Metodologia de Desenvolvimento de Sistemas

Versão 1 – 09/09/2015

1 – Introdução

1.1 – Objetivos do documento

A Metodologia de Desenvolvimento de Sistemas tem por objetivo estabelecer critérios e padrões a serem observados durante todo o processo de desenvolvimento de sistemas pelo Departamento de Sistemas de Informações da STI – UNIVASF.

O processo de desenvolvimento pode ser compreendido desde a solicitação de novos sistemas ou ajustes nos sistemas existentes, estudo de viabilidade, levantamento das necessidades do usuário, análise das informações existentes e a serem processadas pelo sistema/módulo a ser desenvolvido, planejamento, execução, testes, ajustes e homologação do sistema para uso.

2 – Solicitações

2.1 – Desenvolvimento de novo sistema

A solicitação deverá ser feita via memorando direcionado a STI e encaminhada através do sistema STD (www.sistemas.univasf.edu.br/std).

Os solicitantes devem descrever os problemas enfrentados e os objetivos a serem alcançados, inclusive funcionalidades e propriedades que os mesmos esperam encontrar no sistema.

2.1.1 – Artefatos e fluxo

Artefato de entrada: Memorando de solicitação de desenvolvimento de sistema (Item 2.1).
Fluxo: Termo de abertura (Item 3).

2.2 – Desenvolvimento de novo módulo ou grandes modificações

A solicitação deverá ser feita via memorando direcionado a STI e encaminhada através do sistema STD (www.sistemas.univasf.edu.br/std).

Os solicitantes devem relatar e descrever os objetivos, funcionalidades e propriedades que os mesmos esperam encontrar no sistema.

2.2.1 – Artefatos e fluxo

Artefato de entrada: Memorando de solicitação de desenvolvimento de sistema (Item 2.2).
Fluxo: Análise e especificação de requisitos (Item 4).

2.3 – Pequenos ajustes e melhorias

A solicitação deverá ser feita via sistema de suporte da STI (www.sistemas.univasf.edu.br/suporte).

Serão considerados pequenos ajustes e melhorias no sistema aquelas alterações que não implicarão a implementação de novas funcionalidades, mas, tão somente, incrementarão de informações ou reorganização das funcionalidades já existentes.

2.3.1 – Artefatos e fluxo

Fluxo: Desenvolvimento (Item 6).

O acompanhamento do chamado deverá ser realizado através do sistema de suporte até a finalização.

2.4 – Correções

A solicitação deverá ser feita via sistema de suporte da STI (www.sistemas.univasf.edu.br/suporte).

A ocorrência de “erro no sistema” será caracterizada quando o mesmo apresentar funcionamento/ação anômala àquela que se poderia esperar usualmente e/ou para a qual o mesmo foi desenvolvido.

2.4.1 – Artefatos e fluxo

Fluxo: Desenvolvimento (Item 6).

O acompanhamento do chamado deverá ser realizado através do sistema de suporte até a finalização.

3 – Termo de abertura

Para o desenvolvimento de novo sistema será elaborado o documento de Termo de Abertura com as seguintes informações:

. Identificação do sistema
. Requisitante
. Responsável
. Objetivo do projeto
. Público-Alvo
. Justificativa do projeto
. Alinhamento estratégico
. Priorização
. Recursos (Hardware, Software e Peopleware)
. Restrições
. Riscos
. Plano de comunicações
. Principais produtos do projeto
. Cronograma do projeto

3.1 – Artefatos e fluxo

Artefato de entrada: Memorando de solicitação de desenvolvimento de sistema (Item 2).
Artefato de saída: Termo de abertura.
Fluxo: Análise e especificação de requisitos (Item 4).

4 – Análise e especificação de requisitos

O setor de desenvolvimento receberá as solicitações e entrará em contato com o solicitante. Poderá ser agendado um encontro onde serão repassadas as orientações iniciais a fim de viabilizar o início do processo de desenvolvimento.

São especificados requisitos que devem constar no sistema para que o mesmo possa alcançar a sua finalidade desejada. Os requisitos serão analisados a fim de identificar a melhor forma de contemplá-los dentro do ambiente do sistema a ser desenvolvido.

