# Descrição de Metodologias Ágeis para Desenvolvimento de Software

Mauricio de Assis

Raphael Corrêa Vicentino

Valdenir Trevisan Junior.

¹Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas - Instituto Federal do Paraná - Câmpus Paranavaí.

# 1 Introdução

Desenvolver software não é uma tarefa complexa como a maior parte dos indivíduos acredita ser, o desenvolvimento correto e organizado depende de dedicação da equipe de desenvolvimento como um todo. Para que o projeto chegue ao fim a contento, há técnicas e formas de desenvolvimento ágeis que serão definidas ao longo do trabalho.

Este trabalho foi realizado utilizando a Pesquisa Exploratória que envolveu a consulta a livros, artigos e sites que apresentam definições de métodos e ferramentas de trabalho para desenvolvimento de sistemas.

A organização do trabalho está assim disposta na Seção 2 descreve os métodos e práticas de trabalho como o Scrum e Programação Extrema. As ferramentas de desenvolvimento são apresentadas na Seção 3 que são o eclipse e o git; por fim, a plataforma de desenvolvimento, a linguagem de programação que é o java EE e os frameworks Prime Faces e Hibernate são detalhados.

# 2 Métodos de Trabalho

Com o crescente avanço das tecnologias é necessário cada vez mais que uma equipe de desenvolvimento possua certa organização em seus projetos, sendo assim, se torna necessário adquirir metodologias ágeis de trabalho para que assim haja organização e velocidade nos trabalhos, no capitulo serão tratados algumas ferramentas de trabalho como o Scrum e a Programação Extrema, onde cada uma possui suas particularidades de trabalho.

## 2.1 Scrum

O Scrum é caracterizado como uma metodologia ágil que é realizada por equipes que fazem parte de um determinado projeto, estas são divididas e entregam as tarefas em várias iterações que são chamadas também de *sprints* (corrida de velocidade em inglês).

O Scrum é composto por uma equipe multifuncional, os indivíduos são intitulados de Scrum Master, um *Product* *Owner*, e a equipe, na qual todos possuem o conhecimentos específicos para desenvolvimento de sua tarefa.

Segundo Macedo e *Sbrocco* (2012) o Scrum não é utilizado somente em desenvolvimento de software, mas também pode ser aplicado no gerenciamento de qualquer trabalho, pois é interativo incremental, que significa que o produto é construído em partes, ou seja, de forma iterada. As *sprints* devem ser muito bem planejadas, pois há a necessidade de estabelecer um tempo hábil para a execução de determinada tarefa a ser desenvolvida; esse planejamento é feito em reuniões que duram em torno de oito horas; o projeto é iniciado com o *Product* *Owner* que recebe as informações dos *stakeholders* e elabora uma lista de requisitos e cria um *Backlog* de Produto, que seria um “resumo” do trabalho realizado naquele tempo.

Para Pham e Pham (2011) mesmo que o Scrum e outras metodologias ágeis sejam difíceis de aplicar, se utilizadas de maneira correta geram resultados satisfatórios.

Conclui-se então, que por mais trabalhosa que seja a utilização da metodologia para organização de todo o processo de desenvolvimento tanto de software como de outros projetos o Scrum é uma ferramenta essencial para o sucesso do projeto como um todo.

## 2.2 Programação Extrema

Conhecida como Extreme Programming ( XP ou Programação extrema) é considerada uma metodologia ágil, possui uma característica básica que é a de formação de equipes pequenas, com feedback constante, abordagem incremental, que seria a entrega de etapas de tempos em tempos e motiva a comunicação interpessoal. Sbrocco e Macedo (2012, p.143) afirma que “A ideia básica é enfatizar o desenvolvimento rápido do projeto, visando garantir a satisfação do cliente, além de favorecer o cumprimento das estimativas”.A Programação Extrema é composta por práticas, as quais podem ser descritas a seguir:

**Histórias**: É a prática na qual se obtém com o cliente todos os requisitos necessários do sistema, nessa fase é levantado os requisitos do sistema junto com o cliente, verificando suas necessidades, a partir disto é feito todo o planejamento do projeto.

