

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

MARCELO MAZINI BOMFIM JUNIOR

Repositório de Projetos de Software Mensurados em Pontos de Função

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2011

MARCELO MAZINI BOMFIM JUNIOR

Repositório de Projetos de Software Mensurados em Pontos de Função

Proposta de Trabalho de Conclusão de Curso, apresentada à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. José Augusto Fabri

CORNÉLIO PROCÓPIO

2011

LISTA DE ABREVIATURAS

AIE	Arquivo de Interface Externa
ALI	Arquivo Lógico Interno
APF	Análise de Pontos de Função
CE	Consulta Externa
CLF	Construir a Lista de Funcionalidades
CPF	Construir por Funcionalidade
CSS	<i>Cascading Style Sheets</i>
DMA	Desenvolver um Modelo Abrangente
DPF	Detalhar por Funcionalidade
EE	Entrada Externa
FDD	<i>Feature Drive Development</i>
IDE	<i>Integrated Development Environment</i>
IFPUG	<i>International Function Point Users Group</i>
ISBSG	<i>International Software Benchmarking Standards Group</i>
ISO	<i>International Organization for Standardization</i>
PF	Pontos de Função
PFNA	Número de pontos de função não ajustados
SE	Sáida Externa
UML	<i>Unified Modeling Language</i>
UTFPR	Universidade Tecnológica Federal do Paraná

LISTA DE FIGURAS

Figura 1 - Visão Geral do Processo de Análise de Pontos de Função.....	11
Figura 2 - Requisitos Funcionais	16
Figura 3 - Processo de Submissão de Projetos	20
Figura 4 - Usuários da Solução	24
Figura 5 - Arquitetura da Solução.....	25
Figura 6 - Ciclo de Vida do FDD.....	27
Figura 7 - Ciclo de vida do SCRUM	29
Figura 8 - Processo Adaptado.....	30
Figura 9 - Plano de Desenvolvimento	32

SUMÁRIO

1. INTRODUÇÃO	6
2. ANÁLISE DE PONTOS DE FUNÇÃO	9
3. ESPECIFICAÇÃO DA SOLUÇÃO	15
3.1 Requisitos Funcionais.....	15
3.1.1 Administração do Portal	17
3.1.2 Repositório de Projetos	19
3.1.3 Ferramenta para Análise de Pontos de Função.....	21
3.2 Limites da Solução	24
3.3 Descrição dos Usuários	24
3.4 Arquitetura da Solução	25
4. PROCESSO DE DESENVOLVIMENTO	26
4.1 FDD – <i>Feature Driven Development</i>	26
4.2 SCRUM.....	28
4.3 Adaptação do Processo.....	30
5. TECNOLOGIAS E FERRAMENTAS	34
6. ESTIMATIVA DE TAMANHO DA SOLUÇÃO	36
7. CRONOGRAMA	37
REFERÊNCIAS.....	38
APÊNDICE A – Diagrama de Classes – Repositório de Projetos	40
APÊNDICE B – Diagrama de Classes – Ferramenta APF.....	41
APÊNDICE C – Diagrama de Entidade e Relacionamento – Repositório de Projetos	42
APÊNDICE D – Diagrama de Entidade e Relacionamento – Ferramenta APF....	43
APÊNDICE E – Diagrama de Caso de Uso	44
APÊNDICE F – Plano de Desenvolvimento – <i>Sprint 1</i>	46
APÊNDICE G – Plano de Desenvolvimento – <i>Sprint 2</i>	47
APÊNDICE H – Plano de Desenvolvimento – <i>Sprint 3</i>.....	48
APÊNDICE I – Plano de Desenvolvimento – <i>Sprint 4</i>	49
APÊNDICE J – Cronograma Geral	50

1. INTRODUÇÃO

Com o crescimento acelerado da demanda de sistemas de informação, as empresas de *software* têm buscado cada vez mais por qualidade em seus produtos. Entregar um produto dentro do prazo e custos planejados sem perder a qualidade, ainda hoje é um grande desafio, pois a exigência do mercado em relação a tempo e custo menores é cada vez maior. Dessa forma, é fundamental que as empresas de *software* determinem o tamanho dos produtos visando à elaboração de cronogramas e orçamentos mais realistas, diminuindo o risco dos projetos serem subestimados ou superestimados. Com a determinação do tamanho dos produtos de *software* de forma padronizada, as empresas podem realizar comparações entre projetos, definindo também sua produtividade de desenvolvimento em relação a tamanho. Diante desse cenário, já foram propostas várias métricas de tamanho das funcionalidades de *software*, dentre as quais podemos destacar a Análise de Pontos de Função.

A técnica de análise de pontos de função é utilizada para dimensionamento de sistemas de informação, possibilitando a realização de estimativas de tempo e custo, auxiliando no planejamento do projeto de *software* (VAZQUEZ, SIMÕES e ALBERT, 2010). Diferentemente de outras técnicas para métricas de *software*, os pontos de função medem a funcionalidade na visão do usuário, ou seja, o quanto de funcionalidade o usuário final terá na entrega de determinado *software*, independentemente das tecnologias que serão utilizadas para desenvolvê-lo (DIAS, 2004).

A análise de pontos por função é uma das métricas mais utilizadas no mercado de *software*, principalmente pelos órgãos públicos brasileiros, os quais adotam o ponto de função como unidade padrão de medida de tamanho em editais de licitação para contratação de serviços de desenvolvimento de *software* (CAVALCANTE, 2010).

Dentro do contexto apresentado, esse trabalho tem como objetivo construir um repositório de projetos de *software* brasileiro mensurados em pontos de função e que tenham mensurado o total de esforço gasto para seu desenvolvimento. Esse repositório será de caráter público, onde serão armazenadas informações sobre a contagem de pontos de função e as principais características de cada projeto, por

exemplo: qual a linguagem de programação utilizada, processo utilizado para o desenvolvimento, total de esforço gasto para finalização do projeto, entre outras informações. Esses dados estarão disponíveis na web para consulta, análise e *benchmarking* por qualquer entidade interessada.

Atualmente existem algumas organizações que possuem repositórios de projetos com informações de tamanho e produtividade, principalmente repositórios internos, montando uma base histórica do que a organização já desenvolveu. Dessa forma, a cada novo projeto a organização pode consultar sua base histórica, e estabelecer melhores cronogramas de acordo com sua produtividade ao longo do tempo. Existem também organizações que possuem repositórios de projetos de várias organizações e disponibilizam essas informações para a comunidade. É o caso do ISBSG (*International Software Benchmarking Standards Group*), uma entidade sem fins lucrativos sediada na Austrália, que possui informações sobre tamanho funcional, esforço, custo, entre outros, de mais de 5000 projetos de diversos países. Para submeter um projeto, apenas é necessário preencher um questionário com aproximadamente 140 questões e enviá-lo ao ISBSG; porém, para acessar as informações armazenadas no ISBSG, é necessário pagar uma taxa €1500¹, inviabilizando sua aquisição para algumas empresas nacionais.

O repositório de projetos proposto neste trabalho não visa se tornar um repositório igual ao ISBSG. O objetivo desse repositório é ser o mais simples e objetivo possível, tendo um foco maior na realidade do mercado nacional, auxiliando pequenas e médias empresas a iniciar a utilização da técnica de pontos de função, sendo possível visualizar e comparar experiências de projetos similares de outras organizações. As informações do projeto, tais como: as características, a contagem de pontos de função, o total de esforço para desenvolvimento; estarão disponíveis para acesso livre e gratuito, exceto o nome da organização que submeteu o projeto.

Para que a aplicação da técnica de pontos de função seja simples e intuitiva, esse trabalho também contempla o desenvolvimento de uma ferramenta web para auxiliar qualquer organização no processo de contagem dos pontos de função, seguindo a referência oficial da técnica de APF, o manual de Práticas de Contagem do IFPUG (*International Function Point Users Group*). Atualmente, muitas

¹ Informação do ISBSG – International Software Benchmarking Standard Group – Acessado em 12 de maio de 2011. Disponível na Internet em: <http://www.isbsg.org/>

organizações acabam utilizando planilhas eletrônicas para executar esse processo, dificultando a recuperação de informações sobre a contagem dos pontos de função de forma rápida e eficaz. Dessa forma, essa ferramenta irá permitir que profissionais da área de métricas de software realizem a contagem dos pontos de função de seus projetos, podendo manter suas próprias bases históricas, publicando-as também no repositório de projetos para apreciação da comunidade. A princípio, essa ferramenta irá focar apenas no processo de contagem dos pontos de função, possibilitando futuramente o desenvolvimento de outros módulos relacionados à gestão dos projetos. Essa ferramenta será também uma ponte para a submissão de projetos para o repositório, além de simplificar e organizar o processo de contagem.

A proposta é inicialmente coletar projetos de *software* oriundos de trabalhos de diplomação do curso de Análise e Desenvolvimento de Sistemas da UTFPR (Universidade Tecnológica Federal do Paraná), campus Cornélio Procopio. Posteriormente, com o aumento do número de projetos no repositório, será realizada divulgações para as empresas de *software*, para que as mesmas submetam seus projetos, utilizem a ferramenta de aplicação da técnica e usufruam das informações do repositório.

Para melhor apresentar a proposta em questão, este trabalho está estruturado em sete capítulos. No capítulo um, descreve-se sucintamente o tema sobre o que se propõe a desenvolver, demonstrando o objetivo geral do trabalho. No capítulo dois, apresenta-se um referencial bibliográfico sobre a técnica de análise de pontos de função, para melhor entendimento da área de conhecimento em questão. No capítulo três, apresenta-se a especificação da solução, detalhando todo o escopo que será desenvolvido nesse trabalho. No capítulo quatro, apresenta-se o processo de desenvolvimento que será utilizado, detalhando as atividades que já foram realizadas, as atividades que ainda deverão ser cumpridas e os artefatos que serão gerados. No capítulo cinco, apresentam-se as tecnologias e ferramentas necessárias para a execução de todo o trabalho. No capítulo seis é apresentada a estimativa de tamanho da solução utilizando a técnica de pontos de função. No capítulo sete será apresentado o cronograma de todo o trabalho.

2. ANÁLISE DE PONTOS DE FUNÇÃO

A exploração do uso da tecnologia da informação tem ganhado força nos últimos anos, trazendo elevado valor estratégico em todas as atividades das organizações. Dessa forma, a demanda por sistemas de informação é cada vez maior o que gera uma grande agitação e competição no mercado de *software* (CAVALCANTE, 2010). Dessa forma as empresas de *software* têm-se preocupado cada vez mais com a qualidade de seus produtos, principalmente com o cumprimento de cronogramas, buscando novas tecnologias e metodologias de engenharia de *software*. No entanto, existem vários problemas associados à gestão dos projetos, tais como: tempo estimado de duração, custo, alocação de recursos, esforço gasto, entre outros. Muitos desses aspectos podem ser estimados quantitativamente. Dessa forma, a medição de tamanho do *software* se torna um processo essencial para elaboração de cronograma mais realistas, servindo de base para definição de esforço e custo, auxiliando em toda gestão do projeto (HAZAN e STAA, 2005).

Para se chegar a uma medida de tamanho funcional de *software* existem várias métricas². A NBR ISO/IEC 14143 objetiva garantir que todos os métodos de Medição Funcional de Tamanho sejam baseados em conceitos similares e que possam ser testados para assegurar que eles se comportam de maneira similar e da forma esperada por um método de medição (CALAZANS e OLIVEIRA, 2005).

A análise de pontos de função é uma medida funcional utilizada para dimensionamento de sistemas de informação, servindo como base para a realização de estimativas de tempo e custo, auxiliando no planejamento do projeto de *software* (VAZQUEZ, SIMÕES e ALBERT, 2010). Diferentemente de outras técnicas para métricas de *software*, os pontos de função medem a funcionalidade na visão do usuário, ou seja, o quanto de funcionalidade o usuário final terá na entrega de determinado *software*, independentemente das tecnologias que serão utilizadas para desenvolvê-lo (DIAS, 2004).

A APF surgiu na IBM, no início da década de 70 conduzida por Allan Albrecht. A princípio, a técnica seria uma alternativa às métricas baseadas em linhas

² Exemplo: Linhas de Código, Análise de Pontos de Função e Pontos por Caso de Uso.

de código, consideradas as mais relevantes da época. A grande motivação para a criação da técnica de análise de pontos de função se dava pela busca de uma medida que fosse independente da linguagem de programação utilizada, podendo padronizar o tamanho dos projetos. Com a divulgação da técnica à comunidade no final da década de 70, houve um crescimento acelerado de profissionais utilizando a técnica, até que em 1986 foi fundado o *International Function Point Users Group* - IFPUG. O IFPUG é uma organização sem fins lucrativos, que tem como missão promover e incentivar a gestão eficaz de desenvolvimento de aplicações de *software* e atividades de manutenção através do uso da técnica de análise de pontos de função e outras técnicas de métricas de software (VAZQUEZ, SIMÕES e ALBERT, 2010).

Desde sua fundação, o IFPUG passou a ser o responsável pela definição das regras de contagem, treinamento e certificação dos profissionais interessados na aplicação dessa métrica e divulgação de diversas bases de dados históricas de produtividade da indústria de desenvolvimento de *software*, disponibilizadas por vários órgãos, entre eles o *International Software Benchmarking Standards Group* (ANDRADE, 2004). O IFPUG mantém o manual de práticas de contagem, que atualmente está na *release* 4.3.1³, onde se encontra disponível para *download* gratuitamente para membros do IFPUG, e para não membros através de sua aquisição. No Brasil, o uso da técnica começou significativamente no início da década de 90. Com o grande interesse do mercado e a consolidação da técnica, foi fundada uma representação oficial do IFPUG no Brasil, a *Brazilian Function Point Users Group* – BFUPG.

A análise de pontos de função é uma das métricas mais utilizadas no mercado de *software*, principalmente pelos órgãos públicos brasileiros, nos quais adotam o ponto de função como unidade padrão de medida de tamanho em editais de licitação para contratação de serviços de desenvolvimento de *software* (CAVALCANTE, 2010). Em 2002 a técnica se tornou um padrão internacional através da norma ISO/IEC 20.926 (AGUIAR, 2003).

A utilização da técnica é muito útil para estimar custo, cronograma e esforço do projeto, facilitar negociações contratuais, entender e controlar o tamanho

³ Informação do IFPUG - International Function Point Users Group – Acessado em 12 de maio de 2011. Disponível na Internet em: <http://www.ifpug.org/>

de escopo e estimar casos de testes (ANDRADE e OLIVEIRA, 2004). Muitas organizações internacionais investiram no levantamento e armazenamento de grande quantidade de dados sobre projetos, envolvendo pontos de função. Nenhuma outra medida funcional alcançou tal nível de disseminação e/ou investimento (CAVALCANTE, 2010).

