



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DANIEL PAULO DE ASSIS

**INTEGRAÇÃO DE UM APLICATIVO WEB EM JAVA COM O
GOOGLE MAPS API**

2011

Assis, SP



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DANIEL PAULO DE ASSIS

INTEGRAÇÃO DE UM APLICATIVO WEB EM JAVA COM O GOOGLE MAPS API

Trabalho de Conclusão do Programa de Iniciação Científica (PIC) do Instituto Municipal de Ensino Superior de Assis– IMESA.

Aluno: Daniel Paulo de Assis

Orientadora: Profa. Dra. Marisa Atsuko Nitto

Linha de Pesquisa: Informática

2011

Assis, SP

FICHA CATALOGRÁFICA

PAULO DE ASSIS, Daniel

Integração de um aplicativo web em Java com o google maps api / Daniel Paulo de Assis. Fundação Educacional do Município de Assis –Fema : Assis, 2011
50p.

Orientadora: Profa. Dra. Marisa Atsuko Nitto
Projeto de Iniciação Científica (PIC) – Ciência da Computação - Instituto Municipal de Ensino Superior de Assis

1. Java 2. Web. 3. Google Maps

CDD: 001.6
Biblioteca da FEMA

RESUMO

Nos dias atuais a informatização de conteúdos, e a integração entre sistemas através da internet é uma realidade, o consumo de serviços fornecidos através dessa grande rede provê inúmeras facilidades a pequenas, médias, grandes empresas e também aos usuários que podem usufruir de todas essas facilidades.

Uma consulta de um saldo bancário ou pagamento de uma conta online, uma consulta de notas sem que o aluno tenha a necessidade de se locomover até a instituição de ensino, localizar um ponto no mapa tendo vários pontos como referência, promover sua empresa e atualizar conteúdos de maneira dinâmica e intuitiva, são serviços que podem ser oferecidos a um custo baixo e com resultados muito satisfatórios.

Este projeto utiliza os recursos providos pela Linguagem de programação Java para especificar, implementar um aplicativo *web* que disponibiliza serviços on-line como consulta de notas e vencimentos de uma instituição de ensino e também promove uma Integração com a API Google Maps, possibilitando a inserção de “pontos” no mapa com intuito de promover a fácil localização da Instituição bem como dos pontos de referência ao redor.

LISTA DE FIGURAS

Figura 1 - Localização da FEMA no Google Maps	9
Figura 2- Arquitetura JSF baseada no modelo MVC. (SILVEIRA; COSENTINO, 2009).	14
Figura 3 - Visão Geral do Sistema	18
Figura 4 - Modelagem do Problema	19
Figura 5 – WSB (Work BreakDown Structure)	20
Figura 6 –Sequenciamento de Atividades	21
Figura 7 - Mapa do Site	22
Figura 8 - Diagrama Entidade Relacionamento.....	24
Figura 9 - Diagrama de Casos de Uso	25
Figura 10 - Pacote Bean	27
Figura 11 - Pacote Model	28
Figura 12 - Pacote Control	29
Figura 13 - Pacote Dao	30
Figura 14 - Diagrama de Atividades	31
Figura 15 - Diagrama de Sequência Adicionar Marcador.....	32
Figura 16 - Diagrama de Sequência Visualizar Mapa	33
Figura 17 - Diagrama de Sequência Consultar Marcador	34
Figura 18 - Diagrama de Sequência Consultar Notas Alunos	35
Figura 19 - Diagrama de Sequência Consultar Vencimentos.....	36
Figura 20 - Interface Principal do Aplicativo	38
Figura 21 - Menu Área do Aluno	39
Figura 22 - Área do Aluno - Consulta de Notas.....	40
Figura 23 - Área do Aluno - Consulta de Vencimentos	41
Figura 24 - Menu Localize-se	42
Figura 25 - Inserção de um novo Marcador.....	43
Figura 26 - Visualização de Marcador com infoWindow.....	44
Figura 27 - Localização Prefeitura de Assis modo Satélite	45
Figura 28 - Localização Prefeitura de Assis modo mapa	45
Figura 29 - Localize-se Consulta Marcadores	46