Após a finalização da etapa de especificação e análise de requisitos, será elaborado um documento denominado “Acordo de Nível de Serviço”, o qual deverá ser assinado entre as partes envolvidas no processo de desenvolvimento do sistema (STi e solicitante).

4.1 – Artefatos e fluxo

Artefato de entrada: Memorando de solicitação de desenvolvimento de sistema (Item 2).
Artefato de entrada: Termo de abertura (Item 3).
Fluxo: Acordo de nível de serviço (Item 5).
Fluxo: Dados a Serem Utilizados pelo Sistema (Item 6)

Os artefatos “Acordo de nível de serviço” e “Dados a serem utilizados pelo sistema” estão intrinsecamente ligados e dependentes da etapa de análise e especificação de requisitos.

5 – Acordo de nível de serviço

O Acordo de Nível de Serviço tem por objetivo estabelecer critérios a serem observados na prestação do serviço de elaboração, implantação, alimentação e manutenção do Sistema.

Este documento deve apresentar as seguintes informações:

. Identificação do sistema

. Introdução

.. Objetivos do documento

.. Escopo do serviço

.. Visão geral do documento

. Participantes

.. Signatários

.. Responsabilidades

. Requisitos

.. Requisitos do negócio

.. Requisitos mínimos para funcionamento / disponibilidade do serviço

.. Dependência de outros serviços

. Funcionalidades esperadas

. Relatórios

. Objetivos do serviço

.. Indicadores e Metas para Avaliação da Funcionalidade do Serviço

.. Riscos

. Previsão de Custo de Desenvolvimento do Sistema pela STI

. Previsão de atendimento

. Validade do documento

. Termo de compromisso

5.1 – Artefatos e fluxo

Artefato de entrada: Memorando de solicitação de desenvolvimento de sistema (Item 2).
Artefato de entrada: Termo de abertura (Item 3).
Artefato de saída: Acordo de nível de serviço.
Fluxo: Desenvolvimento (Item 7).

6 – Dados a Serem Utilizados pelo Sistema

Este documento apresenta as entidades e atributos do sistema.

As entidades podem ser classificadas em tabelas, mais utilizadas para configuração, e em cadastros, que possuem maior movimentação.

Deverá ser utilizado o nome lógico da entidade, empregado no plural.

Para cada entidade deverão ser informadas as seguintes informações:

a) Atributo

Descrever o nome lógico do atributo.

b) Tipo de Informação

Tipo físico do atributo, exemplos:

. (código interno sequencial)
. N caracteres (2 caracteres, 100 caracteres, 2000 caracteres ...)
. 1 Caractere
. S => Sim
  N => Não
. F => Feminino
  M => Masculino
. Data
. Hora
. Inteiro
. Número decimal
. Imagem
. Calculado (resultante de cálculo ou função)

c) Obrigatório

O atributo será obrigatório quando não for possível ter um valor nulo.
Informar Sim ou Não.

d) Observação

Informar quando atributo for “Chave primária”.

Caso o atributo seja chave estrangeira, deverá informar a entidade estrangeira entre chaves. {Entidade}

Informar quando for obrigatário dependendo do valor de outro atributo. Descrever a condição.

Informar limites de valores quando houver possíveis mínimo e / ou máximo.

Informar se atributo deve ter valor único.

Outras observações.

6.1 – Artefatos e fluxo

Artefato de entrada: Memorando de solicitação de desenvolvimento de sistema (Item 2).
Artefato de entrada: Termo de abertura (Item 3).
Artefato de saída: Dados a serem utilizados pelo sistema.
Fluxo: Desenvolvimento (Item 7).

7 – Desenvolvimento

Todas as tarefas de desenvolvimento, inclusive de novos sistemas e novos módulos, são registradas no sistema de suporte da STI (www.sistemas.univasf.edu.br/suporte).

No caso de novos sistemas / módulos, os chamados serão registrados pelo gerente do projeto. Para ajustes e correções, os chamados devem ser registrados de preferência pelos usuários do sistema, podendo em casos especiais ser registrados pelo gerente do projeto, quando solicitação por e-mail, telefone ou diretamente no setor, sendo que nestes casos o gerente deve abrir o chamado em nome do solicitante.