Para que seja possível aplicar a técnica, é extremamente importante ter os requisitos previamente definidos. É claro que os requisitos não precisam estar totalmente detalhados, pois isso ocorre progressivamente ao longo do projeto. Porém, quanto mais detalhado o requisito estiver, mais precisa será a medição. A aplicação da técnica é bem simples, onde o processo de contagem dos pontos de função pode ser dividido em sete etapas (vide Figura 1):

- I. Determinar tipo de contagem;
- II. Identificar a fronteira da aplicação;
- III. Contar as funções tipo dados;
- IV. Contar as funções tipo transação;
- V. Calcular pontos de função não ajustados (com base nos resultados obtidos em (III) e (IV));
- VI. Calcular o valor do fator de ajuste;
- VII. Calcular os pontos de função ajustados (com base nos resultados obtidos em (V) e (VI)).

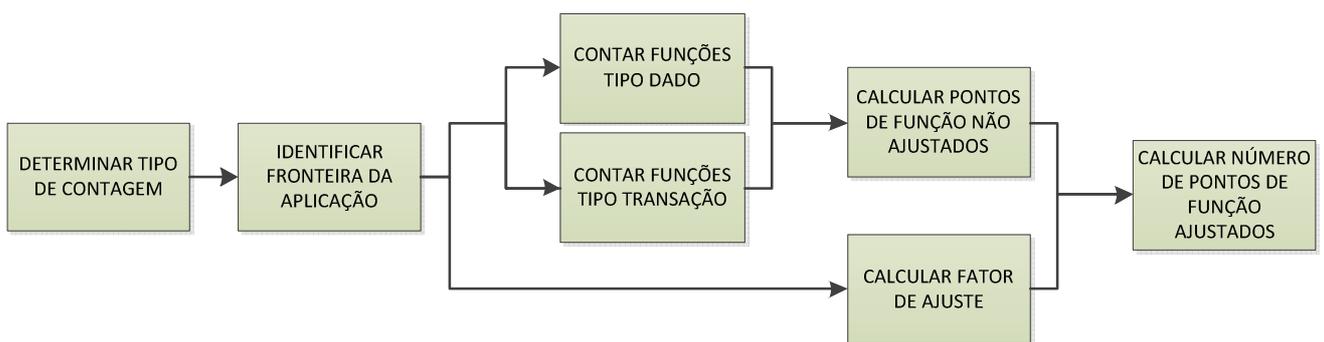


Figura 1 - Visão Geral do Processo de Análise de Pontos de Função
Fonte: Adaptado de (GARMUS e HERRON, 2001)

Segundo Barcellos (2003), na etapa I é necessário escolher o tipo de contagem, sendo determinante para o cálculo dos pontos de função posteriormente. O tipo de contagem pode ser:

- **Projeto de Desenvolvimento:** mede a funcionalidade fornecida aos usuários finais do *software* para a primeira instalação da aplicação, realizada principalmente para estimativa de tamanho de um novo projeto, incluindo as funcionalidades iniciais da aplicação e as funcionalidades requeridas para conversão de dados.

- **Projeto de Manutenção:** mede as modificações realizadas para aplicações existentes, podendo ser uma adição, modificação ou exclusão de funções na aplicação. Após a manutenção, a contagem da aplicação deve ser refeita para refletir as alterações realizadas.

- **Aplicação:** mede uma aplicação instalada, avaliando as funcionalidades correntes providas aos usuários finais de uma aplicação.

Na etapa II é necessário definir a fronteira da aplicação, indicando a separação entre o projeto que está sendo medido e as aplicações externas ao domínio do usuário. É através dela que torna-se possível definir quais funcionalidades serão incluídas no processo de contagem dos pontos de função (DIAS, 2004).

Na etapa III é realizada a contagem das funções tipo dados, avaliando o que é fornecido pela aplicação, não do como é fornecido. As funções tipo dados representam as funcionalidades fornecidas pelo sistema ao usuário, para atender às necessidades referentes aos dados que o sistema irá manipular (BARCELLOS, 2003). Segundo Dias (2004), essas funções podem ser:

- **Arquivos Lógicos Internos (ALI):** São grupos lógicos de dados ou informações de controle reconhecidos pelos usuários, onde são manipulados por meio de um processo elementar ⁴ da aplicação (inclusão, alteração, exclusão) dentro da fronteira da aplicação. Por exemplo: as tabelas e/ou classes do sistema.

- **Arquivos de Interface Externa (AIE):** são grupos lógicos de dados, mantidos por outra aplicação, apenas referenciados pelo projeto sendo desenvolvido. Por exemplo: as tabelas acessadas de um outro sistema.

Tanto ALI quanto AIE, é necessário observar a visão do usuário, não considerando arquivos físicos, arquivos de índice, arquivos de trabalho, tabelas de relacionamento sem atributos próprios e entidades fracas (IFPUG, 2004).

⁴ Processo elementar é menor atividade percebida pelo usuário que dever ser executada pelo sistema

Na etapa IV são contados os pontos de função avaliando as funcionalidades de processamento oferecidas aos usuários, nas quais são chamadas de funções tipo transação, podendo ser:

- **Entradas Externas (EE)** - é uma transação que processa dados ou informação de controle que entram pela fronteira da aplicação. Seu objetivo é manter um ou mais ALIs e/ou alterar o comportamento do sistema (VAZQUEZ, SIMÕES e ALBERT, 2010).

- **Consultas Externas (CE)**: é uma transação que envia dados ou informação de controle para fora da fronteira da aplicação. Seu objetivo é apresentar informação para o usuário através da recuperação de dados ou informação de controle de ALIs ou AIEs. Seu processamento lógico não contém fórmulas matemáticas ou cálculos, e não cria dados derivados. Nenhum ALI é mantido durante a transação, e o comportamento do sistema não é alterado (HAZAN e STAA, 2005).

- **Saídas Externas (SE)**: é uma transação que envia dados ou informação de controle para fora da fronteira da aplicação. O objetivo principal é apresentar informação para um usuário através de um processamento lógico adicional a recuperação de dados ou informação de controle. Seu processamento lógico deve conter no mínimo uma fórmula matemática ou cálculo, ou criar dados derivados. Uma SE pode manter um mais ALIs e/ou alterar o comportamento do sistema (HAZAN e STAA, 2005).

Na etapa V, é realizada a contagem dos pontos de função não ajustados com base na contagem realizada na etapa III e IV, onde foram contadas as funções tipo dado e transação. Para realizar essa contagem, o manual do IFPUG fornece tabelas e diretrizes para determinar a complexidade de cada elemento funcional. A complexidade dos ALI e AIE é baseada número de registros lógicos e no número de itens de dados referenciados e a complexidade das EE, SE e CE é baseada no número de arquivos referenciados e no número de itens de dados referenciados (TAVARES, CARVALHO e CASTRO, 2004).

Na etapa VI é calculado o fator de ajuste, onde são levadas em consideração quatorze características da aplicação: Comunicação de Dados, Processamento Distribuído, Atualização de Dados Online, Entrada de Dados Online, Volume de Transações, Eficiência do Usuário Final, Complexidade do

Processamento, Facilidade de Implantação, Multiplicidade de Locais, Facilidade de Mudanças, Facilidade Operacional, Desempenho, Utilização do equipamento, Reutilização de Código (ANDRADE, 2004). O manual do IFPUG fornece diretrizes para que cada característica seja atribuído um nível de influência de 0 (zero) a 5 (cinco), onde 0 (zero) indica nenhuma influência, 1 (um) influência mínima, 2 (dois) influência moderada, 3 (três) influência média, 4 (quatro) influência significativa e 5 (cinco) grande influência (BARCELLOS, 2003). Atribuídos os pesos, é contado o Nível de Influência Geral, sendo obtido pelo somatório do nível de influência de cada característica. O Fator de Ajuste é obtido pela expressão:

$$\text{Fator de ajuste} = 0,65 + (\text{Nível de Influência Geral} * 0,01)$$

Na etapa VII é calculado os pontos de função ajustados, sendo somente necessário multiplicar o valor dos pontos de função não ajustados, obtidos na etapa V, pelo valor do fator de ajuste, obtido na etapa VI.

$$\text{PFs Ajustados} = \text{PFs Não-Ajustados} * \text{Fator de Ajuste}$$

3. ESPECIFICAÇÃO DA SOLUÇÃO

Neste capítulo será demonstrada a especificação inicial da solução a ser desenvolvida. Serão descritos os requisitos funcionais, detalhando todo o escopo, a arquitetura, os limites e a descrição dos usuários da solução.

3.1 Requisitos Funcionais

A princípio a solução será dividida em três módulos: Administração do Portal, Repositório de Projetos, Ferramenta para aplicação da APF. O módulo de Administração do Portal será utilizado pela equipe do portal para configuração e publicação dos projetos submetidos ao repositório. Já os módulos “Repositório de Projetos” e “Ferramenta para aplicação da APF” serão utilizados por qualquer usuário interessado. Porém, para que o usuário submeta os projetos para o repositório e utilize a ferramenta de APF, o mesmo deverá se cadastrar no portal, obtendo um *login* e senha para acesso. Apenas a busca de projetos no repositório estará disponível para qualquer usuário comum. A Figura 2 a seguir demonstra todos os requisitos funcionais de forma macro agrupados por módulo.

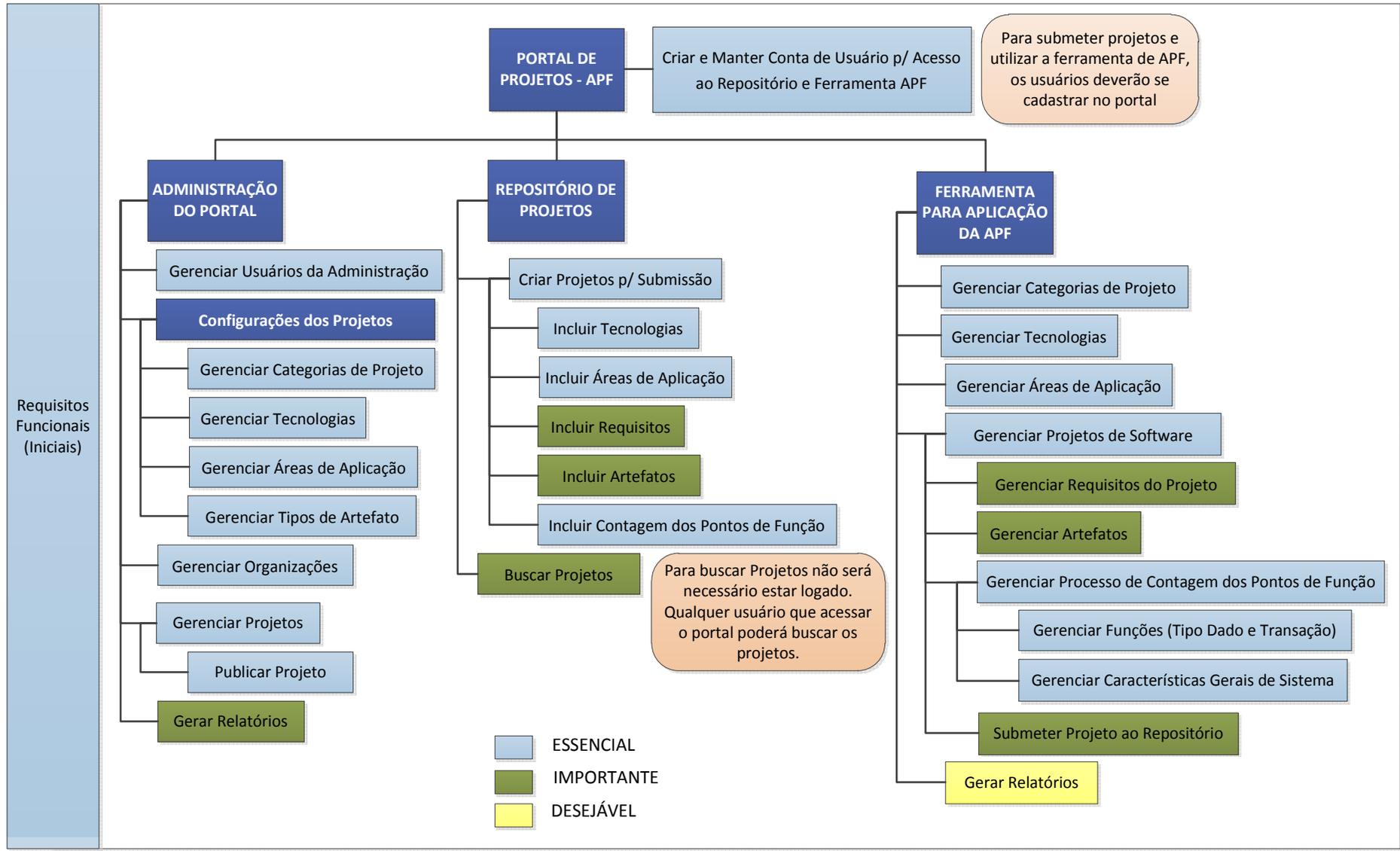


Figura 2 - Requisitos Funcionais

3.1.1 Administração do Portal

Este módulo será utilizado pela equipe do portal para administrar o repositório de projetos, configurando os itens necessários para compor as informações dos projetos e também avaliando os itens submetidos ao repositório para posteriormente publicá-los. Nessa área, será possível gerenciar os dados das organizações que submetem projetos e também que utilizam a ferramenta de aplicação da técnica de pontos de função. A seguir serão listados os requisitos para esse módulo.

Gerenciar Usuários da Administração do Portal

Neste módulo será possível administrar todos os projetos submetidos e configurar os itens que compõem os mesmos, por exemplo: categorias, tecnologias, áreas de aplicação, etc. Porém, somente pessoas autorizadas poderão realizar essas tarefas. Dessa forma, será necessário possibilitar a inclusão, alteração e exclusão de usuários para compor a equipe do portal. Inicialmente a solução terá apenas um usuário Administrador que poderá incluir novos usuários. Esses usuários serão responsáveis por toda configuração dos itens do repositório para que as organizações submetam seus projetos com os itens já pré-definidos. Também serão responsáveis por avaliar e publicar os projetos submetidos pelas organizações.

Gerenciar Categorias de Projeto

Para que seja possível categorizar os projetos, os administradores poderão incluir, alterar e excluir as categorias. Ao longo do tempo, com o crescimento de submissões de novos projetos poderá surgir à necessidade de se incluírem novas categorias, por isso a necessidade de possuir uma gerência das categorias, deixando dinâmico esse item em toda a solução. Esse item servirá de filtro para realização de buscas de projetos em todo o repositório. O cadastro de categorias deverá ser bem simples, contendo apenas dois campos fundamentais: nome e descrição.