**Ciclos Semanais de estação**: Trata-se das iterações, é a analise de tudo o que já foi projetado, e são reavaliadas as histórias que já foram executadas no desenvolvimento do sistema, a partir de então é feita uma previsão das próximos requisitos que serão executados com o tempo necessário e a prioridade, esses ciclos são curtos para que mudanças possam ser observadas em considerações.

**Envolvimento real com o cliente**: O cliente deve estar sempre disponível para analisar tudo oque já foi feito e tirar eventuais dúvidas da equipe de desenvolvimento, pois requer alta disponibilidade do mesmo ou de alguém que conheça do sistema que será desenvolvido, para que assim se possa auxiliar nas etapas necessárias a equipe.

**Área de trabalho informativa**: É responsável por manter a equipe e o cliente informado sobre as etapas do projeto que está sendo executado, o tempo necessário para realização de cada etapa, o que já foi desenvolvido, entre outros, ou seja, é um relatório de tudo o que foi concluido e quais serão as próximas etapas.

**Implantação incremental**: Nessa etapa cabe ao cliente decidir a necessidade dele ou não do sistema ser entregue, sendo assim se obter pela incrementação a cada ciclo é gerado uma nova funcionalidade do sistema e com isso o cliente tem essa funcionalidade disponível no sistema ou seja o sistema é disponibilizado a cada ciclo uma nova funcionalidade, possibilitando o trabalho com a ferramenta.

**Programação em pares**: Na programação extrema é fundamental a programação em pares, trata-se de dois usuários desenvolvendo o mesmo código, um é o piloto, que fica encarregado para digitar o código e o segundo é o copiloto que fica sentado ao lado vendo o código gerado procurando eventuais erros e tentando obter melhoras no código gerado.

**Integração contínua**: Todo o código gerado pela equipe deve ser compartilhado entre si o mais rápido possível, para que assim todos possam ter acesso e verificar o que está sendo executado e feito.

Para o sucesso efetivo do resultado do projeto os papéis devem ser bem definidos, como Gerente de Projeto, o Coach, o Desenvolvedor, o Analista de teste, e o Redator técnico.

# 3. Ferramentas

Ferramentas foram criadas para facilitar e auxiliar o trabalho de um desenvolvedor, para que o processo de desenvolvimento de um software ocorra de forma rápida e segura e seja melhor desenvolvido é necessário contar com tecnologias que possam facilitar o trabalho, uma delas é o Git e o eclipse, nos tópicos a seguir será detalhado a forma como essas ferramentas atuam em um projeto de software e o quão benéficas são, entre outras características.

## 3.1 Eclipse

O Eclipse é uma IDE (*Integrated Development Environment)* para desenvolvimento de programas. O Eclipse é amplamente utilizado pois tem uma documentação robusta, é estável e comumente usado em empresas do segmento de desenvolvimento. Além de ser altamente personalizável pois possui diversos *plugins* que aumentam a produtividade do programador (Eclipse Foundation, 2016).

Inicialmente o Eclipse foi desenvolvido pela IBM que é uma empresa que sempre acompanhou o segmento de computadores e *softwares* desde o início. Atualmente trata-se de uma IDE *open source (*um *software* gratuito para uso) pois a IBM disponibilizou o código fonte do Eclipse para que a comunidade possa fazer constantes melhorias. Gonçalves reforça a ideia afirmando que “Eclipse é uma das IDEs Java mais utilizado no mundo”.

Por se tratar de um ambiente que é *open source*  o eclipse se torna uma IDE que se ajusta de maneira simples as necessidades do programador, pois possui um sistema de perspectiva. Por exemplo, se um determinado individuo está desenvolvendo um sistema web, existe uma perspectiva feita para isso com ferramentas específicas a este propósito, caso esteja sincronizando o código que está em uma máquina local com o repositório, seria equivalente, mesmo que exista várias formas de ver o projeto pode-se ainda personalizar, até que fique da forma desejada. Possui integração fácil com o Git ( Controlador de versões) através de um *plugin* já existente, o módulo específico para *debug* (descobrir falhas ou possíveis problemas no sistema) que é simples e eficaz, auxiliando assim o programador no momento em que ele não entende o problema que ocorrido na função desenvolvida.

