

Manual de Procedimentos para Desenvolvimento de *Softwares*

Secretaria Geral de Informática – Sin

Universidade Federal de São Carlos - UFSCar

Versão 2.1

O(s) critério(s) de escolha pelas tecnologias para desenvolvimento definidas neste documento se baseiam em:

- habilidade técnica da atual equipe de desenvolvedores;
- custo total de propriedade para se manter uma aplicação rodando, o que inclui custos como licença de uso e de desenvolvimento, licença de uso da IDE, licença de uso do servidor, entre outros, optando-se preferencialmente por software livre;
- portabilidade; e
- escalabilidade.

1. Ambiente de Desenvolvimento

No contexto da SIn, foram definidos conjuntos de padrões e *frameworks* para o desenvolvimento de *softwares* para *web* e *softwares* para dispositivos móveis.

1.1. *Softwares* para *Web*

Utilizando plataforma Java¹, recomenda-se a utilização da última versão oficial (estável) dos *softwares* a seguir:

- Oracle Java SE JDK² ou OpenJDK³;
- Maven⁴ como ferramenta de automação de compilação;
- HTML5⁵ como linguagem de marcação para web;
- CSS⁶ 3 como linguagem de folhas de estilo para definir a apresentação web;
- JQuery⁷ e JQuery UI⁸ como bibliotecas JavaScript e AJAX;
- Bootstrap⁹ como *framework* para interfaces responsivas;
- *Framework* MVC JavaServer Faces¹⁰;
- *Framework* Facelets¹¹, *Framework* RichFaces¹² e *Framework* Primefaces¹³ para enriquecer e melhorar as interfaces;
- EclipseLink¹⁴ ou Hibernate¹⁵ como *frameworks* ORM (mapeamento objeto/relacional);

¹ <http://www.java.com>

² <http://www.oracle.com/technetwork/java/javase/overview/index.html>

³ <http://openjdk.java.net>

⁴ <http://maven.apache.org>

⁵ <http://www.w3.org/TR/html5>

⁶ <http://www.w3.org/Style/CSS>

⁷ <http://jquery.com>

⁸ <http://jqueryui.com>

⁹ <http://getbootstrap.com>

¹⁰ <http://javaserverfaces.java.net>

¹¹ <http://facelets.java.net>

¹² <http://richfaces.jboss.org>

¹³ <http://primefaces.org>

¹⁴ <http://www.eclipse.org/eclipselink>

¹⁵ <http://hibernate.org>

- EJB - *Enterprise JavaBeans*¹⁶ como componente arquitetural para aplicações Java EE;
- NetBeans IDE¹⁷ como ferramenta para desenvolvimento para projetos Java Web e Java EE e GGTS¹⁸ para projetos Grails;
- JBoss AS¹⁹ como servidor de aplicação Java EE e Tomcat²⁰ como servidor *web*;
- JUnit²¹ como *framework* para testes de unidade;
- SeleniumHQ²² como *framework* para testes de integração;
- PostgreSQL²³ como servidor de banco de dados

1.2. Softwares para Dispositivos Móveis

Utilizando o *software* Android²⁴, recomenda-se a utilização da última versão oficial (estável) dos *softwares* a seguir:

- Android SDK²⁵
- Android Studio²⁶ ou Eclipse IDE²⁷ como ferramentas para desenvolvimento

2. Controle de versões

Deve-se utilizar Subversion²⁸ ou Git²⁹ com a finalidade de gerenciar diferentes versões de artefatos gerados durante o processo de desenvolvimento de um software. A UFSCar, sob gerenciamento da SIn, disponibiliza repositórios para ambos os sistemas de controle de versão.

Para que os desenvolvedores possam atualizar (*commit*) a documentação ou os códigos dos sistemas no repositório é necessário solicitar o cadastro enviando um email para o Analista de TI da SIn, gerente responsável pelo projeto.

No repositório existem três diretórios: **branches**, **tags** e **trunk**.

No diretório trunk é armazenada a versão mais recente do projeto em desenvolvimento. Este diretório possui outros dois repositórios: implementação e documentação.