Gerenciar Tecnologias

Para que ao submeter os projetos, as organizações possam informar quais tecnologias foram utilizadas, será necessário configurar as principais tecnologias já existentes. Dessa forma, a solução deverá permitir que os administradores incluam, alterem e excluam essas tecnologias, para que sejam utilizadas nos projetos do repositório. Essas tecnologias poderão ser uma linguagem de programação, banco de dados, processo de desenvolvimento, entre outros. As tecnologias servirão de filtro para realização de busca de projetos no repositório. O cadastro de tecnologia deverá ser bem simples, contendo apenas três campos essenciais: nome, descrição e tipo (Linguagem de Programação, Banco de Dados, IDE, entre outros).

Gerenciar Áreas de Aplicação

Além de categorizar os projetos, também será necessário definir quais as áreas de aplicação dos mesmos que foram submetidos, por exemplo: Finanças e *Business Service*, *Banking*, Indústria Química, entre outros. Dessa forma a solução deverá permitir que os administradores possam incluir, alterar e excluir essas áreas de aplicação, para configurar os projetos no repositório. As áreas de aplicação também servirão de filtro para realização de busca de projetos no repositório. O cadastro de desse item deverá ser bem simples, contendo apenas dois campos importantes: nome e descrição.

Gerenciar Tipos de Artefato

Cada projeto poderá ter vários artefatos gerados durante o processo de desenvolvimento. Ao submeter os projetos para o repositório, as organizações poderão enviar esses artefatos produzidos durante seu desenvolvimento para complementar as informações sobre o mesmo. Dessa forma, os artefatos deverão ser categorizados para que seja possível identificá-los rapidamente. Assim, a solução deverá permitir que os administradores configurem os tipos de artefato, realizando as seguintes ações: inclusão, alteração e exclusão. Essas configurações irão refletir também na ferramenta de APF, onde será possível incluir artefatos aos projetos.

Gerenciar Organizações

A solução deverá permitir que seja possível gerenciar as organizações que submeteram projetos ou utilizam a ferramenta APF. Uma organização poderá ser uma empresa, um órgão do governo ou simplesmente uma pessoa física. As principais informações sobre as organizações deverão ser mantidas, tais como: nome da organização, endereço, cidade, estado, CEP, e-mail, telefone, nome da pessoa de contato, cargo da pessoa de contato, e-mail da pessoa de contato.

Gerenciar Projetos

Todo projeto submetido para o repositório será avaliado pela equipe do portal. Dessa forma, o sistema deverá possuir uma tela onde os administradores possam visualizar os projetos submetidos, avaliá-los e publicá-los no repositório. Ao publicar o projeto no repositório, o mesmo já estará disponível para consulta.

Gerar Relatórios

Para cada projeto, deverá ser possível gerar um relatório geral com todos os dados submetidos, pois posteriormente será enviado para o responsável pela organização que submeteu o projeto.

3.1.2 Repositório de Projetos

O repositório de projetos tem como foco principal armazenar informações de projetos de *software*, onde seu tamanho tenha sido mensurado utilizando a técnica de pontos de função. Isso oferecerá às empresas que estão iniciando a utilizar a técnica de APF em seus projetos, um banco de exemplos para que as mesmas tenham algum parâmetro logo de início. Para que isso seja possível, outras organizações precisarão contribuir submetendo seus projetos para o repositório. A seguir serão listados os requisitos mínimos para que o repositório ofereça o que está sendo proposto.

Criar Projetos para Submissão

Para submeter os projetos para o repositório, cada organização (usuário autenticado) deverá responder um questionário em três etapas: características do

projeto (categoria, áreas de aplicação, tecnologias), a contagem dos pontos de função e esforço, e os requisitos e artefatos do projeto; sendo a terceira etapa não obrigatória. Essas informações deverão ser armazenadas para que posteriormente a equipe do portal avalie e publique o projeto. Na Figura 3 é mostrado o fluxograma do processo de submissão dos projetos.

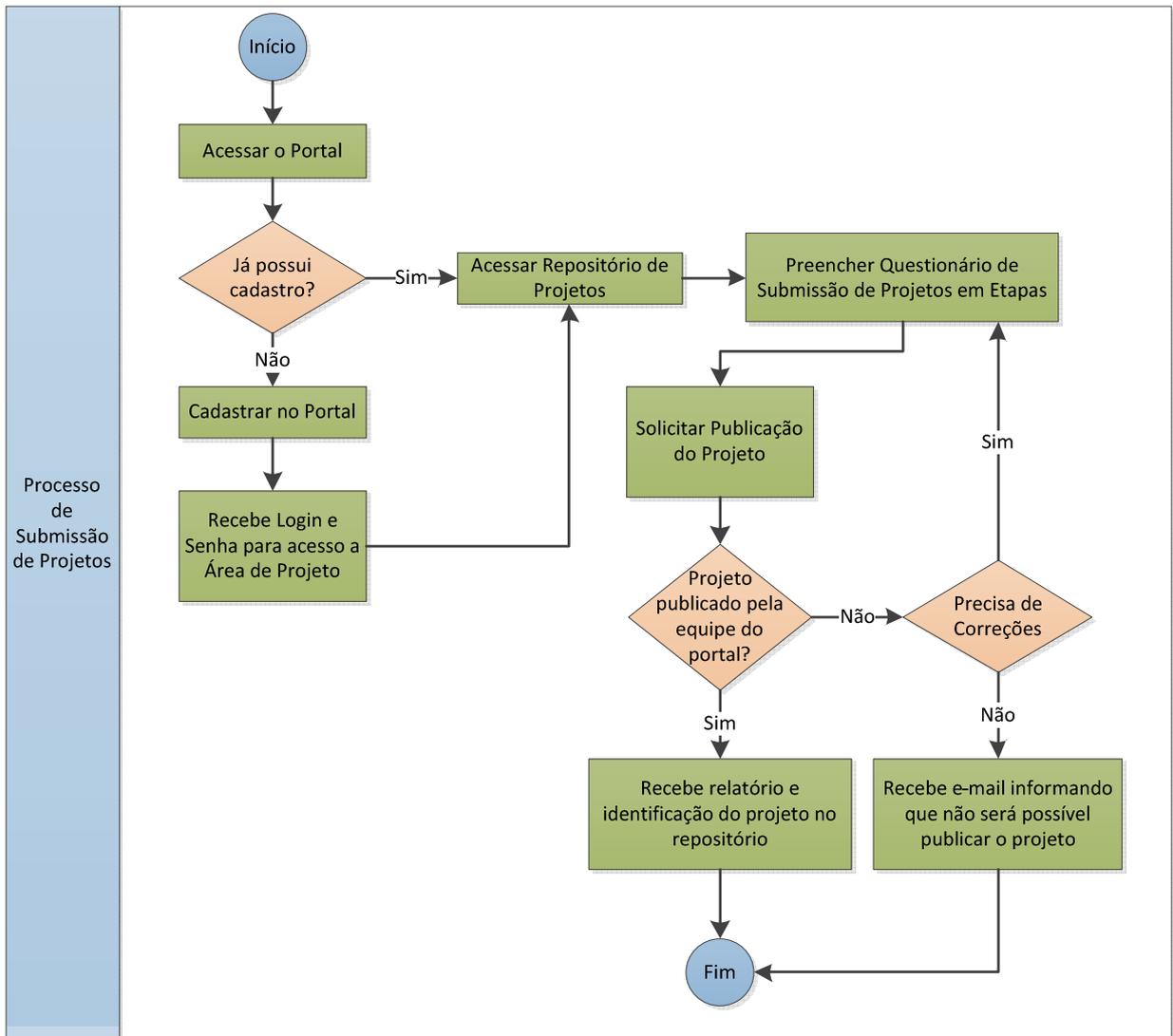


Figura 3 - Processo de Submissão de Projetos

Buscar Projetos no Repositório

Deverá ser desenvolvido um mecanismo de busca de projetos no repositório, para que entidades interessadas em realizar análise e *benchmarking* entre os projetos possam interagir com a solução. Essa busca deverá possuir vários filtros para que a mesma seja bem específica.

3.1.3 Ferramenta para Análise de Pontos de Função

Para simplificar e organizar o processo de contagem da técnica de APF será desenvolvido uma ferramenta para aplicação da técnica, seguindo o manual 4.3.1 de Práticas de Contagem do IFPUG. Essa ferramenta deverá ser bem intuitiva para fácil assimilação da técnica, agilizando o processo de contagem. Para acessar a ferramenta, cada organização deverá se cadastrar no portal, criando um *login* e senha para acesso. A seguir serão listados os requisitos mínimos para que a ferramenta ofereça o que está sendo proposto.

Gerenciar Categorias de Projeto

Cada organização poderá categorizar seus projetos de forma independente do repositório. Dessa forma, a ferramenta deverá permitir que seja possível incluir, alterar e excluir as categorias de projetos de cada organização. Quando a organização se cadastra no portal, automaticamente as categorias de projetos do repositório são importadas para a ferramenta de APF da organização.

Gerenciar Áreas de Aplicação

Além de categorizar os projetos, as organizações também poderão definir quais as áreas de aplicação dos mesmos, por exemplo: Finanças e *Business Service*, *Banking*, Construção Civil, entre outros. Dessa forma a solução deverá permitir que as organizações possam incluir, alterar e excluir essas áreas de aplicação, para configurar seus projetos. Quando a organização se cadastra no portal, automaticamente as áreas de aplicação do repositório são importadas para a ferramenta de APF da organização.

Gerenciar Tecnologias

Cada organização poderá definir as tecnologias que a mesma utiliza nos projetos de forma independente do repositório. Dessa forma, a ferramenta deverá permitir que as organizações incluam, alterem e excluam essas tecnologias, para que sejam utilizadas em seus projetos. Quando a organização se cadastra no portal, automaticamente as tecnologias do repositório são importadas para a ferramenta de APF da organização.

Gerenciar Projetos de Software

Cada organização poderá manter informações de seus projetos para realizar a contagem dos pontos de função, armazenando as seguintes informações: nome, breve descrição, categoria, área de aplicação e tecnologias utilizadas. Posteriormente, essas informações poderão ser utilizadas para submeter os projetos para o repositório.

Gerenciar Requisitos do Projeto

Para cada projeto poderá ser incluído os requisitos funcionais e não funcionais para melhor organizar e auxiliar no processo de contagem dos pontos de função. Dessa forma a ferramenta deverá permitir cadastrar os requisitos de forma simplificada, persistindo as seguintes informações: código, nome, tipo (funcional ou não funcional) e descrição.

Gerenciar Artefatos

Qualquer artefato gerado nos projetos poderá ser incluído na ferramenta para ter um controle maior sobre as informações do mesmo. Deverá ser possível anexar um arquivo em cada artefato, para melhor controle sobre o mesmo.

Gerenciar o processo de contagem de pontos por função

Cada projeto poderá ter várias contagens dos pontos de função. Assim, a ferramenta deverá permitir que as organizações gerenciem todas as contagens do projeto. As contagens poderão ser do tipo: desenvolvimento, melhoria ou aplicação (*baseline*). Para cada contagem deverá ser armazenado as seguintes informações: nome, descrição, propósito, escopo, total de pontos não ajustados, fator de ajuste, total de pontos ajustados. Os três últimos campos serão calculados ao serem incluídas as funções tipo dado e transação e o preenchimento das influências das características gerais de sistemas, servindo de base para se obter o valor do fator de ajuste.

Cada contagem possui funções tipo dado e transação. Dessa forma a ferramenta deverá permitir gerenciar todas essas funções, para que seja calculado o total de pontos de função. Essas funções poderão ser do tipo: Arquivo Interno Lógico, Arquivo de Interface Externa, Entrada Externa, Saída Externa e Consulta

Externa. Para cada tipo existe uma maneira de calcular a contribuição para o total de pontos de função de toda a contagem, definindo sua complexidade de acordo com a quantidade de itens relacionados. O sistema deverá permitir a inclusão, edição e exclusão desses itens de forma simples e objetiva para que esse processo seja o mais rápido possível, otimizando o tempo do usuário.

Para cada contagem também é fundamental o preenchimento das características gerais de sistemas, servindo de base para calcular o fator de ajuste. As características gerais de sistemas são como nossos requisitos não funcionais. Existem quatorze CGS (Características Gerais de Sistemas), na qual deverá ser armazenado o grau de influência de cada uma delas em cada contagem. Caso o usuário não queira utilizar o fator de ajuste na contagem, o mesmo poderá escolher a opção de adotar o valor de fator de ajuste igual a um.

Submeter projeto ao Repositório

Ao final de cada projeto, o usuário poderá submeter o mesmo para o repositório. A ferramenta deverá permitir que o projeto seja convertido para a estrutura do repositório, onde o usuário deverá apenas registrar mais algumas informações complementares.

Gerar Relatórios

A ferramenta deverá permitir que sejam gerados alguns relatórios sobre as informações produzidas no processo de contagem dos pontos de função. Inicialmente deverão ser criados os seguintes relatórios:

- Relatório Resumido de cada contagem de pontos de função de cada projeto.
- Relatório Geral do Projeto com resumo de todas as contagens realizadas no projeto.
- Relatório detalhado de cada contagem de pontos de função.

3.2 Limites da Solução

Por se tratar de uma primeira versão da solução, a ferramenta para aplicação da técnica de pontos de função não conterà módulos para gestão dos projetos de forma mais detalhada e ferramentas para consulta de projetos na base histórica de cada organização, apenas se restringindo ao processo de medição de tamanho funcional em pontos de função. O repositório de projetos inicialmente não oferecerá ferramentas para realização de análise mais aprofundada, apenas oferecendo uma busca de projetos por similaridade de forma simples. Esses itens citados anteriormente poderão ser desenvolvidos em trabalhos futuros.

3.3 Descrição dos Usuários

A solução possui três tipos de usuário, representados na figura abaixo.

