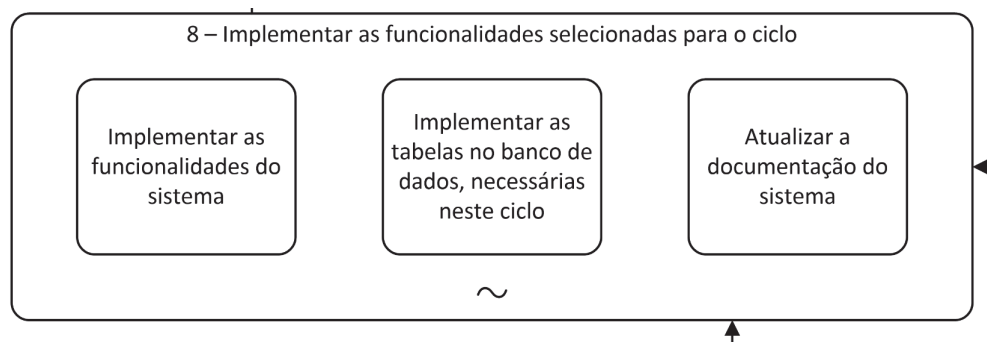
Como forma de validação da usabilidade do sistema, deve-se realizar a análise das interfaces do ciclo utilizando-se ao menos um dos métodos de avaliação de usabilidade descritos por [Junior e Silva \(2003\)](#), a fim de estabelecer um nível adequado de usabilidade do sistema. Adicionalmente, as interfaces devem ser validadas pelo cliente do sistema antes da programação das mesmas, a fim de evitar a inclusão de incrementos de correção de interfaces na lista de incrementos.

#### Saídas

- Interfaces do sistema;

## 8 - Implementar as funcionalidades do ciclo

Figura 22 – Subprocesso Expandido 8.



Nesta atividade é realizada a implementação da funcionalidade selecionada para o ciclo de desenvolvimento, envolvendo a codificação do sistema utilizando-se as ferramentas definidas anteriormente. Adicionalmente, nesta etapa do sistema também são implantadas no Banco de Dados as tabelas necessárias ao sistema neste ciclo.

As inconsistências identificadas no diagrama de classes e no modelo entidade-relacionamento deverão ser corrigidas, pois a documentação deve refletir a implementação realizada.

#### Entradas

- Diagrama de casos de uso do novo sistema (saída da atividade 3);
- Casos de uso expandidos do novo sistema (saída da atividade 3);
- Diagrama de classes do sistema (saída da atividade 4);
- Modelo entidade-relacionamento do sistema (saída da atividade 4);

- Incrementos selecionados para o ciclo de desenvolvimento (saída da atividade 6);
- Interfaces do sistema (saída da atividade 7);

#### Atividades

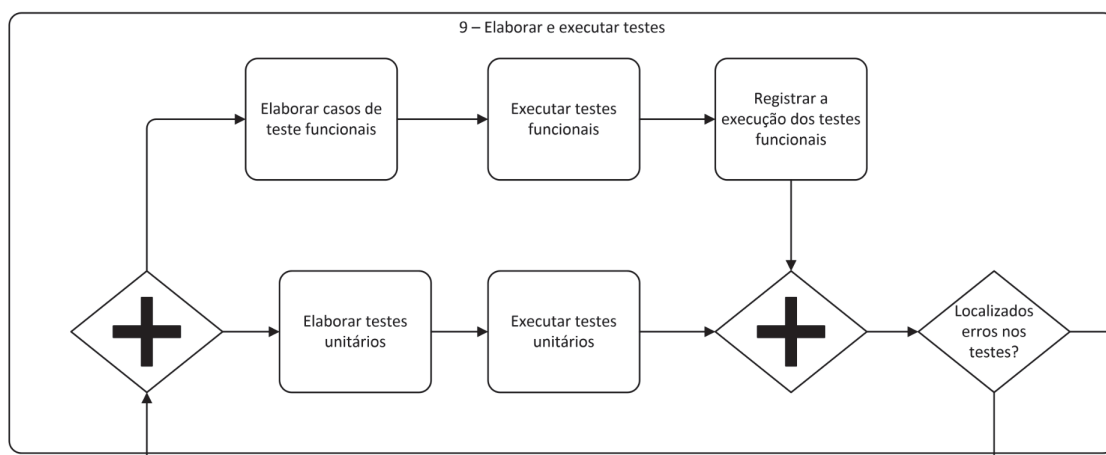
- Implementar as funcionalidades do sistema;
- Implementar as tabelas necessárias, no banco de dados do sistema;
- Atualizar a documentação do sistema;