### 

## 3.2 Git

O Git é um sistema de controle de versão, ele registra as mudanças realizadas em um ou mais conjuntos de arquivos durante um determinado período, possibilitando a visualização ou recuperação de versões anteriores (Chacon, 2009).

Foi desenvolvido por Linus Torvalds (o criador do Linux) e mais alguns desenvolvedores da comunidade Linux, depois do relacionamento comercial da comunidade com a empresa que desenvolvia comercialmente o software BitKeeper (software de controle de versão) se desfazer em 2005.

De forma diferente de alguns outros deste, o Git não registra apenas as mudanças efetuadas no arquivo, ele trabalha com o sistema de *Snapshot* (captura de algo em um determinado instante), guardando uma cópia do arquivo inteiro, de como ele estava naquele momento, e gravando somente os arquivos que sofreram alguma alteração, não guardando arquivos repetidos, tornando-se eficiente.

Funciona de forma local e distribuída: Possui uma área de trabalho e um repositório local, na área de trabalho os desenvolvedores fazem as mudanças que são salvas no repositório, onde fica o histórico de mudanças realizadas. A forma distribuída é utilizada para trabalho em equipe, após os desenvolvedores executarem e salvarem suas mudanças, os arquivos são compartilhados em um repositório na internet, onde toda a equipe envolvida no projeto pode ter acesso ao mesmo.

## 

## 4. Plataforma, Linguagem e Frameworks

Para que um projeto seja desenvolvido é necessário contar com alguns itens que trabalham em conjunto, que é plataforma linguagem e framework, ambos os 3 itens trabalham em conjunto com a linguagem, onde inicia-se o projeto, plataforma é o mesmo que a escolha se o software irá trabalhar em ambiente web ou desktop, e framework vem para poder facilitar a parte do desenvolvimento, agilizando o trabalho tanto na parte de designer quanto no desenvolvimento do projeto.

### 4.1 Linguagem de programação Java

Com o intuito de desenvolver um aplicativo portátil para funcionar em qualquer tipo de equipamento, em 1990 uma equipe da Sun Microsystems ao tentar adaptar a linguagem C/C++, criou uma nova linguagem de programação, que ganhou o nome de “OAK”, porém esta inovação não fez sucesso de início. Somente em 1995 com popularização da internet, por ser uma linguagem independente e portátil podendo ser utilizada em qualquer plataforma, a “OAK” ganhou sucesso. No mesmo ano a Sun Microsystems fez algumas alterações, renomeou a linguagem chamando-a de “Java” e a relançou, sendo muito aceita pelos desenvolvedores. O nome Java é um sinônimo de café, algo que foi muito consumido durante o desenvolvimento da linguagem, daí o nome.

Segundo Gudwin (1997 p.9) é uma linguagem parcialmente compilada e parcialmente interpretada. Através do compilador o código fonte é transformado em arquivo-objetos, os *bytecodes*, que são interpretados pela *Java Virtual Machine* (Maquina Virtual Java - JVM). O java funciona como um emulador, que independente da plataforma onde a JVM foi instalada, consegue executar os mesmos *bytecodes* sem a necessidade de adaptação.

O Java é divido em vários tipos segundo o segmento de desenvolvimento:

* Java SE - desenvolvimento de aplicações para desktop;
* Java EE - desenvolvimento de aplicações para WEB;
* Java ME - desenvolvimento de aplicações para dispositivos moveis;
* Java FX - desenvolvimento de aplicações para desktop, browser e dispositivos móveis;
* Java DB - desenvolvimento de banco de dados;
* Java TV - desenvolvimento de aplicações para Tv´s.

### 

### 4.2 Java EE

Aplicações *web* tem como objetivo conseguir acessar o seu sistema de qualquer lugar, dando comodidade e praticidade, porém nem sempre é uma tarefa fácil fazer o desenvolvimento de tais aplicações, pois envolve muita regras para torna-lá funcional e segura. É nesta ideia que vem o Java EE (*Enterprise Edition*) ele vem com varias receitas prontas, ou seja, você tem códigos já feitos que auxiliam na execução, pois desta forma você economiza muito tempo que seria gasto com configurações do projeto. E usando esta versão do java você estará também seguindo um padrão mundial que é usado em sistemas web de grande porte passando assim maior confiabilidade do seu *software*. Podemos destacar algumas funcionalidades que a versão EE do java nos ajuda, como:

* Servlets: é um classe java que transforma uma página estática em dinâmica, ou seja a página conseguira receber uma informação e depois ira processar esse dado recebido para dar um resultado desejado.
* JSP (Java Server Pages): Uma versão aprimorada dos Servlets onde você consegue fazer o processamento das informações sem a necessidade de um Servlets especifico para para fazer, ou seja, na mesma página em que você terá o código que exibira as informações do seu sistema fara o processamento das informações de acordo com o que o usuário informar, assim como o Servlets faz mais sem a necessidade de ter um propriamente dito.
* JPA (Java Persistence API): API é equivalente ao código já feito por outros usuarios que disponibilizou para que pudéssemos usar sem a necessidade de refazer pois trata-se de algo pronto. agora a parte da *Persistence*  é o nome dado quando você deve persistir, salvar um dado em um banco de dados. Em resumo JPA é uma API que auxilia o sistema persistir dados no banco de dados da aplicação.

Essas são as principais vantagens na escolha do Java EE para realizar uma aplicação web, esses padrões foram desenvolvidos pela Sun empresa que criou o a linguagem de programação java.

### 

### 4.3 Framework de mapeamento objeto relacional Hibernate

Em nosso dia a dia é cada vez mais necessário agilidade em projetos, e sendo assim o Hibernate é utilizado para poder aperfeiçoar e facilitar o trabalho dos profissionais no desenvolvimento. Tem como principal objetivo reduzir e aperfeiçoar o tempo de trabalho, é uma ferramenta que faz a conversão da linguagem Orientada a objetos para linguagem relacional que é a forma que os bancos de dados interpretam, ele trata e gerencia todo o código SQL e gera a tempo de execução, e é gerado assim um SQL para um banco de dados específico, ele trata toda a parte de *Create*, *Read*, *Update* e *Delete*, que é toda a parte de criação e gerencia de um banco de dados, reduzindo ao máximo a necessidade de escrever *Querys* Sql, o ponto positivo é a velocidade, pois facilita o trabalho dos desenvolvedores, sem necessidade de trabalhar com o SQL, o framework gera toda a parte de tabelas e banco de dados de forma automática sem que se tenha necessidade de criar manualmente. O ponto negativo é que deixa a desejar no que diz respeito a aprendizagem, pelo fato dos usuários não tratarem de se preocupar com a parte SQL, que é algo importante e consequentemente o desenvolvedor passa a ter foco só na parte de desenvolvimento.

O *framework* realiza o mapeamento de uma classe através de *annotations*, temos vários tipos, e serão representadas por “@” seguida da anotação, ele faz a comunicação dos códigos com o banco de dados através de um arquivo chamado *persistence.xml* que nele é dado algumas configurações como nome do banco de dados, tipo, entre outras características, o Hibernate é tratado como uma ferramenta de conexão entre a linguagem de programação e o banco de dados.

### 4.4 Java Server Faces (JSF)

O Java Server Faces (JSF) é uma ferramenta de rápida resposta para o desenvolvedor, tenta diminuir as dificuldades de desenvolvimento WEB Java; essa tecnologia utiliza componentes prontos, onde o criador pode somente clicar em opções pré-definidas e arrastar e já são exibidos em HTML; este recurso segue o padrão arquitetura MVC (*Model-View-Controller*) no qual faz o papel de *Controller*.

### 4.5 Primefaces

Com do decorrer dos anos e o avanço das tecnologias surgem novos meios para facilitar a vida do usuário e do desenvolvedor, uma delas é o Primefaces, é uma biblioteca que trabalha junto com o Java Server Face (JSF), que tem por principal objetivo definir um padrão de interfaces (Estilos) para as aplicações, ou seja, a ferramenta possui uma ampla quantidade de componentes, desde botões, campos de texto em geral, agregando também o *ajax*, que é um *plug-in* integrado, que facilita o desenvolvimento, pois não há necessidade do programador configurar os componentes para o *ajax*, o próprio Primefaces já efetua a configuração deste componente.