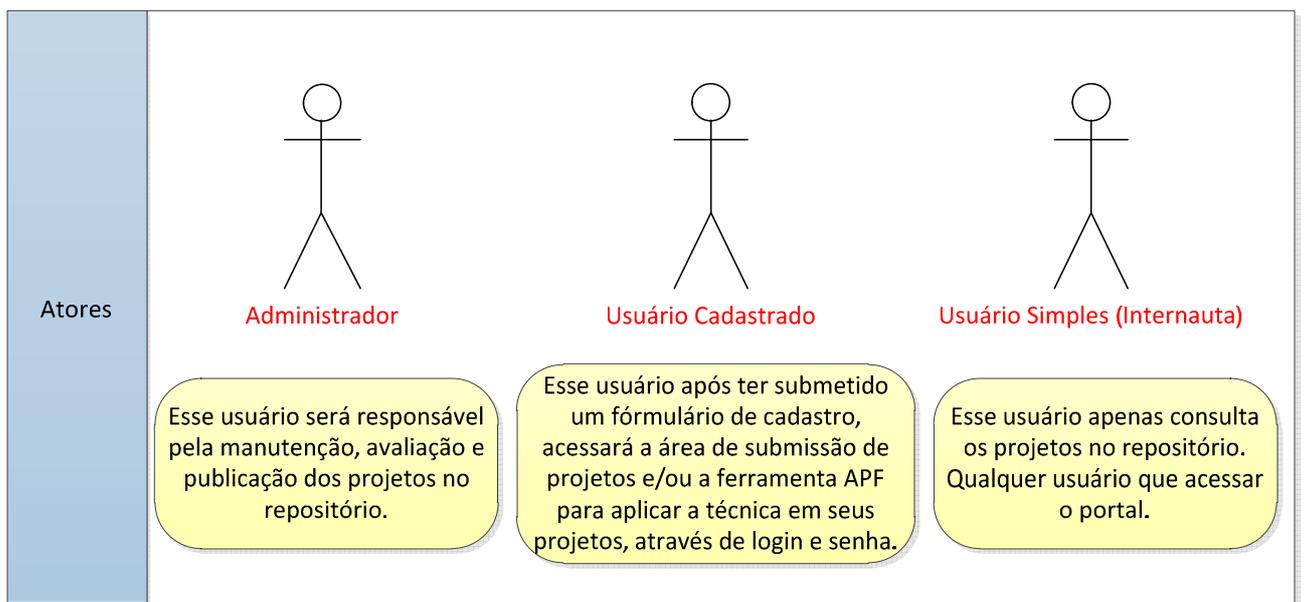


Figura 4 - Usuários da Solução

3.4 Arquitetura da Solução

Uma das arquiteturas de sistemas mais utilizadas atualmente é a aplicação em três camadas. Esta por sua vez, é muito utilizada para aplicações web, separando a lógica de apresentação, a lógica de negócio e a conexão com o banco de dados (lógica de acesso a dados).

Como a solução será desenvolvida especificamente para ambiente web, será adotada essa arquitetura em camadas, sendo elas: apresentação (browser do cliente), negócio e base de dados (Persistência).

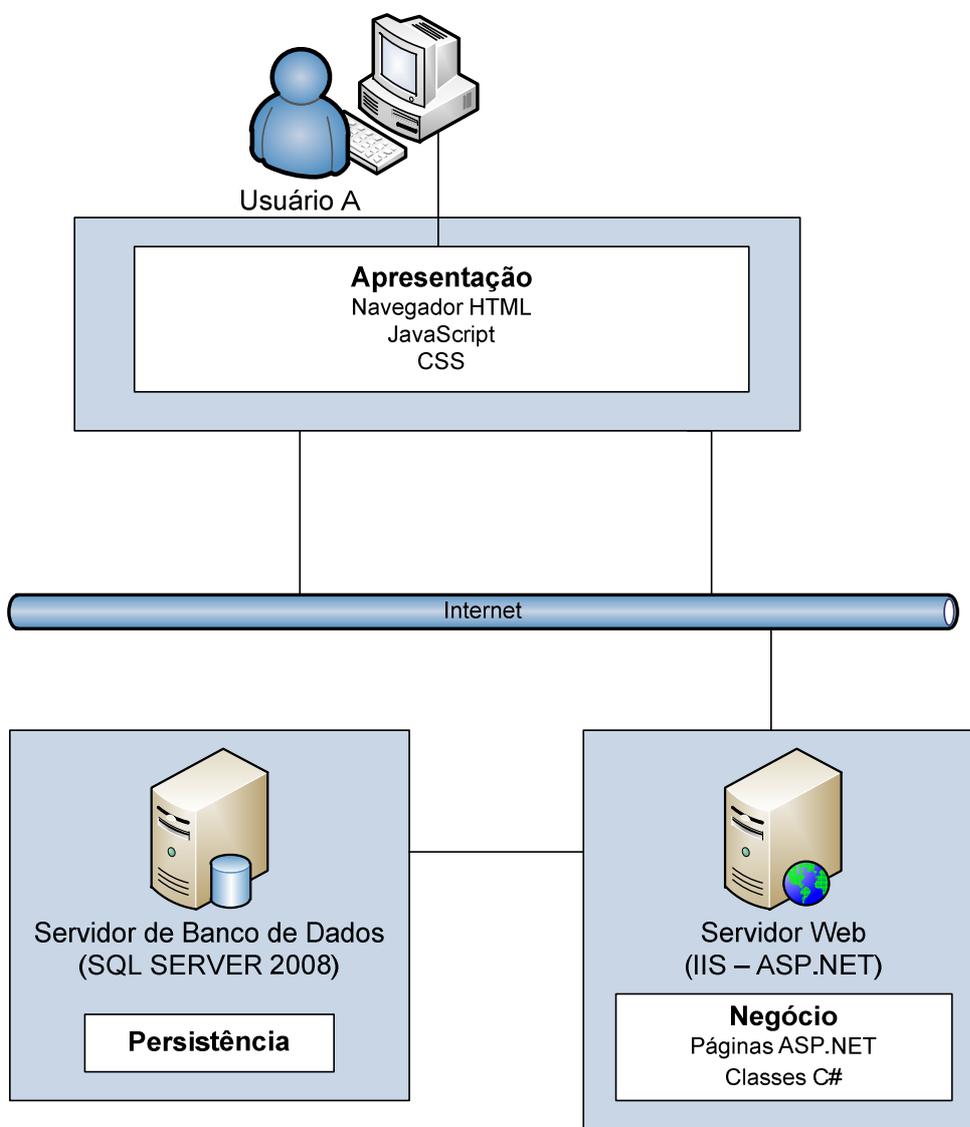


Figura 5 - Arquitetura da Solução

4. PROCESSO DE DESENVOLVIMENTO

A solução proposta será desenvolvida seguindo um processo híbrido de desenvolvimento, utilizando práticas das metodologias FDD (*Feature Driven Development*) e SCRUM, dando foco maior na FDD, tendo em vista que a mesma tem uma capacidade gerencial mais satisfatória em seus processos. A grande motivação para a montagem de um processo ágil e ao mesmo tempo robusto se deve pelo fato de que o projeto será gerenciado e desenvolvido por apenas uma pessoa, inviabilizando o uso de um processo completo seguido à risca. Dessa forma será feitas adaptações para simplificar o gerenciamento, documentação e desenvolvimento da solução. O ciclo de vida segue um modelo iterativo e incremental, possibilitando a entrega do produto em pequenas partes, já oferecendo ao “cliente” as principais funcionalidades já de início. A seguir será demonstrada uma breve descrição sobre FDD e SCRUM, para um melhor entendimento do que será adaptado entre as duas metodologias.

4.1 FDD – *Feature Driven Development*

A FDD é uma metodologia ágil de desenvolvimento de *software*, criada a partir das experiências de Peter Coad e Jeff de Luca, no ano de 1997 em um projeto para o *United Overseas Bank*. A metodologia combina as principais práticas de outras metodologias ágeis com uma abordagem centrada no modelo de objetos (GIUFFRA e VILAIN, 2010).

Segundo Heptagon (2011), a FDD chama a atenção pelas seguintes características:

- Resultados úteis a cada duas semanas ou menos;
- Blocos bem pequenos de funcionalidade valorizada pelo cliente, chamados "Features";
- Planejamento detalhado e guia para medição;
- Rastreabilidade e relatórios com incrível precisão;
- Monitoramento detalhado dentro do projeto, com resumos de alto nível para clientes e gerentes, tudo em termos de negócio;

- Fornece uma forma de saber, dentro dos primeiros 10% de um projeto, se o plano e a estimativa são sólidos.

A FDD é composta por apenas duas fases: “Concepção e Planejamento” e “Construção”. A fase de Concepção e Planejamento tem como foco principal compreender toda a regra de negócio envolvida, mapear as funcionalidades necessárias e planejar o desenvolvimento de todas as funcionalidades. Esta fase está dividida em três processos: Desenvolver um Modelo Abrangente (DMA), Construir a Lista de Funcionalidades (CLF) e Planejar por Funcionalidade (PPF). Já a fase de Construção visa realmente especificar, implementar e testar as funcionalidades planejadas na fase anterior de maneira interativa e incremental. Esta fase está dividida em dois processos: Detalhar por Funcionalidade (DPF) e Construir por Funcionalidade (CPF).

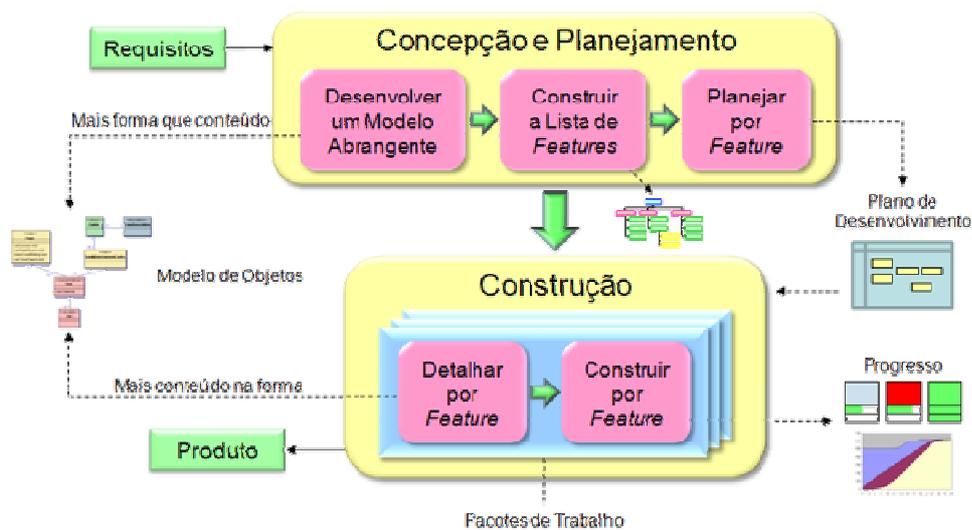


Figura 6 - Ciclo de Vida do FDD

Fonte: Heptagon(2011)

Iniciando a fase de Concepção e Planejamento é executado o processo DMA, onde é realizado todo um estudo sobre a regra de negócio do escopo em questão, definindo quais objetos fazem parte da aplicação e quais suas ligações, levando em conta mais a forma dos objetos do que o conteúdo. Dessa forma é criado um modelo de objetos utilizando a UML(*Unified Modeling Language*) para documentá-lo.

Em sequência é executado o processo CLF, onde são identificadas as funcionalidades da aplicação a partir do conhecimento adquirido no processo DMA.

As funcionalidades devem ser mapeadas em áreas e atividades de negócio, formando assim a lista categorizada de funcionalidades.

Para finalizar a fase de Concepção e Planejamento é executado o processo PPF, onde são analisadas todas as funcionalidades definidas no processo CLF, identificando as dependências entre as mesmas. É definida uma sequência de desenvolvimento das funcionalidades, estabelecendo um plano geral de desenvolvimento com prazos de entrega previstos.

Após a fase de Concepção e Planejamento, se dá início a fase de Construção executando o processo DPF e CPF de maneira interativa e incremental, conforme o plano de desenvolvimento definido no processo PPF.

No processo DPF é especificado cada funcionalidade refinando o modelo de objetos definido no processo DMA. O objetivo é especificar e documentar as funcionalidades para facilitar sua implementação e teste no processo seguinte.

Finalizando a fase de Construção é executado o processo CPF, onde é desenvolvida e testada as funcionalidades especificadas no processo DPF. Ao final deste processo é entregue ao cliente a funcionalidade pronta para uso.

Maiores detalhes sobre os processos do FDD podem ser obtidos em Retamal (2008).

4.2 SCRUM

O SCRUM é uma metodologia ágil de desenvolvimento de software que busca lidar com o caos de forma empírica, onde em ambientes complexos, um "processo bem definido" tem sua probabilidade sucesso drasticamente reduzida quando aplicado ao desenvolvimento de um produto. Uma das principais idéias do SCRUM para atacar este problema é a implantação de um controle descentralizado, que é capaz de lidar mais eficientemente com cenários pouco previsíveis (SCHWABER, 2004).

No SCRUM, um projeto se inicia com uma visão do produto que será desenvolvido, contendo a lista das características e funcionalidades do produto estabelecidas pelo cliente, gerando assim o *Product Backlog*. (MARÇAL *et. all*, 2007).

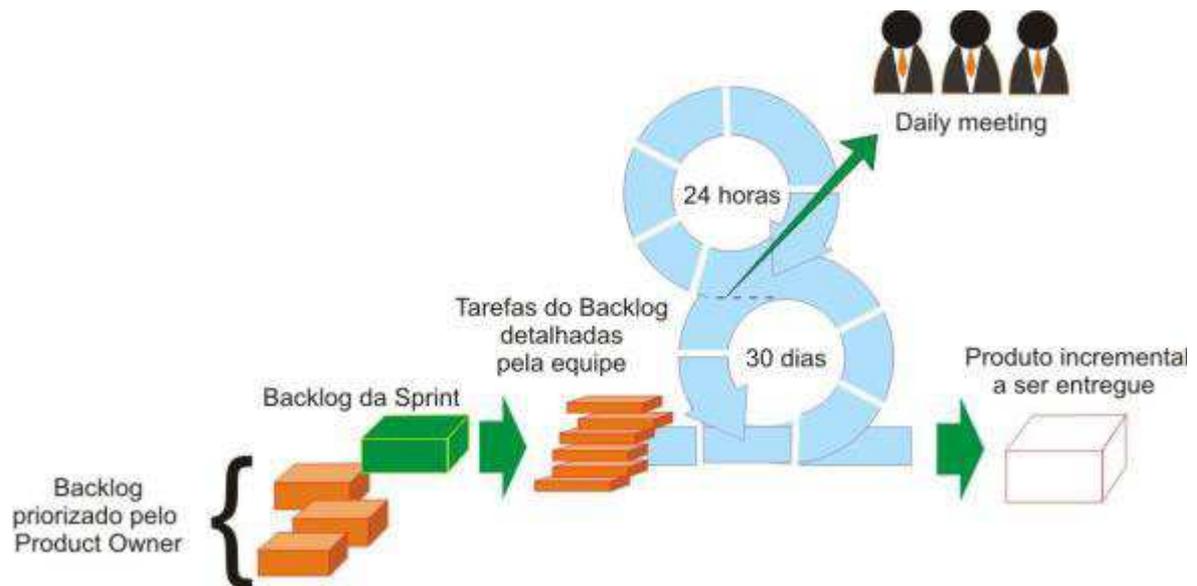


Figura 7 - Ciclo de vida do SCRUM

Fonte: Salaberri(2008)

O ciclo de vida do SCRUM tem o seu progresso baseado em um série de iterações, tendo em média uma duração de 2 a 4 semanas, nas quais são chamadas de *Sprints*. Antes de cada *Sprint*, realiza-se uma reunião de planejamento (*Sprint Planning Meeting*) onde a equipe de desenvolvedores tem contato com o cliente, também chamado de *Product Owner*, para priorizar o trabalho que precisa ser feito, selecionando e estimando as tarefas que a equipe poderá realizar dentro da *Sprint*, gerando assim o *Sprint Backlog*. A próxima fase é a execução da *Sprint*, onde a equipe controla o andamento do desenvolvimento realizando reuniões diárias rápidas (*Daily Meeting*), não mais que quinze minutos de duração, observando o seu progresso usando um gráfico chamado *Sprint Burndown*. Ao final de cada *Sprint*, é realizada uma revisão no produto entregue para verificar se tudo realmente foi implementado. Ao final da *Sprint*, deve-se realizar uma reunião de revisão (*Sprint Review*), onde a equipe demonstra o produto gerado na *Sprint* e valida se o objetivo foi atingido. Logo em seguida, realiza-se a reunião de retrospectiva (*Sprint Retrospective*), uma reunião de lições aprendidas, com o objetivo de melhorar o processo ou o produto para a próxima *Sprint* (PEREIRA, TORREÃO e MARÇAL, 2007).