Atualmente o Departamento de Sistemas de Informações da STI-UNIVASF desenvolve seus próprios sistemas usando por padrão a arquitetura cliente-servidor com utilização de servidores e aplicações web.

Como base para o desenvolvimento de sistemas, poderá ser utilizado como referência o SGS (Sistema de Gestão de Serviços) para serviços independentes ou o SGE (Sistema de Gestão de Eventos) para serviços integrados ao SGS.

O SGS gerencia os outros sistemas, podendo ser realizado o acesso a estes outros através de um único login ao SGS.

7.1 – Modelagem

A modelagem do banco de dados deverá ser realizada a partir do documento “Dados a serem utilizados no sistema”.

7.1.1 – Padrões para a modelagem de dados

O banco de dados utilizado atualmente é o PostgreSQL 9.3.

a) Usuário proprietário

Para cada projeto é criado um usuário, com o mesmo nome do projeto, que deverá ser o proprietário de todas as entidades no banco de dados.

b) Entidades

Nome lógico e/ou comentário conforme descrito no documento “Dados a serem utilizados no sistema”.

Nome físico da entidade no plural, com primeira letra de cada nome em maiúsculo. Exemplos: TiposAfastamentos e Afastamentos.

Informar o usuário proprietário para cada entidade.

c) Atributos

Nome lógico e/ou comentário conforme descrito no documento “Dados a serem utilizados no sistema”.

Nome físico do atributo no singular, com primeira letra de cada nome em maiúsculo, devendo ser adicionado um prefixo ao nome de acordo com o tipo de informação:

ID_ => Código de identificação (código interno sequencial)
CD_ => Código informado (não sequencial)
DS_ => Descrição (campos com caracteres em geral)
NM_ => Número (campos numéricos em geral)
DT_ => Data
HR_ => Hora
BA_ => Imagem

Exemplos: ID_TipoAfastamento, DS_TipoAfastamento, DT_Inicial e DS_Portaria.

Para atributos caracteres, informar CHARACTER quando o tamanho for sempre o mesmo, por exemplo para códigos, e informar CHARACTER VARYING para outros casos.

Informar o tamanho da informação, quando for o caso, e informar quando faz parte da chave primária (pk) e se obrigatório (not null), de acordo com o documento “Dados a serem utilizados no sistema”.

d) Sequências

Para atributos sequenciais (código interno sequencial) deverá ser informado o tipo SERIAL. Deverá ser gerada uma sequência para o atributo com nomenclatura entidade_atributo_seq (exemplo: tiposafastamentos_id_tipoafastamento_seq).

e) Relacionamentos

Os nomes dos relacionamentos devem ter a nomenclatura entidadepai_entidadefilha (exemplo: tiposafastamentos_afastamentos).

f) Índices

Para o PostgreSQL serão criados índices automaticamente para as chaves primarias e chaves estrangeiras.

Incluir na modelagem as chaves alternativas, se existirem, com a nomenclatura ak_entidade_atributos (exemplo: ak_tiposafastamentos_dstipoafastamento).

7.2 – Scripts

O script de criação das entidades, atributos, relacionamentos, sequências e índices são obtidos a partir da modelagem (Item 7.1).

Além deste, existem outros scripts para criação e configuração do banco de dados.

O arquivo checklist.txt, existente para cada projeto, apresenta a ordem de execução dos scripts:

. Criação do usuário, tablespace e database.
. Criação do schema.
. Criação das entidades.
. Alteração do proprietário das entidades e sequências.
. Geração de procedures e funções, inclusive objetos públicos.
. Geração das permissões do sistema.
. Geração das triggers.
. Geração das views.

Observação: Para cada projeto deverá existir um usuário simplificado, com menos permissões, que deverá ter um database própria em uma tablespace própria. Os scripts devem ser executados pelo usuário postgres (super usuário).

As funções e gatilhos de log (triggers) são geradas automaticamente a partir de aplicativo desenvolvido pela STI, utilizando a própria definição do banco de dados (entidades, atributos, relacionamentos, sequências e índices).

