**Configuração das máquinas utilizadas no desenvolvimento do projeto**

***Máquina 01 - Dell Inspirion 14 R:***

Proprietário: Wesley Gomes da silva

***Configurações:***

* Memória de 6 Gigabytes.
* Processador Core i 5 intel.
* HD de 1 TB.
* Placa de vídeo de 1 GB.

***Máquina 02 - Dell Inspirion 14 R:***

Proprietário: Heitor Vinicius Mariano

***Configurações:***

* Memória de 6 Gigabytes.
* Processador Core i 5 intel.
* HD de 1 TB.
* Placa de vídeo de 1 GB.

***Máquina 03 – Asus A431:***

Proprietário: Hirohito Pereira

***Configurações:***

* Memória de 8 Gigabytes.
* Processador Core i 5 intel.
* HD de 750 GB.

**tecnologias que serão utilizadas em nosso sistema**

1. **JAVA**

Logo abaixo alguns fatores decisivos na escolha da linguagem JAVA para o desenvolvimento do projeto.

* ***Java é gratuito***

Java depende de uma JVM, que é gratuita. O programador depende de um compilador e do JDK, que também é gratuito. Programadores costumam utilizar IDEs, como Eclipse ou Netbeans. Para Java, as duas maiores IDEs do mercado (Eclipse e NetBeans) também são gratuitas. Há vários frameworks e ferramentas para a implementação de sistemas em Java, e os mesmos também costumam ser gratuitos. A documentação é gratuita. Servidores são gratuitos. Drivers de Banco de Dados são gratuitos.

* ***Java é interoperável***

Java é interoperável, ou seja, funcionam em várias arquiteturas distintas, sistemas operacionais distintos e trabalha com paradigmas de programação distintos como desktop e web. Na web Java pode ser utilizado no servidor (JSP e Servlets) ou no cliente (Applets). Apesar de haver diferenças para a implementação (J2EE, J2SE e J2ME), a linguagem é a mesma e o conceito é o mesmo. Indo além, Java permite também uma facilidade para internacionalização com a utilização nativa de arquivos Properties e definição automática de valores financeiros, numéricos, de data e texto. Além disto, a utilização de Unicode pode garantir a visualização de textos em vários alfabetos distintos.

* ***Java possui frameworks, padrões de projeto e Componentes.***

Padrões de projetos, frameworks e componentes são exemplos destas propostas. Há vários exemplos de como implementar padrões de projetos em Java e vários padrões de projeto que foram propostos pensando nesta linguagem. Frameworks Java facilitam a implementação de sistemas complexos e componentes fazem com que o programador Java possa compartilhar seu conhecimento com toda a comunidade Java. Alguns exemplos:

* + Hibernate
  + Jakarta Velocity
  + JSF
  + GWT
  + Jakarta POI
  + Log4J
  + JSTL
* ***Java integra com Bancos de dados***

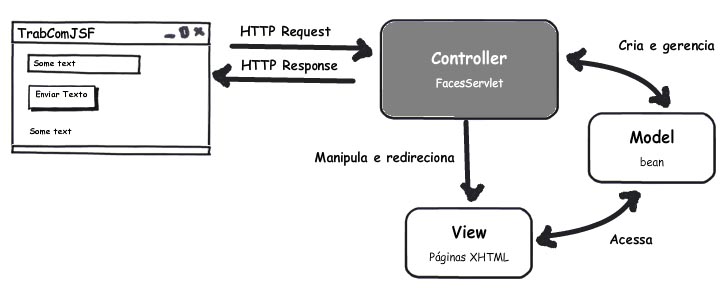
A interface JDBC faz com que a comunicação com bancos de dados em Java funcionem de maneira homogênea. Além disto, há atualmente implementações de drivers de banco de dados para Java para a maioria dos bancos comerciais. Inclusive ODBC. Inclusive Access. Inclusive bancos de dados feitos em Java e que não dependem de servidores. A API JDBC permite a migração de um sistema de um banco de dados para outro a partir da alteração de sua conexão. Isto facilita a passagem de uma aplicação do ambiente de desenvolvimento e testes para o ambiente de produção.

Como o projeto se trata de um sistema totalmente web, para o desenvolvimento utilizaremos um frameworks web baseados em componentes JavaServer Faces - JSF.

JSF é uma tecnologia que nos permite criar aplicações Java para Web utilizando componentes visuais pré-prontos, de forma que o desenvolvedor não se preocupe com Javascript e HTML. Basta adicionarmos os componentes (calendários, tabelas, formulários) e eles serão renderizados e exibidos em formato html.

Outra característica marcante na arquitetura do JSF é a separação que fazemos entre as camadas de apresentação e de aplicação. Pensando no modelo Model-view-controller - MVC, o JSF possui uma camada de visualização bem separada do conjunto de classes de modelo.

Abaixo na figura 1, o modelo da arquitetura JSF;

Figura 1 – Representação da arquitetura JSF.

A imagem acima representa a arquitetura do JSF, onde a solicitação do usuário envia um HTTP Request para o Controller que e responsável por receber as requisições vinda das views, gerenciar os models, representando a lógica do negócio.

Para atender a demanda dos desenvolvedores por componentes mais sofisticados, há várias extensões do JSF que seguem o mesmo ciclo e modelo da especificação. Utlizaremos o PrimeFaces, para a definição da interface do projeto “Busca de Serviços” utilizaremos Oracle Mojarra com PrimeFaces.