4.3 Adaptação do Processo

Para desenvolvimento de toda a solução será seguido o processo FDD, adaptando algumas tarefas e artefatos, incluindo características do SCRUM. Como mostrado anteriormente, o FDD possui cinco processos divididos em duas fases, no qual a fase de concepção e planejamento já foi executada, e será relatada mais adiante. A Figura 8 apresenta os processos adaptados das duas fases, bem como os artefatos gerados em cada processo.

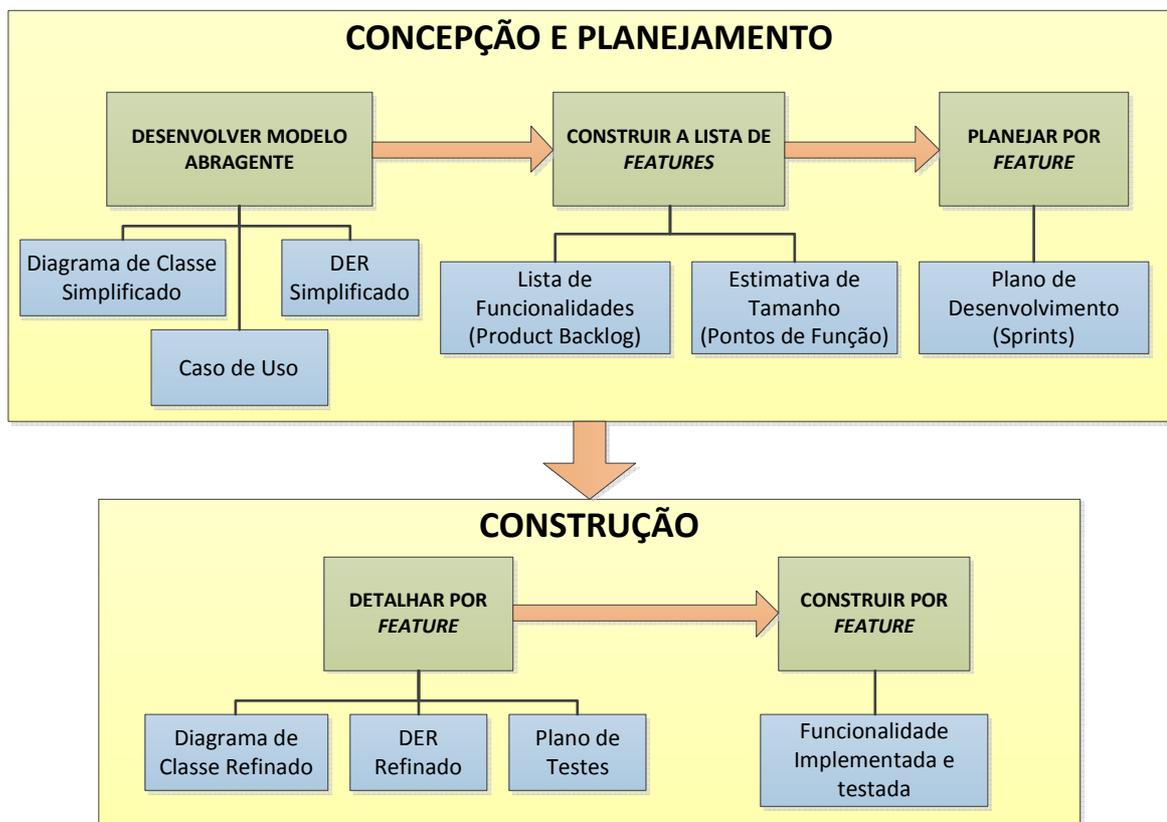


Figura 8 - Processo Adaptado

Como dito anteriormente, já foram executadas todos os processos da fase de concepção e planejamento. Esta fase foca no planejamento de todo o desenvolvimento, desde o entendimento da regra de negócio, definição dos requisitos e planejamento da construção das funcionalidades.

No processo “Criar um Modelo Abrangente”, no qual possui o objetivo de entender o domínio da solução, foi realizado vários estudos sobre a técnica de análise de pontos de função, colecionando artigos e materiais que serão úteis em todo o processo de desenvolvimento. Para mapear a regra de negócio foi desenvolvido um diagrama de classe simplificado (vide apêndice A e B) apenas

mostrando as entidades e suas ligações, e também o diagrama de entidade e relacionamento (vide apêndice C e D), transformando as classes em objeto relacional. Para que as principais funcionalidades da solução começassem a tomar forma, foi desenvolvido diagramas de caso de uso para auxílio do entendimento do contexto da solução (vide apêndice E).

No processo “Construir a lista de *Features*” foram levantados os requisitos de acordo com a modelagem geral da regra de negócio, mapeando-os em áreas de negócio gerando o “*Product Backlog*” em uma estrutura analítica mostrada na Figura 2 no capítulo três. Com os diagramas gerados no processo anterior, foi possível realizar também uma estimativa de tamanho das funcionalidades, utilizando a própria técnica de pontos de função, onde será mostrada no capítulo 6.

Com as funcionalidades definidas e mensuradas, foi executado o processo “Planejar por *Features*”, onde foi analisado as dependências entre as funcionalidades, definindo a sequência de desenvolvimento. Dessa forma foi possível planejar as *Sprints*, alocando as funcionalidades de acordo com a sequência de desenvolvimento, compondo o *Sprint Backlog*. Foi definida uma previsão de início e término, e o objetivo que se espera alcançar ao terminar cada *Sprint*, gerando assim o plano de desenvolvimento de toda fase de construção. Foram planejadas três *Sprints* para desenvolvimento de todas as funcionalidades, sendo que cada *Sprint* terá duração de quatorze dias de trabalho, dividindo quatro dias para detalhar as funcionalidades e dez dias para construí-las e testá-las. Porém, foi analisado que antes de iniciar a implementação das funcionalidades, será necessário preparar o ambiente de desenvolvimento para obter uma melhor produtividade. Dessa forma, foi planejada mais uma *Sprint* de 14 dias para desenvolver o layout gráfico da solução, definir padrões para codificação, criar classes de conexão com banco de dados, dentre outras atividades que servirão para estruturar o projeto. Essa *Sprint* por sua vez não terá duas etapas como as demais, pois seu foco não é implementar funcionalidades, sendo apenas para estruturação do projeto. O plano de desenvolvimento de cada *Sprint* pode ser encontrado nos apêndices F, G, H e I. A Figura 9 apresenta o plano geral de desenvolvimento.

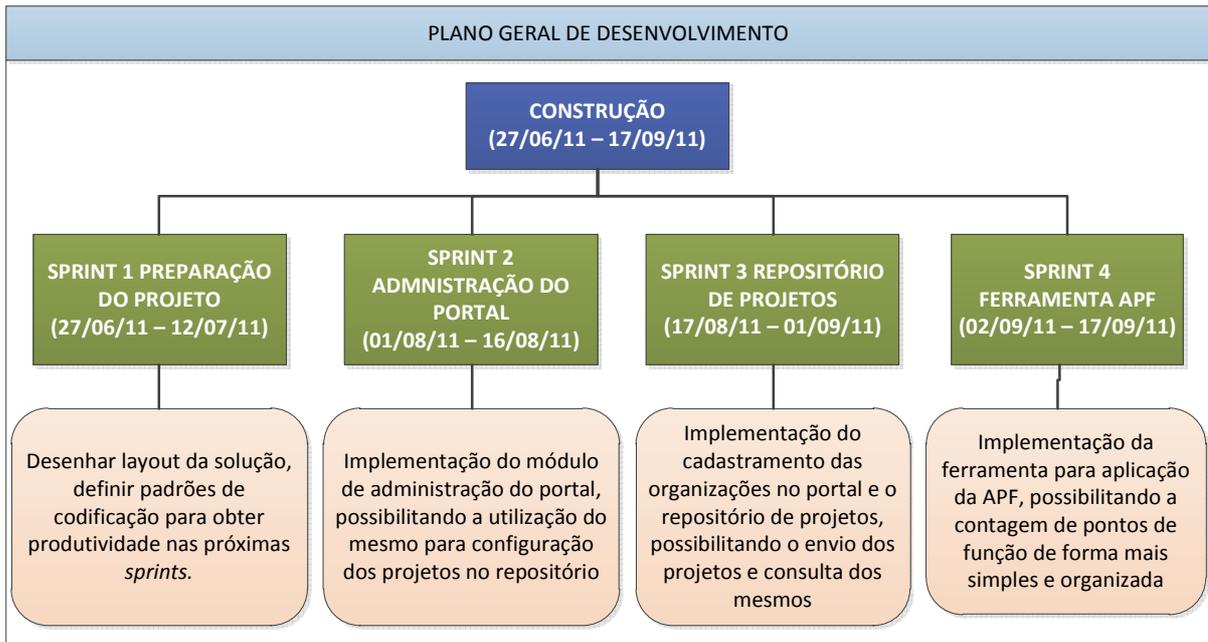


Figura 9 - Plano de Desenvolvimento

Com o plano de desenvolvimento, será possível iniciar a fase de construção, seguindo cronograma que será mostrado no capítulo 7. A fase de construção possui dois processos : “Detalhar por *Feature*” e “Construir por *Feature*”; tendo o objetivo de documentar e implementar as funcionalidades.

No processo “Detalhar por *Feature*”, será detalhado os requisitos da *Sprint*, atualizando e refinando os diagramas gerados no processo “Desenvolver Modelo Abrangente”. Também será criado um plano de teste para que seja possível a realização de testes de maneira mais organizada e eficaz. Dessa forma será possível estimar todas as atividades necessárias para implementação das funcionalidades, gerando assim o “*Sprint Backlog*” completo, podendo dar início a construção das funcionalidades.

No processo “Construir por *Feature*”, será realmente implementada a solução, utilizando as tecnologias e ferramentas citadas no capítulo 5, sendo gerada a cada *Sprint* uma parte de toda solução. Neste processo também será realizado o testes funcionais, seguindo o plano de teste gerado no processo anterior.

De modo geral, na *Sprint* 1 será dado início a preparação do projeto, onde será criado o layout da solução, arquivos de estilos e imagens do layout, classes de conexão com o banco de dados, e outras classes que serão úteis para a implementação de todo o projeto. Como dito anteriormente, essa primeira *Sprint* será diferente das demais, sendo executada em uma única etapa, tendo como objetivo

estruturar o projeto em nível de implementação para que o mesmo tenha alta produtividade nas demais *Sprints*, onde o foco será realmente só implementar as funcionalidades. Na *Sprint 2* será implementado todas as funcionalidades do módulo administração do portal, dando base para a inserção da configuração dos projetos dentro do repositório. Na *Sprint 3* será implementado o cadastramento das organizações no portal e o repositório de projetos, possibilitando a submissão e a realização de buscas dos mesmos. Por fim, na *Sprint 4* será desenvolvida a ferramenta de aplicação da técnica de pontos de função. A partir da *Sprint 2*, o produto gerado já será publicado na web para realização de testes com usuários reais.

5. TECNOLOGIAS E FERRAMENTAS

Neste capítulo estão descritas as tecnologias e ferramentas que serão utilizadas em todo o processo de desenvolvimento.

- **ASP.NET:** é uma plataforma de desenvolvimento de aplicações web criada pela Microsoft. Atualmente o *framework* está na versão 4.0, lançada em 2010. Porém para desenvolvimento da solução proposta será utilizado os recursos do *framework* 3.5, utilizando a linguagem de programação C#.

- **Microsoft SQL Server:** é um sistema gerenciador de bando de dados relacional criado pela Microsoft. Será utilizado neste projeto por ser robusto, ou seja, possui a capacidade de gerenciar uma grande quantidade de dados. Será utilizada a versão 2008 deste banco de dados.

- **Microsoft SQL Server Management Studio Express:** é uma ferramenta utilizada para facilitar a configuração, manutenção e administração do banco de dados SQL Server. Será utilizada a versão *Express* da ferramenta, pois inicialmente a mesma será utilizada apenas para auxiliar no desenvolvimento da solução.

- **Microsoft Visual Studio 2010:** é uma IDE (*Integrated Development Environment*) criada pela Microsoft, utilizada para agilizar o processo de desenvolvimento de aplicações. É a melhor IDE disponível no mercado para a plataforma .NET, e será utilizada para desenvolvimento de toda a solução proposta.

- **Microsoft Visio 2010:** é um programa desenvolvido pela Microsoft para o desenvolvimento de vários tipos de diagramas. No caso deste projeto, será utilizado para criação dos diagramas de classe, diagrama de entidade e relacionamento, entre outros diagramas.

- **IIS – Internet Information Service:** é um servidor web criado pela Microsoft, que será utilizado para rodar a aplicação ASP.NET que será desenvolvida.

- **CSS - Cascading Style Sheets :** são folhas de estilos para páginas *Web*, adicionadas ou não ao código HTML (*HyperText Markup Language*). Com CSS, pode-se criar folhas de estilo em um arquivo separado do código, o que será bastante explorado neste trabalho.

- **JavaScript:** É uma linguagem de programação interpretada do lado do cliente, ou seja, é executada no *browser* do cliente, onde será utilizada para: validar

formulários, cálculos complexos, manipulação dos dados na página, entre outros, tornando as páginas mais interativas e dinâmicas.

- **Astah Community:** é uma ferramenta gratuita para criação de diagramas UML. Neste projeto, essa ferramenta será utilizada para criação dos diagramas de caso de uso.

- **Tortoise SVN:** também conhecido como *Subversion*, é um sistema de controle de versões. Será utilizado para controlar as versões de codificação, possibilitando a rastreabilidade de codificação e outros artefatos.

6. ESTIMATIVA DE TAMANHO DA SOLUÇÃO

Neste capítulo será apresentada a estimativa de tamanho da solução, onde foi aplicada a técnica de análise de pontos de função, tendo o foco na visão do usuário. O Quadro 1 apresenta o total de cada elemento da análise, mostrando sua complexidade.