#### Saídas

- Implementações das funcionalidades dos incrementos selecionados para o ciclo de desenvolvimento;

## 9 - Elaborar e executar testes

Figura 23 – Subprocesso Expandido 9.



A atividade de elaboração e execução de testes envolve testes funcionais e testes unitários. A realização de testes unitários poderá ser apoiada por ferramentas que automatizem o processo. O controle dos testes funcionais deverá ser realizado através de uma ferramenta que permita a criação dos testes funcionais e registre a execução dos casos de testes e seus resultados.

A realização de testes deve se focar em verificar a funcionalidade do sistema e também em verificar se a implementação da solução corresponde a mesma funcionalidade equivalente do software legado e as modificações solicitadas.



### Entradas

- Diagrama de casos de uso do novo sistema (saída da atividade 3);
- Casos de uso expandidos do novo sistema (saída da atividade 3);
- Diagrama de classes do sistema (saída da atividade 4);
- Modelo entidade-relacionamento do sistema (saída da atividade 4);
- Incrementos selecionados para o ciclo de desenvolvimento (saída da atividade 6);
- Implementações das funcionalidades dos incrementos selecionados para o ciclo de desenvolvimento (saídas da atividade 8);

### Atividades

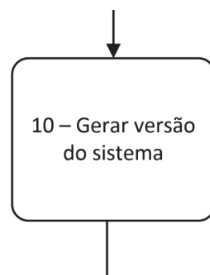
- Elaborar e executar testes unitários;
- Elaborar casos de teste funcionais;
- Executar e registrar a execução de casos de teste funcionais;

### Saídas

- Implementações das funcionalidades testadas;
- Casos de testes com o registro de execução;

## 10 - Gerar versão do sistema

Figura 24 – Atividade 10.



Após a realização dos testes é gerada uma versão do sistema, que incluirá todas as funcionalidades realizadas neste ciclo e também as funcionalidades realizadas nos ciclos anteriores, para posterior implantação e avaliação por parte do cliente.

A versão será identificada pelo número do ciclo que estiver em execução, como forma de manter o histórico das versões do sistema. Ex.: caso esteja ocorrendo o terceiro ciclo de desenvolvimento, ao final deste será gerada a versão 3 do sistema.

#### Entradas

- Implementação das funcionalidades testadas (saída da atividade 9);
- Funcionalidades implementadas em ciclos anteriores;

#### Atividades

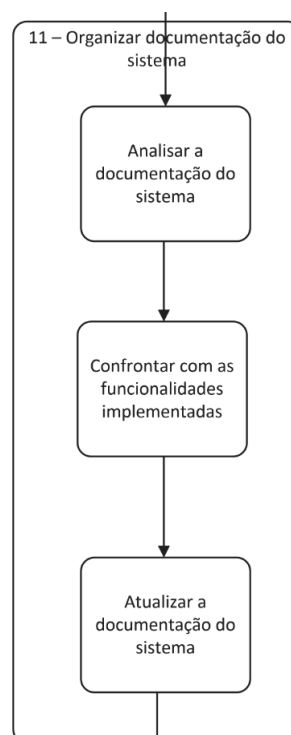
- Gerar versão do sistema;

#### Saídas

- Versão executável do sistema;
- Lista de funcionalidades implementadas no ciclo;

### 11 - Organizar documentação do sistema

Figura 25 – Subprocesso Expandido 11.