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivos	10
1.2	Justificativas	10
1.3	Motivação	11
1.4	Estrutura do Trabalho	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Tecnologias Java	12
2.2	PostgreSQL	15
2.3	Google Maps API	16
3	DESENVOLVIMENTO DO APLICATIVO	17
3.1	Descrição do Problema	17
3.2	Modelagem do Problema	19
3.3	Especificação	20
3.3.1	Levantamento de Requisitos	22
3.3.2	Diagrama Entidade Relacionamento (DER)	23
3.3.3	Diagramas dos Casos de Uso	25
3.3.4	Diagrama de Classes	26
3.3.5	Diagrama de Atividades	31
3.3.6	Diagramas de Sequência	31
3.3.7	Adicionar Marcador	32
3.3.8	Visualizar Mapa	33
3.3.9	Consultar Marcador	34
3.3.10	Consultar Notas Alunos	35
3.3.11	Consultar Vencimentos	36
3.4	IMPLEMENTAÇÃO DO APLICATIVO	36
3.4.1	Criação da Base de Dados	37
3.4.2	Criação de Interface	37
3.4.3	Integração com a Api Google Maps	37

3.4.4	Funcionamento do Aplicativo.....	38
3.4.5	Área Do Aluno	39
3.4.6	Localize-se.....	42
3.4.7	Markers.....	46
4	CONCLUSÃO.....	47
	REFERÊNCIAS.....	48

1 INTRODUÇÃO

Grandes empresas como Google vêm disponibilizando Interfaces de Programação de Aplicativos (API - *Application Programming Interface*) de comunicação com suas ferramentas *web*. Desta forma, proporcionam aos desenvolvedores, o uso da grande capacidade de processamento e a enorme fonte de informação proveniente dos seus serviços. São exemplos as atuais API do *Google Maps* e a API do *Orkut* para desenvolvimento de aplicações. As APIs têm se tornado cada vez mais difundidas na *internet*, especialmente por *softwares* livres que disponibilizam aos seus usuários a possibilidades para que estes criem *plugins*, ou seja, ferramentas que complementam a funcionalidade inicial do software.

A API do *Google Maps* é uma interface de desenvolvimento para aplicações baseadas no próprio *Google Maps*, fornecendo ao desenvolvedor uma alternativa tanto de construção de aplicações baseadas em localização quanto à utilização de mapas e criação de camadas adicionais – *Overlays* – que permitem dar mais expressividade gráfica para as aplicações *web*. Essa API ajuda a integrar mapas e geocodificação aos sites, possibilitando que as aplicações com conteúdo georeferenciado sejam facilmente apresentadas em qualquer navegador (GLORIA; CAMPELO; RIBEIRO, 2007). A API *Google Maps* foi feita inteiramente em *JavaScript* e utiliza formas revolucionárias de trabalhar com AJAX (*Asynchronous Javascript And XML*). AJAX faz uso metodológico de tecnologias como *JavaScript* e XML (*Extensible Markup Language*), providas por navegadores, para tornar páginas *web* mais interativas com o usuário, utilizando de solicitações assíncronas de informações. O *Google Maps* é um serviço gratuito fornecido pela empresa *Google*, usado para realizar pesquisas, visualização de mapas e imagens de satélite da Terra e possibilita à produção de um mapa personalizado completo, sendo possível marcar locais, adicionar vídeos, fotos e compartilhar todo este conteúdo na rede mundial de computadores (GOOGLE MAPS, 2010 e LIU, 2010). A figura 1 mostra a localização da FEMA (Fundação Educacional do Município de Assis) no *Google Maps*.