	Baixa	Média	Alta	Total PF
Arquivo Lógico Interno	16 * 7 = 112	1 * 10 = 10	0 * 15 = 0	122
Arquivo Interface Externa	0 * 5 = 0	0 * 7 = 0	0 * 10 = 0	0
Entrada Externa	18 * 3 = 54	15 * 4 = 60	17 * 6 = 102	216
Saída Externa	0 * 4 = 0	0 * 5 = 0	4 * 7 = 28	28
Consulta Externa	6 * 3 = 18	0 * 4 = 0	3 * 6 = 18	60
			Total PFNA:	426

Quadro 1 - Estimativa de Tamanho em Pontos de Função

A contagem realizada é do tipo “Projeto de Desenvolvimento”, ou seja, apresenta apenas uma estimativa inicial de tamanho da solução. Ao detalhar melhor cada funcionalidade, novos campos e ações do usuário poderão surgir e conseqüentemente poderão alterar o tamanho da solução. Dessa forma, após o desenvolvimento de todas as funcionalidades será realizada uma contagem do tipo “Aplicação (*baseline*)” para se obter um tamanho mais preciso de toda a solução implementada. Essa contagem será postada no próprio repositório como primeiro projeto a ser submetido e estará disposta na monografia.

7. CRONOGRAMA

Neste capítulo será apresentado o cronograma previsto do projeto, desde seu início.

	Início	Termínio	Duração
Concepção e Planejamento	04/abr/11	14/mai/11	36 dias
+ Desenvolver Modelo Abrangente	04/abr/11	16/mai/11	12 dias
+ Construir Lista de <i>Features</i>	18/abr/11	30/abr/11	12 dias
+ Planejar por <i>Feature</i>	02/mai/11	14/mai/11	12 dias
Escrita da Proposta	04/abr/11	31/mai/11	50 dias
Construção	27/jun/11	17/set/11	56 dias
+ <i>Sprint 1 – Preparação do Projeto</i>	27/jun/11	12/jul/11	14 dias
+ <i>Sprint 2 – Administração do Portal</i>	01/ago/11	16/ago/11	14 dias
+ Detalhar por <i>Feature</i>	01/ago/11	04/ago/11	4 dias
+ Construir por <i>Feature</i>	05/ago/11	16/ago/11	10 dias
+ <i>Sprint 3 – Repositório de Projetos</i>	17/ago/11	01/set/11	14 dias
+ Detalhar por <i>Feature</i>	17/ago/11	22/ago/11	4 dias
+ Construir por <i>Feature</i>	23/ago/11	01/set/11	10 dias
+ <i>Sprint 4 – Ferramenta APF</i>	02/set/11	17/set/11	14 dias
+ Detalhar por <i>Feature</i>	02/set/11	06/set/11	4 dias
+ Construir por <i>Feature</i>	07/set/11	17/set/11	10 dias
Contagem dos PF da Aplicação para postagem no Repositório	19/set/11	24/set/11	6 dias
Escrita da Monografia	26/set/11	19/nov/11	48 dias

Quadro 2 - Cronograma do Projeto

Como dito anteriormente no capítulo 4, as atividades da fase de concepção e planejamento já foram executadas dentro do cronograma previsto. A próxima atividade a ser executada já dá início à fase de construção do projeto. Após o desenvolvimento da solução, será realizada a contagem dos pontos de função para postagem no próprio repositório de projeto e dado início a escrita da monografia. No apêndice J é apresentado o gráfico de Gantt deste cronograma.

REFERÊNCIAS

AGUIAR, M. Pontos de função ou pontos de caso de uso? Como estimar projetos orientados a Objetos. **Developer's Magazine** v. 7, n. 77. Jan., 2003.

ANDRADE, E. L. P. **Pontos de Caso de Uso e Pontos de Função na Gestão de Estimativa de Projetos de Software Orientados a Objetos**. Dissertação de Mestrado, Universidade Católica de Brasília (UCB), 2004.

ANDRADE, E. L. P.; OLIVEIRA K. M. **Aplicação de Pontos de Função e Pontos de Casos de Uso de Forma Combinada no Processo de Gestão de Estimativa de Tamanho de Projetos de Software Orientado a Objetos**. 2004. Disponível em: <http://www.ip.pbh.gov.br/ANO7_N1_PDF/IP7N1_andrade.pdf>. Acesso em: 12 maio 2011.

BARCELLOS, M. P. **Planejamento de Custos em Ambientes de Desenvolvimento de Software Orientados à Organização**. Rio de Janeiro 2003 VIII, 208 f. Dissertação (Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, COPPE, 2003.

CALAZANS, A. T. S.; OLIVEIRA, M. A. L. **Avaliação de estimativa de tamanho para Projetos de Manutenção de software**. 2005. Disponível em: <http://www.angelicatoffano.pro.br/upload_arquivos/pt/Artigo%20ASSE%202005_2.pdf>. Acesso em: 25 maio 2011.

CAVALCANTE, R. E. **Um estudo de avaliação da eficiência e eficácia na utilização da análise de pontos de função no desenvolvimento de software do serviço público**. 2010. 55 f. Trabalho de Conclusão de Curso (Gestão e Administração Pública). – Uniceub, Brasília, DF, 2010. Disponível em: <<http://bdjur.stj.jus.br/dspace/handle/2011/34477>>. Acesso em: 23 abr. 2011

DIAS, R. **Análise por Pontos de Função: Uma Técnica para Dimensionamento de Sistemas de Informação**. 2004. Disponível em: <<http://revistas.facecla.com.br/index.php/reinfo/article/view/134/28>>. Acesso em: 14 maio 2011.

GARMUS, D., HERRON, D. **Function Point Analysis: Measurement Practices for Successful Software Projects**, Boston: Addison Wesley, 2001.

GIUFFRA, C. E.; VILAIN, P. **Modelagem da Interação do Usuário no Desenvolvimento Ágil**. 2010. Disponível em: <<http://periodicos.unesc.net/index.php/sulcomp/article/viewArticle/245>>. Acesso em: 14 maio 2011.

GONÇALVES, E. S.; REIS FILHO, H. B. **Ferramenta para Gerenciamento de Requisitos em Metodologias Ágeis**. 2008. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/view/4593>>. Acesso em: 14 maio 2011.

HAZAN, C.; STAA, A. V. **Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software**. 2005. Disponível em: <ftp://139.82.16.194/pub/docs/techreports/05_04_hazan.pdf>. Acesso em: 25 maio 2011.

HEPTAGON. **FDD – Feature Driven Development**, 2011. Disponível em: <<http://www.heptagon.com.br/fdd>>. Acesso em: 12 maio 2011.

IFPUG. **Counting Practices Manual**. Version 4.2, 2004.

MARÇAL, A. S. C. *et al.* **Estendendo o SCRUM segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI**, 2007. Disponível em: <http://www.followscience.com/library_uploads/28adceaba40574ef816b7893f071bc01/529/estendendo_o_scrum_segundo_as_areas_de_processo_de_gerenciamento_d_e_projetos_do_cmmi.pdf>. Acesso em: 15 maio 2011.

PEREIRA, P.; TORREÃO, P.; MARÇAL, A. S. **Entendendo Scrum para Gerenciar Projetos de Forma Ágil**. 2007. Disponível em: <<http://www.lapjor.cce.ufsc.br/elgg/html/pg/file/benhur/read/349/entendendo-scrum-para-gerenciar-projetos-de-forma-gil>> Acesso em: 15 maio 2011.

RETAMAL, A. M. **Feature-Driven Development Descrição dos Processos**. 2008. Disponível em: <<http://www.heptagon.com.br/files/FDD-Processos.pdf>>. Acesso em: 14 maio 2011.

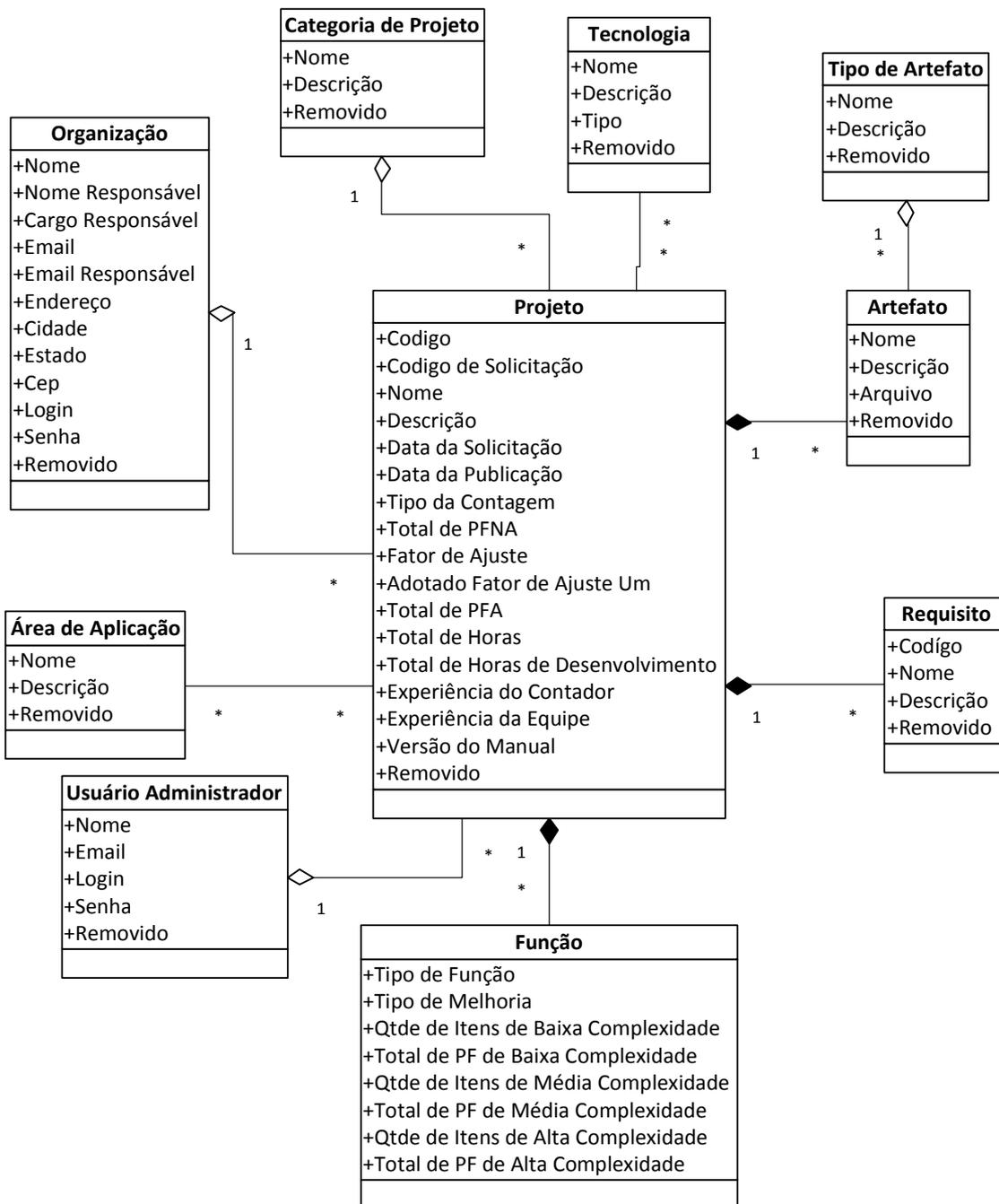
SALABERRI, P. S. **Um Modelo de Processo para o Desenvolvimento de Aplicações Interativas em TV Digital**. 2008. 73p. Trabalho acadêmico – Curso de Bacharelado em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

SCHWABER, K. **Agile Project Management With Scrum**. 1ª ed. Redmond, Washington: Microsoft Press, 2004.

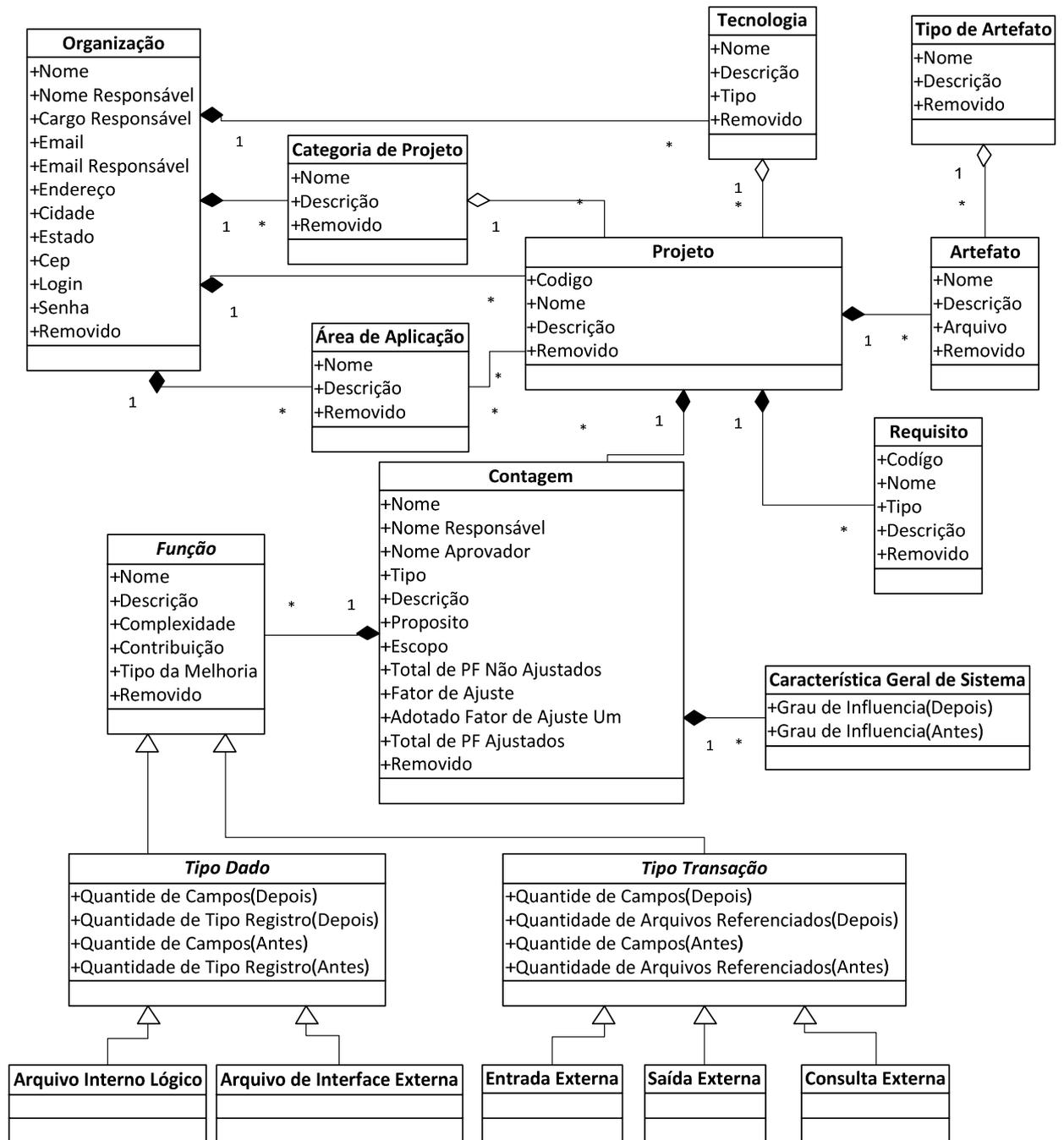
TAVARES, H. C. A. B.; CARVALHO A. E. S.; CASTRO J. F. B. **Medição de Pontos por Função a Partir da Especificação de Requisitos**. Pernambuco, 2004. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_wer02/tavares.pdf>. Acesso em: 15 maio 2011.