7.3 – Programação

A plataforma de sistemas atual utiliza linguagem PHP 5.3, rodando em servidor Apache 2, em conjunto com outras linguagens: CSS, Javascript, Jquery e Ajax.

Este framework tem uma biblioteca mista com códigos desenvolvidos pela STI e códigos livres de terceiros encontrados na internet, tais como diversos componentes do PEAR, entre outros.

Os códigos de terceiros não são alterados, sendo normalmente criada uma nova classe para utilizar estes componentes, com configurações e adaptações para facilitar o uso nos projetos.

Cada projeto tem uma pasta específica no Apache, para os arquivos dos mesmos, e estes utilizam as bibliotecas comuns que estão localizadas em suas respectivas pastas.

7.3.1 – PEAR

A pasta do PEAR, PHP Extension and Aplication Repository, fica externa ao Apache, em local especificado no arquivo de configuração do PHP, funcionando como extensão deste.

Os componentes do PEAR são normalmente utilizados nas apresentações de listas e formulários relacionados a informações de banco de dados.

Componentes do PEAR utilizados no framework:

. Structures_DataGrid
. Structures_DataGrid_DataSource_Array
. Structures_DataGrid_DataSource_DataObject
. Structures_DataGrid_DataSource_DBQuery
. Structures_DataGrid_DataSource_DBTable
. Structures_DataGrid_DataSource_MDB2
. MDB2_Driver_oci8
. MDB2_Driver_pgsql
. Structures_DataGrid_Renderer_Pager
. Structures_DataGrid_Renderer_HTMLTable
. Structures_DataGrid_Renderer_HTMLSortForm
. html_common2
. html_quickform2

. mail

7.3.2 – _Imagens

Na pasta "_Imagens" são armazenadas todas as imagens fixas comuns utilizadas pelos sistemas, tais como botões especiais e logos. Nota: O framework normalmente utiliza links (ancoras) formatados como botões para realizar a navegação e seleção de opções nos sistemas.

7.3.3 – _lib

Na pasta "_lib" são armazenadas outros componentes de terceiros e a parte central da biblioteca desenvolvida pela STI, com as classes e funções que são utilizadas pelos projetos.

7.3.3.1 – Outras bibliotecas de terceiros

. captcha – Gerador de imagens com código para dificultar acessos automatizados indesejados.
. dhtml-suite calendar – Apresentação de calendários.
. fpdf – Geração de arquivos pdfs.
. mpdf – Geração de arquivos pdfs.
. PHPExcel – Leitura e geração de planilhas.
. TreeMenu – Apresentação de menu em arvore.

7.3.3.2 – Componentes desenvolvidos do framework

a) captcha – Cria um captcha e carrega na sessão. Utiliza componente captcha de terceiro.
b) charset – Arquivo a ser carregado para o HEAD HTML de todos os programas, com linha de configuração do conjunto de caracteres. Atualmente é utilizado o conjunto iso8859-1.

c) func_acesso – Verifica se usuário tem acesso a uma atividade. Se usuário ainda não estiver logado, verifica e registra o login.

d) func_bd, func_bd_pgsql (class BD) – Classe de conexão ao banco de dados, com funções para abrir e fechar conexão, consultar, alterar, incluir, excluir, executar procedimentos e apresentar mensagens de erro, entre outras. Esta classe também armazena, em tabela temporária do PostgreSQL, as informações do usuário logado para que possam ser acessadas em procedimentos do banco de dados (functions e triggers).

e) func_email – Função para envio de e-mails. Utiliza mail do PEAR.

f) func_mensagem – Função para apresentação de mensagens com ou sem opções.

g) func_mensagem_pdf – Função para apresentação de mensagens com ou sem opções, durante geração de pdfs.

h) func_php – Funções gerais, não relacionadas a banco de dados.

i) func_titulo – Função para apresentação de títulos.

j) pdf – Classe para a geração de pdfs. Utiliza componente fpdf de terceiro.

k) proc_altura_tela – Verifica altura e largura da tela no cliente.

l) proc_array_php_js – Códifica array PHP em array Javascript.

m) proc_calendario – Apresentação de calendário. Utiliza dhtmlSuite-calendar.

n) proc_check_session – Verifica sessão do framework.

o) proc_form – Apresentação de formulários envolvendo registros de banco de dados. Utiliza html_quickform2 do PEAR.

p) proc_form_simples – Apresentação de formulários sem tratamento de banco de dados. Utiliza html_quickform2 do PEAR.

q) proc_tabela – Apresentação de listas a partir de banco de dados. Utiliza Structures_DataGrid do PEAR.

r) Arquivo de configuração da sessão (nome omitido por segurança).

s) Style_* – Arquivos css com formatação e cores.