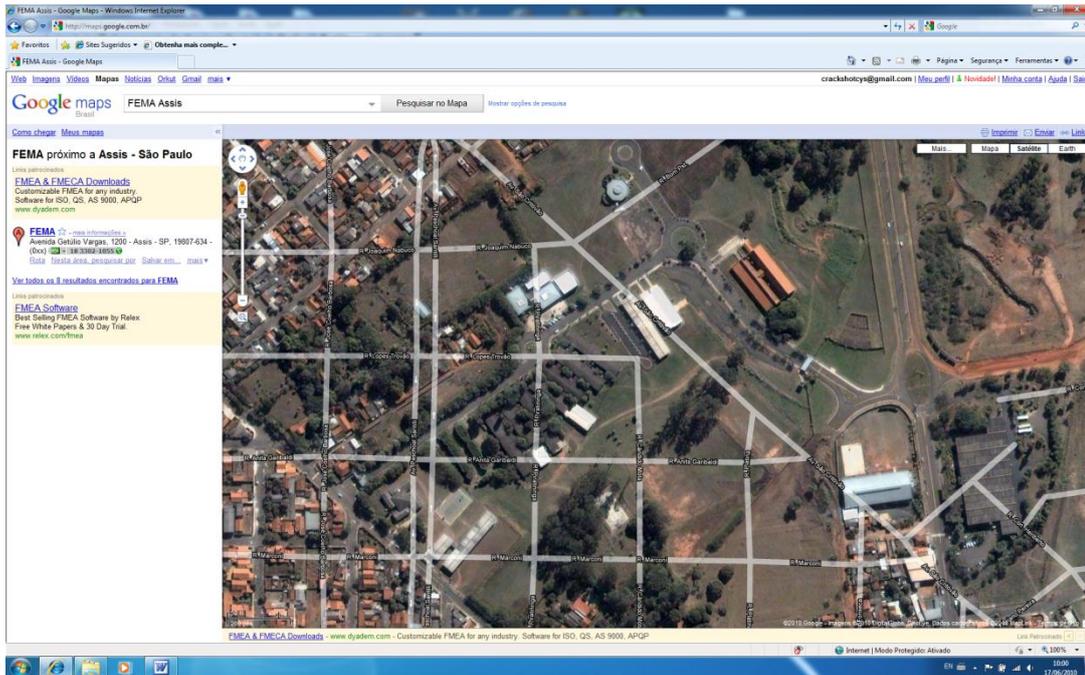


Figura 1 - Localização da FEMA no Google Maps

Como a API é feita em *JavaScript*, pode se criar alguns mapas estáticos para ilustrar seu funcionamento, mas, para trabalhos profissionais, nada melhor que estar usando um banco de dados e uma linguagem de programação por trás.

Em se tratando de aplicativo *web*, a tecnologia Java se sobressai em relação às outras existentes por ser *open-source*. A linguagem de programação Java apresenta uma série de características que a torna muito atrativa para o desenvolvimento de aplicações. Essas características abrangem a orientação a objetos, independência da plataforma, robustez, segurança, confiabilidade e conectividade com a *web* (LEMA; PERKINS, 1996 e DEITEL, 2003). Isto proporciona o desenvolvimento de projetos com maior qualidade, portabilidade e com menor tempo de desenvolvimento.

Este estudo de caso envolve uma aplicação *web* focado na área educacional que utiliza os serviços baseados em localização (*location-based services*). Eles têm se tornado uma das funcionalidades mais interessantes no contexto de serviços disponíveis para aplicação *web*. A integração deste aplicativo com o *Google Maps* permite a visualização de uma lista de escolas e suas respectivas localizações no

mapa por meio do componente *Overlay*. Além da localização das escolas serão apresentados os serviços disponíveis na vizinhança da localidade (como farmácias, restaurantes e outros). Neste estudo de caso será incorporado também o gerenciamento de uma das escolas, onde o usuário pode consultar o horário de aula, notas, vencimentos do boleto e outras informações necessárias.

1.1 Objetivos

Este projeto tem como objetivo desenvolver um aplicativo utilizando as tecnologias Java para desenvolvimento *Web* e integrar com o Google Maps API. As tecnologias envolvidas para o desenvolvimento do aplicativo são *Serves*, JSP (*Java Server Pages*), JSF, servidor de aplicação Apache Tomcat, XML, AJAX e a *Google Maps API* para manipulação de dados espaciais.

O aplicativo em questão é capaz de disponibilizar a localização das escolas cadastradas em um banco de dados, e localizar também outros serviços disponíveis na proximidade da escola escolhida (farmácias, restaurantes, etc.). Disponibilizar outras informações aos usuários como, horário de aula, notas e outras.

1.2 Justificativas

O desenvolvimento deste projeto de pesquisa se justifica, tendo em vista, que nos dias atuais, o usuário final precisa realizar determinada ação que pode ser facilitada se utilizadas aplicações que estejam enquadradas no contexto de serviços baseados em localização, como guias para rotas, descoberta de serviços disponíveis em determinada localidade (como farmácias, restaurantes, pizzarias e outros). Estes tipos de serviços estão sendo timidamente explorados, mas mostra uma demanda grande no mercado de negócios.

Outro fator importante é que a API do *Google Maps* está sendo cada vez mais utilizada e aprimorada pela Google, fazendo com que o conteúdo dos *sites* seja mais

dinâmico e de fácil acesso com a utilização de mapas. Este aprimoramento pode ser observado pelo lançamento recente do *Google Street View* (Google Visto da Rua) que é um recurso do *Google Maps* e do *Google Earth* que disponibiliza vistas panorâmicas de 360° na horizontal e 290° na vertical e permite que os usuários ou utilizadores vejam partes de algumas regiões do mundo ao nível do chão ou solo.

1.3 Motivação

Uma das principais motivações para o desenvolvimento deste trabalho, sem dúvida alguma, é o crescente mercado de aplicações *web*. O avanço tecnológico tem propiciado desenvolvimento de aplicativos cada vez mais sofisticados, e isso requer conhecimento cada vez maior das tecnologias envolvidas nesta área. Algumas das tecnologias que serão abordadas visam outro mercado futuramente, que é a de aplicativos para dispositivos móveis.