VAZQUEZ, C.; SIMÕES, G.; ALBERT, R. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software**. 10ª ed. São Paulo: Érica, 2010.

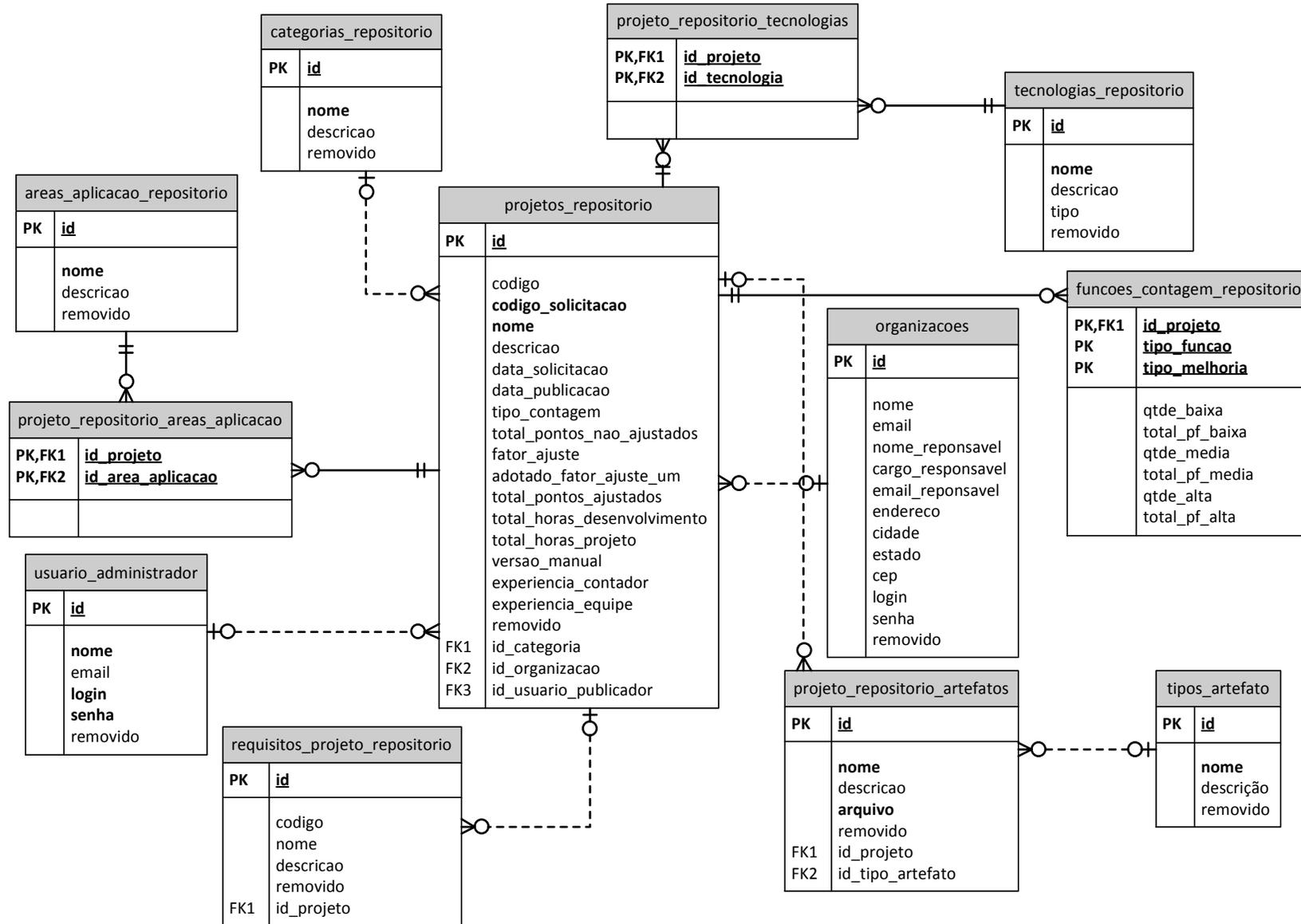
APÊNDICE A – Diagrama de Classes – Repositório de Projetos



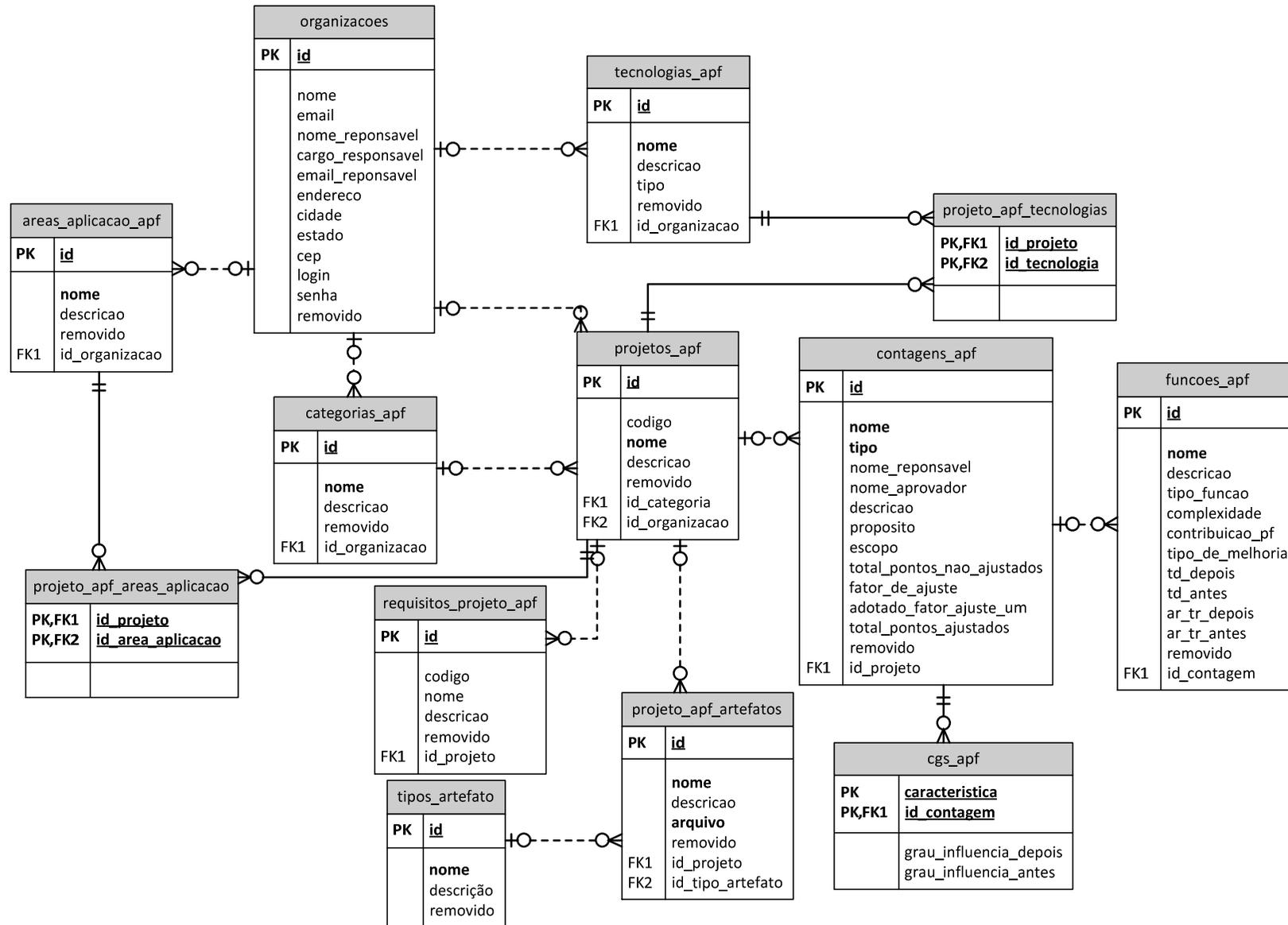
APÊNDICE B – Diagrama de Classes – Ferramenta APF



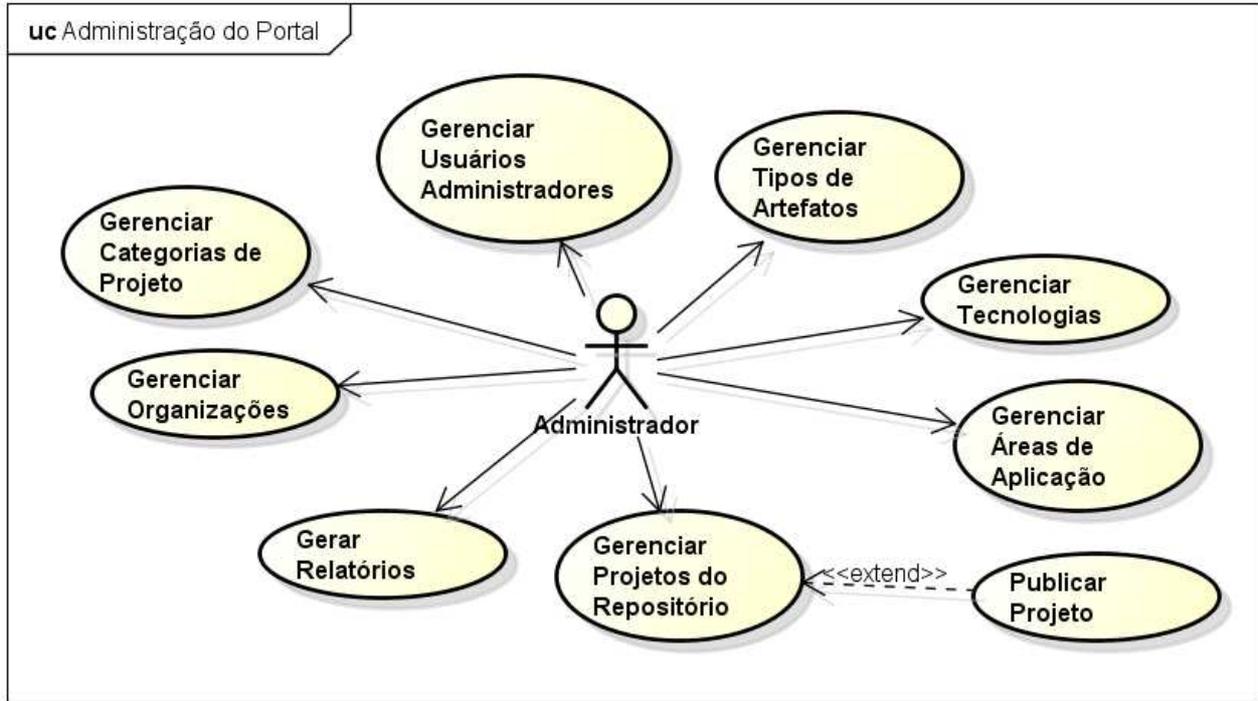
APÊNDICE C – Diagrama de Entidade e Relacionamento – Repositório de Projetos



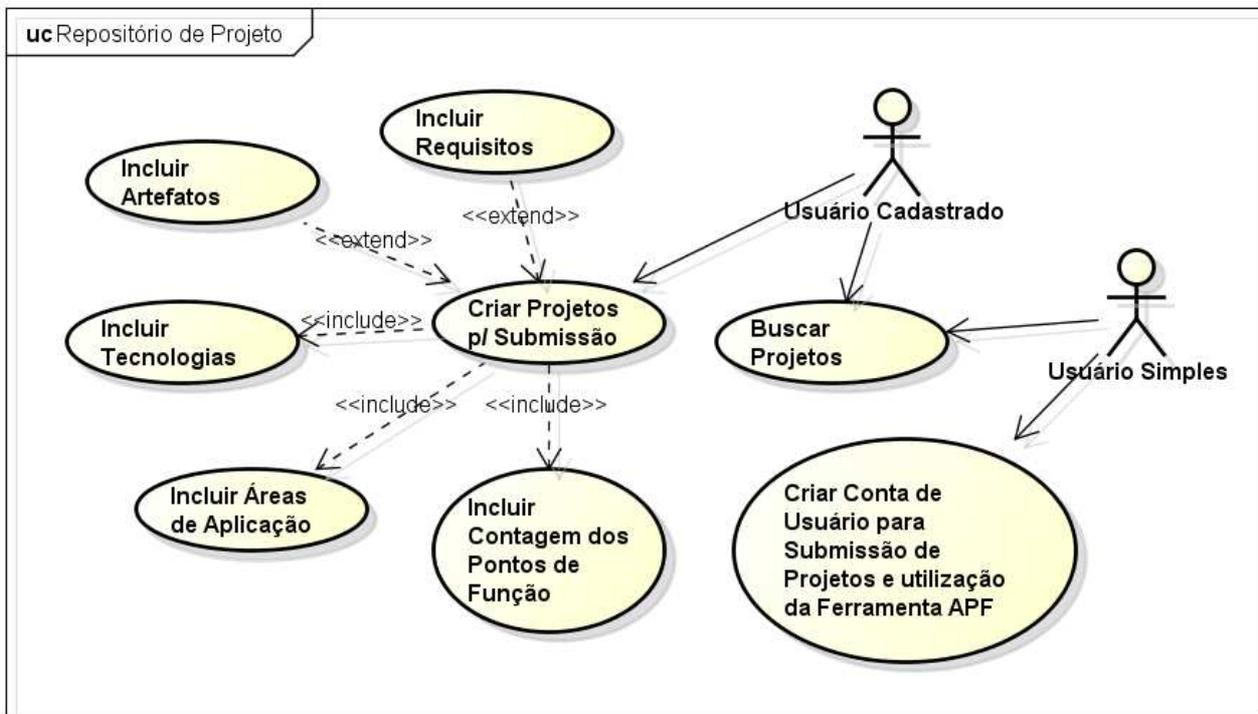
APÊNDICE D – Diagrama de Entidade e Relacionamento – Ferramenta APF