¹⁶ <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

¹⁷ <https://netbeans.org>

¹⁸ <https://grails.org/products/ggts>

¹⁹ <http://jbossas.jboss.org>

²⁰ <http://tomcat.apache.org>

²¹ <http://junit.org>

²² <http://seleniumhq.org>

²³ <http://www.postgresql.org>

²⁴ <http://www.android.com>

²⁵ <http://developer.android.com/sdk/index.html>

²⁶ <http://developer.android.com/sdk/installing/studio.html>

²⁷ <http://www.eclipse.org>

²⁸ <http://subversion.apache.org>

²⁹ <http://www.git-scm.com>

O diretório tags contém versões específicas do projeto que recebem um nome ou rótulo mais amigável, por exemplo, release1.0, release2.1, etc. A versão disponível nesse diretório representa uma cópia do projeto em um determinado momento e não pode ser alterado.

No diretório branches são armazenadas cópias do projeto com funcionalidades que, no momento, não devem ser incluídas na versão atual do desenvolvimento.

OBS: Cada submissão (*commit*) realizada no repositório sempre deve incluir um comentário objetivo e explicativo das alterações realizadas.

2.1. Ferramentas de apoio para controle de versão

Ambos os *softwares* para o controle de versão possuem um conjunto de comandos que podem ser executados via terminal (para Linux) ou *prompt* (para Windows) com a finalidade de gerenciar as revisões, ramificações e históricos. Contudo, *softwares* clientes como RabbitVCS³⁰ (para Linux) e TortoiseSVN³¹ (para Windows) possibilitam a manipulação de repositórios e cópias de trabalho por interface gráfica.

3. Organização do Código-fonte

A estrutura adotada para a organização do código para os sistemas é:

<base>.<sistema>.<camada>.<diretório>

As partes <base>, <camada>, <ferramenta>, <componente> e <diretório> devem ser formadas pelas seguintes normas:

1. Letras ASCII minúsculas, ou seja, sem espaços, sem letras maiúsculas ou caracteres especiais;
2. <base>: sugestão é br.<nome empresa>, para projetos desenvolvidos por terceiros ou br.ufscar.<sin>, para projetos desenvolvidos pela SIN;
3. <sistema>: nome do sistema;
4. <camada>.<diretório>: nesta parte são separados os componentes Java.

Exemplos:

- a. br.ufscar.sin.tramite.apresentacao – localização dos managed beans
- b. br.ufscar.sin.tramite.apresentacao.conversores – localização das classes que realizam conversões de tipo de objetos
- c. br.ufscar.sin.tramite.apresentacao.util – classes que implementam funcionalidades que são utilizadas por diversos managed beans
- d. br.ufscar.sin.tramite.negocio.fachada – classes de negócio do sistema.
- e. br.ufscar.sin.tramite.negocio.entidade – classes das entidades do mapeamento objeto-relacional, seguindo as notações JPA.
- f. br.ufscar.sin.tramite.persistence.dao – classes responsáveis pela persistência dos objetos

³⁰ <http://rabbitvcs.org>

³¹ <http://tortoisesvn.net>

4. Gerência de Projetos

O Scrum foi adotado pela SIn como metodologia ágil para apoio ao processo de desenvolvimento de software.

O gerenciamento de projetos e requisitos da UFSCar é realizado através de uma ferramenta livre e de código aberto denominada Redmine³². Esta ferramenta web permite o controle de tarefas durante todo o ciclo de desenvolvimento de um projeto. Além disto, o Redmine integra-se com os repositórios Subversion e Git da UFSCar permitindo o acompanhamento de revisões e versões dos artefatos, permite o planejamento de *releases* e inclui calendário e gráfico de Gantt para ajudar representação na visual dos projetos e seus prazos.

A integração dos códigos e documentação é realizada no momento da atualização do repositório, adicionando um comentário, onde se menciona o identificador da tarefa relacionada precedido pelo caractere “#”. Pode-se fazer o caminho inverso também, mencionando na definição da tarefa o número da revisão que deu origem a aquela tarefa precedido pelo caractere “r”.

O Redmine também permite a criação de “Fóruns” para discutir determinados assuntos entres os *stakeholders* do projeto, bem com um local para armazenamento artefatos, documentos e arquivos gerados durante o processo de desenvolvimento e documentação de um projeto. Pode-se também criar paginas de Wiki que auxiliam aos usuários finais com instruções para instalação, configuração e uso do software produzido.

³² <http://www.redmine.org>

Anexo I

O anexo apresenta o fluxo de processo para sub-contratação de serviços de desenvolvimento de software.