7.3.4 – Componentes desenvolvidos para cada sistema

a) index – Inicializa a sessão do serviço, divide o corpo html em dois frames: Parte superior com o título e parte principal onde ficará toda a lógica do sistema.

b) Arquivo de configuração da conexão do banco de dados (nome omitido por segurança).

c) _titulo – Apresenta a parte superior da tela, com identificação do sistema e do usuário logado.

d) _principal – Inicializa a sessão principal, carrega arquivos de estilo (css) e biblioteca principal, faz conexão com o banco de dados e controla a navegação do sistema verificando as permissões de acesso.

e) _menu – Apresenta o menu no lado direito da parte principal.

f) ini_identificação – Tela para login.

g) ini_principal – Tela com informações iniciais para o usuário logado.

h) Programas do projeto:

. tb_* - Programas do módulo de tabelas.
. ca_* - Programas do módulo de cadastros.
. rl_* - Programas do módulo de relatórios.
. fe_* - Programas do módulo de ferramentas (importação, cópia de dados , etc).
. cf_* - Programas do módulo de configuração.
. ad_* - Programas do módulo administrador (logs).

Uma primeira versão dos programas de tabelas e cadastros (tb_* e ca_*) é gerada automaticamente a partir de aplicativo desenvolvido pela STI, utilizando a própria definição do banco de dados (entidades, atributos, relacionamentos, sequências e índices).

7.4 – Testes

Atualmente quem desenvolve um programa faz os primeiros testes. Após enviar versão para o SVN, o gerente do projeto faz uma verificação do código, validando lógica e padrão, seguido de testes de simulação, executando o código e fornecendo diversas possíveis entradas de dados. Até o momento ainda não foram implantados testes automatizados.

7.5 – Artefatos e fluxo

Artefato de entrada: Acordo de nível de serviço (Item 5).
Artefato de entrada: Dados a serem utilizados pelo sistema (Item 6).
Artefato de saída: Modelagem de banco de dados.
Artefato de saída: Scripts de banco de dados.
Artefato de saída: Scripts de atualização de banco de dados.
Artefato de saída: Fontes do sistema.
Fluxo: Implantação de sistemas (Item 8).

8 - Implantação de sistemas desenvolvidos.

Após a etapa de desenvolvimento, incluindo os testes, uma nova versão do sistema oficial é disponibilizada. O administrador do banco de dados realiza a atualiza o banco através de scripts de atualização e o gerente de projeto atualiza os fontes do projeto no servidor oficial.

Nos casos em que a atualização possa necessitar de algum tempo maior de processamento, o sistema é alterado para apresentar mensagem de manutenção.

Em casos críticos, com muitas alterações no banco de dados, é realizada uma cópia do banco de dados oficial para ambiente de teste para a execução do script de atualização neste antes de atualizar o banco de dados oficial.

9 - Suporte aos usuários dos sistemas.

O suporte aos usuários, com atendimento de solicitações de ajuda na utilização, de melhorias ou de correções é realizado através do sistema de suporte desenvolvido pela STI (www.sistemas.univasf.edu.br/suporte).

A STI, quando necessário, realiza treinamento dos principais usuários do sistema, normalmente o usuário administrador do setor solicitante. O setor solicitante deverá realizar o treinamento dos demais usuários quando necessário.

Atualmente está sendo estudada a possibilidade da STI disponibilizar treinamentos EAD para os sitemas desenvolvidos.

Após o desenvolvimento do sistema, são elaborados e disponibilizados manuais para os usuários que podem ser obtidos na própria página web do serviço, clicando em ícone vermelho de ajuda próximo ao nome do sistema.