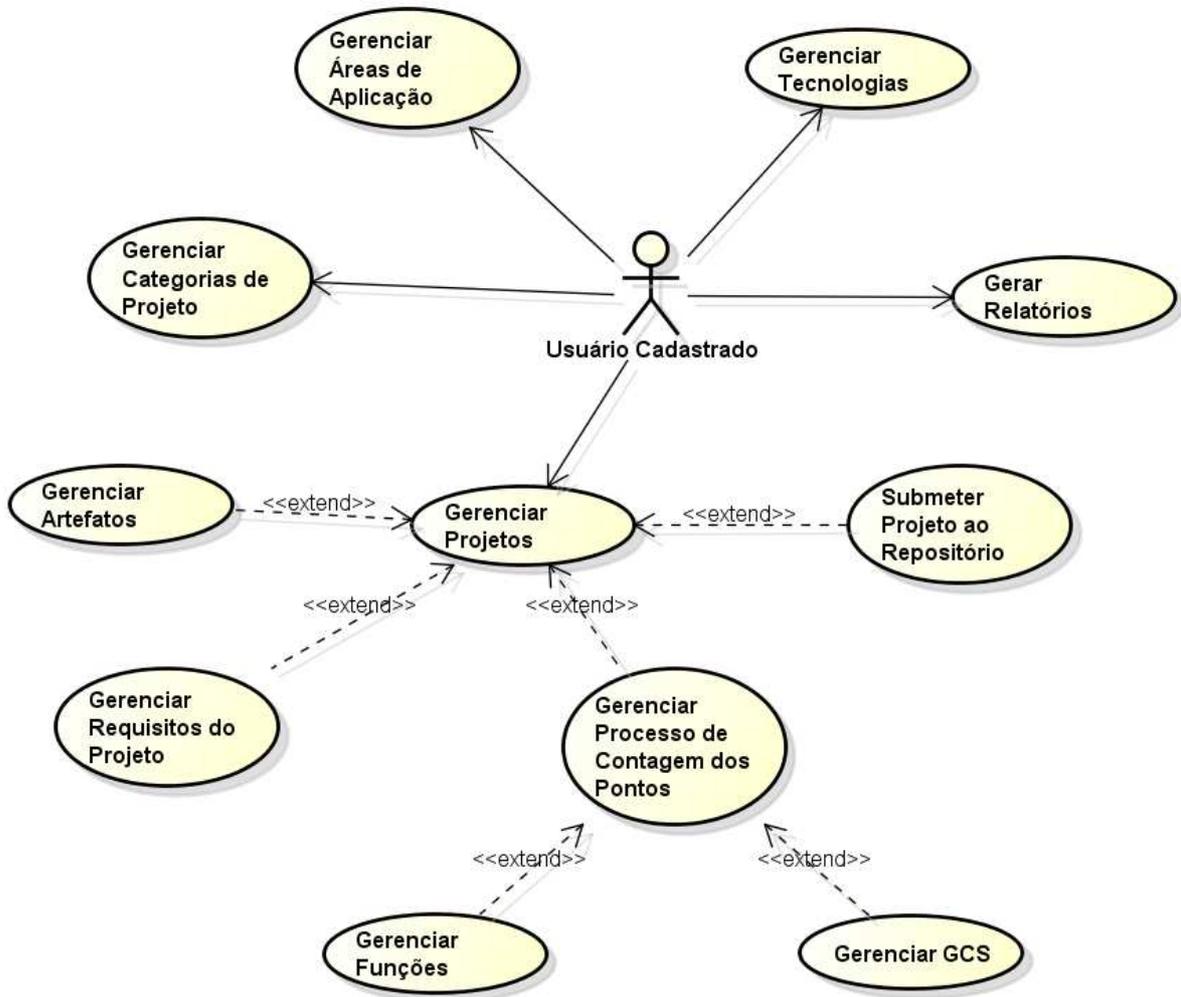
APÊNDICE E – Diagrama de Caso de Uso



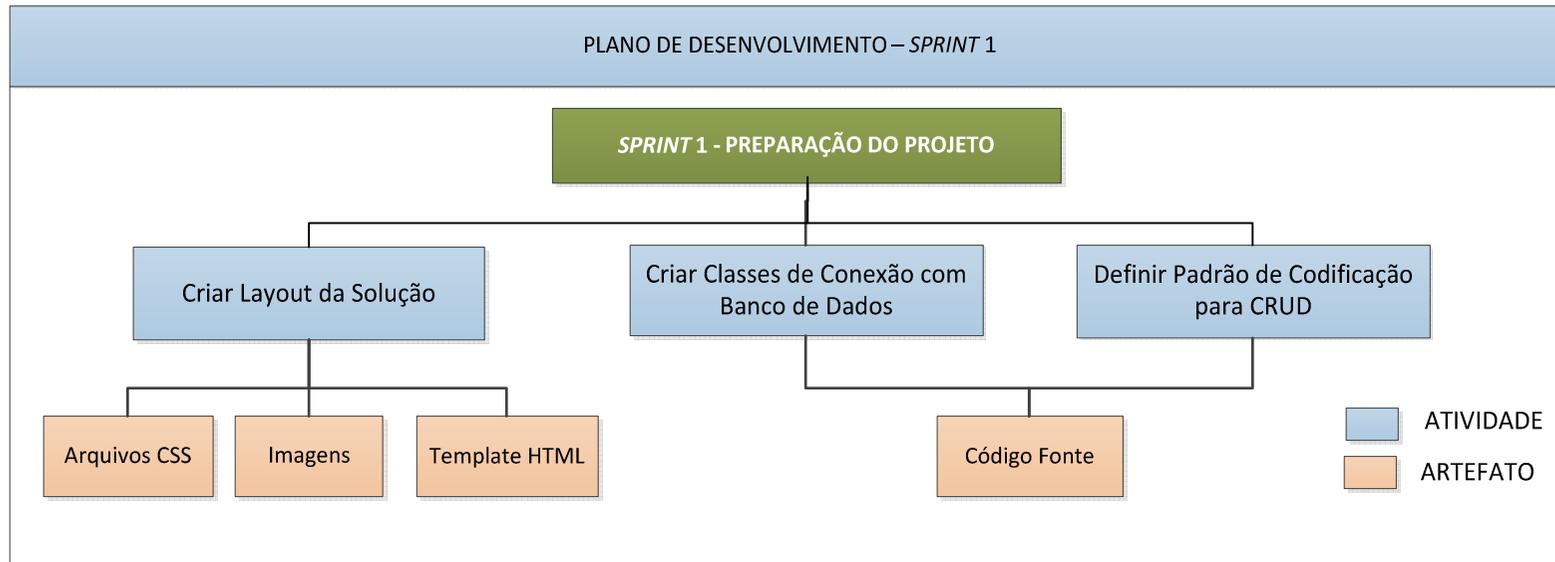
powered by astah



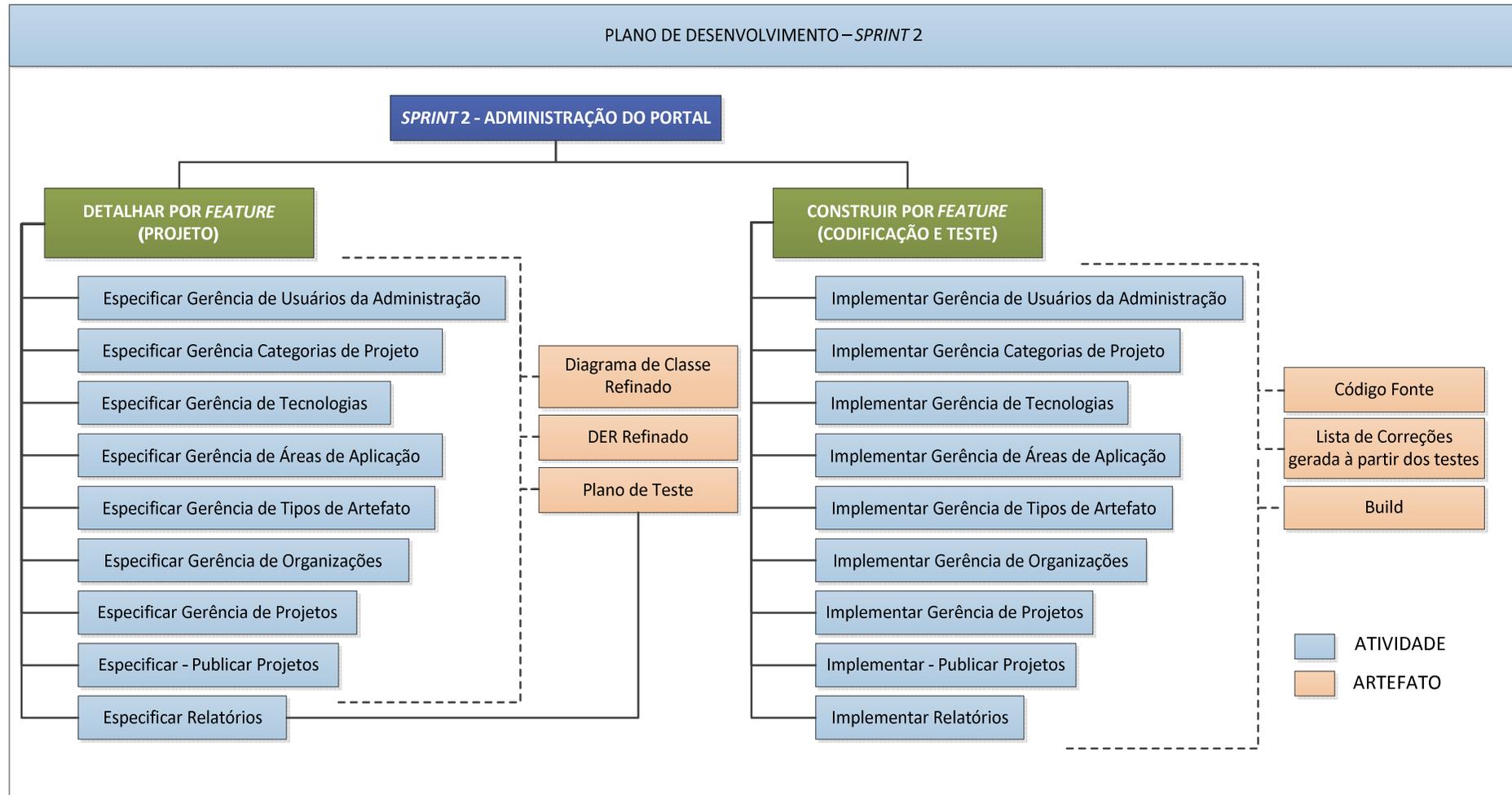
powered by astah



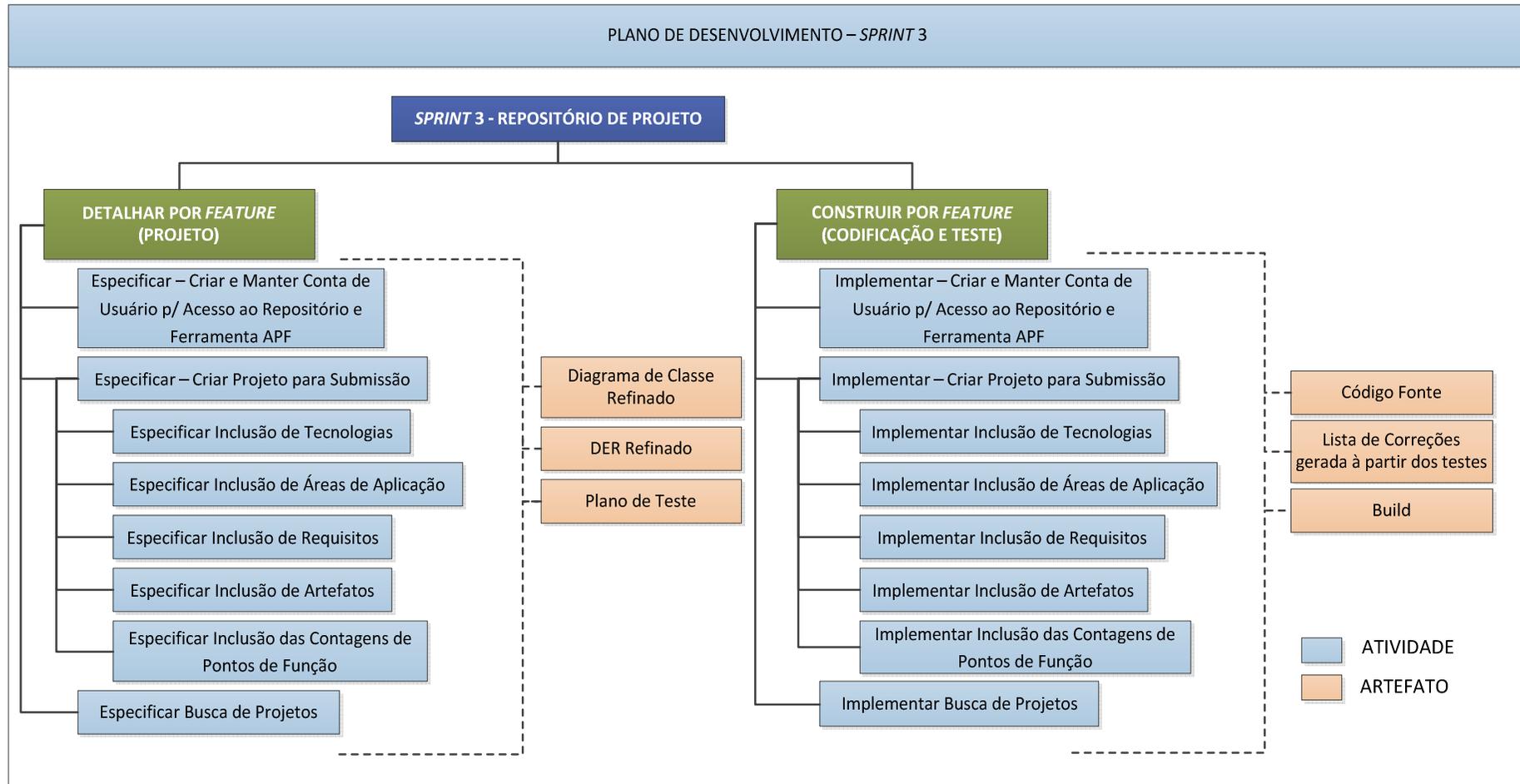
APÊNDICE F – Plano de Desenvolvimento – *Sprint* 1



APÊNDICE G – Plano de Desenvolvimento – *Sprint 2*



APÊNDICE H – Plano de Desenvolvimento – *Sprint 3*



APÊNDICE I – Plano de Desenvolvimento – *Sprint 4*