É composto de uma tag padrão, que é a letra **p**, já no JSF é pela **h**,

Exemplo:

**JSF** <**h**:body>

**Primefaces** <**p**:body>

A grande vantagem na utilização da plataforma é que possibilita uma maior opção de componentes estilizados e possui um padrão de componentes dos mesmos.

# 

**REFERÊNCIAS**

**Book**. Disponível em: < https://git-scm.com/book/pt-br/v1//> Acesso em: 20 de Março de 2016.

CHACON, Scott. **Pro Git**. Apress 2009. Disponível em: <https://git-scm.com/book/pt-br/v1/>. Acesso em: 18 mar 2016.

**Conceitos Básicos de Controle de Versão de Software — Centralizado e Distribuído.** Disponível em: <http://www.pronus.eng.br/artigos\_tutoriais/gerencia\_configuracao/conc.php?pagNum=0 /> Acesso em: 20 de Março de 2016.

**Conhecendo o Eclipse - Uma apresentação detalhada da IDE.** Disponível em: <http://www.devmedia.com.br/conhecendo-o-eclipse-uma-apresentacao-detalhada-da-ide/25589> Acesso em : 21 de Março de 2016.

DIAS, André Felipe. **Conceitos Básicos de Controle de Versão de Software - Centralizado e Distribuído**. Pronus Engenharia de Software, 2011. Disponivel em: <http://www.pronus.eng.br/artigos\_tutoriais/gerencia\_configuracao/conceitos\_basicos\_controle\_versao\_centralizado\_e\_distribuido.php?pagNum=0>. Acesso em: 18 mar 2016.

**Estudo de Mapeamento Objeto Relacional Com Framework Hibernate**.Disponível.em:<http://pt.slideshare.net/fernandoabcampos/estudo-de-mapeamento-objetorelacional-com-framework-hibernate/> . Acesso em: 15 de Março de 2016.

**Eclipse no Brasil: 10 anos de história.** Disponível em: <http://www.devmedia.com.br/eclipse-no-brasil-10-anos-de-historia/22989> Acesso em : 21 de Março de 2016.

GUDWIN, Ricardo R. **Linguagens de Programação**. Campinas: UNICAMP, 1997. Disponível em: <ftp://ftp.dca.fee.unicamp.br/pub/docs/ea877/lingpro.pdf>. Acesso em: 26 mar 2016.

**Introdução ao JSF e Primefaces.** Disponível em: <http://docente.ifrn.edu.br/fellipealeixo/disciplinas/tads-2012/desenvolvimento-de-sistemas-corporativos/book/book-jsf-3.pdf/>. Acesso em: 15 de Março de 2016.

**Introdução ao PrimeFaces** Disponível em:<http://www.devmedia.com.br/introducao-ao-primefaces/33139/>. Acesso em: 15 de Março de 2016.

**Java EE 7 com JSF, PrimeFaces e CDI**. Disponível em: <http://docente.ifrn.edu.br/fellipealeixo/disciplinas/tads-2012/desenvolvimento-de-sistemas-corporativos/book/book-jsf-3.pdf>. Acesso em: 15 de Março de 2016.

OLHAR DIGITAL. **Linguagem Java: um pouco de história**. 2013. Disponível em: <http://olhardigital.uol.com.br/noticia/linguagem-java-um-pouco-de-historia/35728>. Acesso em: 26 mar 2016.

ORACLE. **Java Developer Center**. Disponível em: <http://www.oracle.com/technetwork/pt/java/index.html>. Acesso em: 26 mar 2016.

PHAM, Andrew; PHAM, Phuong-Van. **Scrum em ação: gerenciamento e desenvolvimento ágil de projetos de software**.São Paulo: Novatec 2011.

SBROCCO, Jose Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. **Metodologias Ágeis: engenharia de software sob medida**. São Paulo:Érica 2012.

**Uma introdução prática ao JPA com Hibernate** . Disponível em: <https://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/>. Acesso em: 15 de Março de 2016.

**Veja o que é ORM e os frameworks disponíveis para .NET.** Disponível em: < http://www.princiweb.com.br/blog/programacao/aspnet/veja-o-que-e-orm-e-os-frameworks-disponiveis-para-net.html>. Acesso em: 15 de Março de 2016.