**Descrição do Ambiente ALM**

**ALM**: Application Lifecycle Management (ALM), é todo o processo que guia a vida útil de uma aplicação desde a sua concepção, passando pela construção, operação e evolução. A ALM tem como objetivos: possibilitar maior controle ao projeto, previsibilidade no processo de desenvolvimento, garantia do acompanhamento das tarefas do projeto, disponibilizando um controle de versões mais eficiente, dentre outras [1].

**Linguagem:** Para a programação, será utilizada a linguagem Java, por ser uma linguagem de programação orientada a objetos estática e fortemente tipada. O Java, é também uma plataforma de desenvolvimento enriquecida de vantagens assim, será possível desenvolver aplicações para desktop, celular, web, etc. Arquivos podem ser compilados em Java, gerando um bytecode que é interpretado pelo JMV. Assim, será é permitido executa-lo em outros sistemas operacionais, desde que o mesmo possua uma JMV [2].

**IDE:** A IDE a ser usada no desenvolvimento do trabalho, será o NetBeans IDE 8.0, pois segundo [3] oferece suporte abrangente e de primeira classe para as tecnologias e melhorias de especificação Java mais recentes, antes de outros IDEs, além de funcionalidade avançadas. É o primeiro IDE gratuito a oferecer suporte a JDK 8, JDK 7, Java EE 7, inclusive a seus aprimoramentos de HTML5 relacionados, e a JavaFX2. O NetBeans IDE oferece suporte para as linguagens C / C ++ e desenvolvedores PHP, fornecendo editores e ferramentas abrangentes para os seus quadros e tecnologias relacionadas. Sua característica marcante é a instalação em todos os sistemas operacionais que suportam Java.

**Controle de Versão**: A ferramenta para controle de versão será o Git. Segundo [4], o Git é um controle de versão distribuída e é utilizado principalmente para gerenciar versões de software, desenvolvidos por um ou mais desenvolvedores, podendo ser implementado novas funcionalidades ficando registrado tudo em um histórico, o qual é permitido retroceder sempre que necessário, além dos integrantes do projeto, participar enviando correções, atualizações, etc. Alterações não comprometem o projeto principal, pois o dono do projeto sempre optará por incluir ou não as alterações realizadas. Utilizando-se o Git, é evidente algumas vantagens marcantes, tais como: consistência, velocidade, espaço e simplicidade.

**SGDB**: O Sistema Gerenciador de Banco de Dados (SGBD) a ser implementado será o MySQL, por ser conhecido pela facilidade de uso e capacidade de rodar em vários sistemas operacionais. Conforme [5], o banco possui vantagens de ser: gratuito, código fonte aberto, leve, rápido, seguro e prático.

**Ferramenta para Issue/Bug Tracking**: O GitHub é o maior anfitrião de código do planeta, com mais de **21,3 milhões** de repositórios. Grande ou pequeno, cada repositório vem com as mesmas ferramentas. A ferramenta issue tracker é flexível e por isso permite que você fique em cima dos erros e se concentre nos recursos [6].

**Criação de Diagramas UML**: Para a criação de diagramas UML, o programa utilizado será o Dia. O Dia é um programa baseado em gtk+ para criação do diagrama liberado sob a licença do GPL. Ele é inspirado no programa do Windows comercial "Visio", embora seja voltada para diagramas informais para uso casual. O Dia pode ser utilizado para desenhar tipos diferentes de diagramas, possuindo diversos objetos especiais para ajudar a desenhar diagramas de entidade relacionamento, diagramas UML, fluxogramas, diagramas de rede, e muitos outros; além da possibilidade de adicionar suporte para novos formatos de gravação de arquivos em XML, usando um subconjunto de SVG para desenhar a forma [7].

**Arquitetura de Software**:A arquitetura a ser utilizada para o desenvolvimento do software será o padrão dearquitetura Model View Controller (MVC). Pois, segundo [8] é dividida em 3 camadas distintas: Modelo (Model), Visão (View) e Controlador (Controller) e visa separar dados ou lógica de negócios (Model) da interface do usuário (View) e do fluxo da aplicação (Controller), permitindo que uma mesma lógica de negócios possa ser acessada e visualizada através de várias interfaces.

**Referências**

[1] **Introdução ao Application Lifecycle Management (ALM)**. Disponível em: <https://msdn.microsoft.com/pt-br/library/ee156630.aspx >. Acesso em: 04/04/2015.

[2] DE SANTANA, Otávio Gonçalves. **Por que Java?**. Disponível em: <<http://www.devmedia.com.br/por-que-java/20384>>. Acesso: 04/04/2015.

[3] **NetBeans IDE**. Disponível em: < https://netbeans.org/features/index.html>. Acesso em: 04/04/2015.

[4] SILVESTRE, Henrique. **O que é** **Git?**. Disponível em: <<http://pt.slideshare.net/HenriqueSilvestre1/git-e-github-conceitos-bsicos>>. Acesso em: 05/04/2015.

[5] SANTOS, Adriano. **MySQL: Quem é você?**. Disponível em: <<http://www.devmedia.com.br/mysql-quem-e-voce/1752>>. Acesso em: 05/04/2015.

[6] **GitHub – Features**. Disponível em: <<https://github.com/features>>. Acesso em: 05/04/2015.

[7] WILLIAM, Jon Mc Cann. **Dia**. Disponível em: <<https://wiki.gnome.org/action/show/Apps/Dia?action=show&redirect=Dia>>. Acesso em: 05/04/2015.

[8] DE ALMEIDA, Rodrigo Rebouças. **Model – View - Controller (MVC)**. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>>. Acesso em: 05/04/2015.