10 – Artefatos

10.1 – SVN

Todos os artefatos são versionados e armazenados em repositório através de SVN.

Os repositórios do SVN são criados e gerenciados pelo administrador do SVN, que também é responsável por criar os usuários e permitir o acesso aos repositórios.

Repositórios existentes:

. _Imagens
. _lib
. pear

. docs (para documentação de todos os projetos, separados por pasta)
. scripts (para scripts de banco de todos os projetos, separados por pasta)
. Repositório específico para cada projeto, com os programas.

10.2 – Catálogo de sistemas

Os principais documentos do serviço, da solicitação até o dicionário de dados, serão publicados no catálogo de sistemas. Somente usuários logados no Sistema de Gestão de Serviços (SGS) poderão ter acesso aos documentos do catálogo.

11 – Anexos

11.1 – Configurações

11.1.1 – PHP / Apache (ambiente de desenvolvimento)

a) Instalação do Apache 2.2 em computador de desenvolvimento através de executável binário.

Domain: localdomain

Host: localhost

e-mail: e-mail do desenvolvedor

Instalação Típica

Endereço de Instalação: c:\apache2\

Aterar no httpd.conf:

diretório dos documentos

DocumentRoot "c:/apache2/htdocs"

não permitir listar arquivos dos diretórios

options ...

permitir abrir automaticamente index.php

... index.html index.php

b) Instalação do PHP 5.3 através de arquivo zip para windows

Descompactar na pasta c:\php

Alterar no PHP.INI

; para apresentar mensagens de erro na página

;display_errors = Off

display_errors = On

; para apresentar mensagens de erro de inicialização do PHP

;display_startup_errors = Off

display_startup_errors = On

; diretório dos modulos (extensão)

extension_dir = "c:\php\ext\"

tirar comentário da linha extension=php_gd2.dll no php.ini

```
# tirar comentário da linha extension=php_openssl.dll no php.ini
# tirar comentário da linha extension=php_pgsql.dll no php.ini
```

c) Configuração do PHP no Apache

```
# LoadModule ...
# para PHP 5:
      LoadModule php5_module "c:/php/php5apache2_2.dll"
     AddType application/x-httpd-php .php
     # configure the path to php.ini
     PHPIniDir "c:/php"

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
   DirectoryIndex index.html index.php
</IfModule>
```

11.1.2 – Configuração do PEAR

Instalação do PEAR executando go-pear na pasta do PHP.

A pasta do PEAR poderá ser copiada de outro computador ou através do repositório pear do SVN.

Instalação de componentes (sem copiar a pasta):

```
pear install -o Structures_DataGrid
pear install -o Structures_DataGrid_DataSource_Array
```

```
pear install -o Structures_DataGrid_DataSource_DataObject

pear install -o Structures_DataGrid_DataSource_DBQuery

pear install -o Structures_DataGrid_DataSource_DBTable

pear install -o Structures_DataGrid_DataSource_MDB2

pear install -o MDB2_Driver_oci8

pear install -o MDB2_Driver_pgsql

pear install -o Structures_DataGrid_Renderer_Pager

pear install -o Structures_DataGrid_Renderer_HTMLTable

pear install -o Structures_DataGrid_Renderer_HTMLSortForm

pear install -o html_common2

pear install -o html_quickform2

pear install -o mail
```

(ver observações para instalar versões beta)

11.1.3 – Eclipse (projeto e repositório)

Para configurar um repositório no eclipse, deverá acessar a perspectiva “SVN Repositories” e selecionar opção “new” → “Repository location...”. Informar o diretório do repositório, por exemplo http://ipe-amarelo.univasf.local/svn/pear.

Na perspectiva “PHP Explorer” deverá criar o novo projeto (php project), especificando o diretório local do sistema.

Com o projeto criado, selecionar “Team” → “Share Project” → “SVN” → “Use existing repository location”, selecionar o repositório, por exemplo http://ipe-amarelo.univasf.local/svn/pear. Selecionar opção “Use specified folder name” e informar “trunk/[nome do projeto]”, por exemplo trunk/pear.

Petrolina-PE, 09 de setembro de 2015.

DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO

SECRETARIA DE TECNOLOGIA DA INFORMAÇÃO