Após a implementação, teste e geração da versão do sistema, a documentação deve ser revisada, com o objetivo de verificar se as inconsistências identificadas nas etapas anteriores foram devidamente corrigidas e também se os diagramas foram complementados com informações omitidas inicialmente.

#### Entradas

- Documentação do sistema;
- Implementação das funcionalidades testadas (saída da atividade 9);

#### Atividades

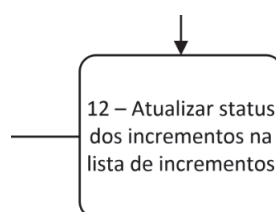
- Analisar a documentação do sistema e da Implementação das funcionalidades;
- Confrontar a documentação do sistema com as funcionalidades do sistema;
- Atualizar a documentação do sistema;

#### Saídas

- Documentações do sistema atualizadas;

## 12 - Atualizar status dos incrementos na lista de incrementos

Figura 26 – Atividade 12.



Após a conclusão do ciclo, a lista de incrementos deve ser atualizada, alterando os status dos incrementos da lista para “Realizado no ciclo”, acompanhado do número do ciclo em execução, referenciando em qual ciclo e versão do sistema o incremento foi realizado.

#### Entradas

- Lista de incrementos do sistema (saída da atividade 5);
- Lista de funcionalidades implementadas no ciclo (saída da atividade 10);

### Atividades

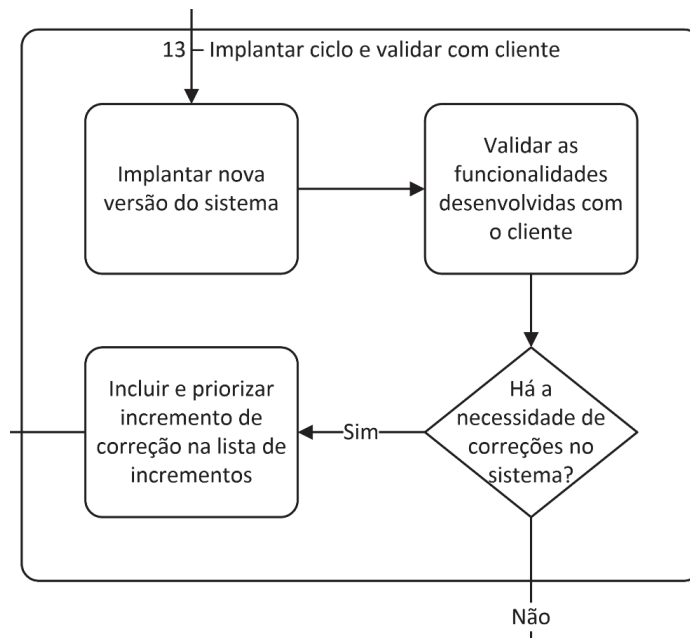
- Atualizar a lista de incrementos do sistema;

### Saídas

- Lista de incrementos do sistema atualizada;

## 13 - Implantar ciclo e validar

Figura 27 – Subprocesso Expandido 13.



A nova versão do sistema é então implantada no cliente, realizando a migração das funcionalidades que até então eram utilizadas no sistema legado para o novo sistema. O cliente então valida as novas funcionalidades implementadas no ciclo e avalia a necessidade de correções no sistema, para posteriormente serem incluídas na lista de incrementos do sistema.

### Entradas

- Versão executável do sistema (saída da atividade 10);

### Atividades

- Implantar nova versão do sistema no cliente;

- Validar as funcionalidades desenvolvidas e migradas (pelo cliente), gerando incrementos de correção, de acordo com a necessidade. Caso sejam gerados incrementos de correção, os mesmos devem ser priorizados, seguindo o método proposto por [Bezerra \(2007\)](#) para a priorização de casos de uso, e posteriormente inseridos na lista de incrementos. A correção será selecionada e realizada em um ciclo posterior, de acordo com a prioridade definida para o mesmo.;

#### Saídas

- Incrementos de correção (caso sejam necessárias correções no sistema);