1.4 Estrutura do Trabalho

A estrutura do trabalho foi dividida em capítulos:

Capítulo 1: Introdução

Capítulo 2: Fundamentação Teórica

Capítulo 3: Desenvolvimento do Projeto

Capítulo 4: Conclusão

Referências Bibliográficas

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos que fundamentam esse projeto de pesquisa. Primeiramente, serão feitas considerações sobre as tecnologias utilizadas para o desenvolvimento de aplicações *web*, e sobre o banco de dados PostgreSQL utilizado e por fim, serão feitos alguns comentários sobre a ferramenta do Google Maps de visualização de imagens da terra.

2.1 Tecnologias Java

Aplicações *Web* estão cada vez mais presentes e seu desenvolvimento representa boa parte da produção de organizações desenvolvedoras de *software* bem como de mídia em geral. O rápido crescimento das aplicações *Web*, tanto em seu escopo quanto na extensão de seu uso, tem afetado todos os aspectos de nossas vidas (GINIGE; MURUGESAN, 2000). Por representar uma evolução do *software* convencional, algumas preocupações adicionais motivaram as pesquisas relacionadas à engenharia de aplicações *Web* (*Web Engineering*), mantendo o objetivo de aplicar princípios de engenharia para desenvolver aplicações *Web* de qualidade (PRESSMAN, 2002). Com isso, a tecnologia Java ganhou espaço, pois foi projetada para se mover através de redes de dispositivos heterogêneos, como a *Internet*. Agora as aplicações podem ser executadas dentro dos *Browsers* nos *Applets* Java e tudo seria disponibilizado pela *Internet* instantaneamente. Foi o estático HTML dos *Browsers* que promoveu a rápida disseminação da dinâmica tecnologia Java.

O berço da linguagem Java é um projeto da *Sun Microsystems* de 1991, chamado de Projeto Green, que tinha como mentores Patrick Naughton, Mike Sheridan, e James Gosling. Era um projeto sobre *software* para produtos eletrônicos de consumo. O objetivo era que os eletrônicos tivessem um *software* embarcado que possibilitasse o seu controle via rede de computadores e uma maior interatividade

com o usuário (DEITEL; DEITEL, 2005, SILVEIRA; COSENTINO, 2009 e DRJT, 2010).

O modelo de programação utilizando na tecnologia Java recebe a denominação na literatura de *Java Web*, pois é voltado para aplicações *Web*. *Servlets* e *Java Server Pages* (JSP) são os programas Java que executam em um servidor *Web*, e respondem a requisições feitas por clientes através de navegadores. *Servlets* é uma API Java utilizada para o desenvolvimento de aplicações *web* e utilizada para processar requisições de clientes recebidas através do modelo solicitação resposta (KURNIAWAN, 2002). As duas tecnologias podem ser utilizadas de maneira integrada, ou separadamente. Em aplicações cujo foco é apresentação, JSP se configura como a melhor opção. Em aplicações orientadas a tarefas, a melhor opção é utilizar *servlets* (HALL; BROWN, 2003). A tecnologia JSP é parte da família de tecnologias Java. Páginas JSP são compiladas em *servlets* e podem chamar componentes *JavaBeans* (*beans*) ou componentes *Enterprise JavaBeans* (*EJB*) para a realização do processamento no servidor (BONFIN JUNIOR, 2002 e GIGAFLOPS, 2010).

O *Enterprise JavaBeans* (*EJB*) são objetos distribuídos que apresentam uma estrutura bem definida, isto é, implementam interfaces específicas e que executam no lado do servidor. A vantagem na utilização dos *EJBs* é que a aplicação irá se beneficiar de serviços como transações, segurança, tolerância a falhas, *clustering*, distribuição, controle de sessão entre outros (ARMSTRONG et al., 2005).

Outra tecnologia baseada em Java é o *Java Server Faces* (*JSF*) que incorpora características de um *framework* MVC (*Model-view-controller*) para *web* e de um modelo de interfaces gráficas baseado em eventos. Por ser baseado no padrão de projeto MVC, uma de suas melhores vantagens é a clara separação entre a visualização e regras de negócio (modelo). O modelo representa os objetos de negócio e executa uma lógica de negócio ao receber os dados vindos da camada de visualização. Finalmente, a visualização é composta por *component trees* (hierarquia de componentes *User Interface* (UI)), tornando possível unir um componente ao outro para formar interfaces mais complexas. A figura 2 mostra a arquitetura JSF baseada no modelo MVC.

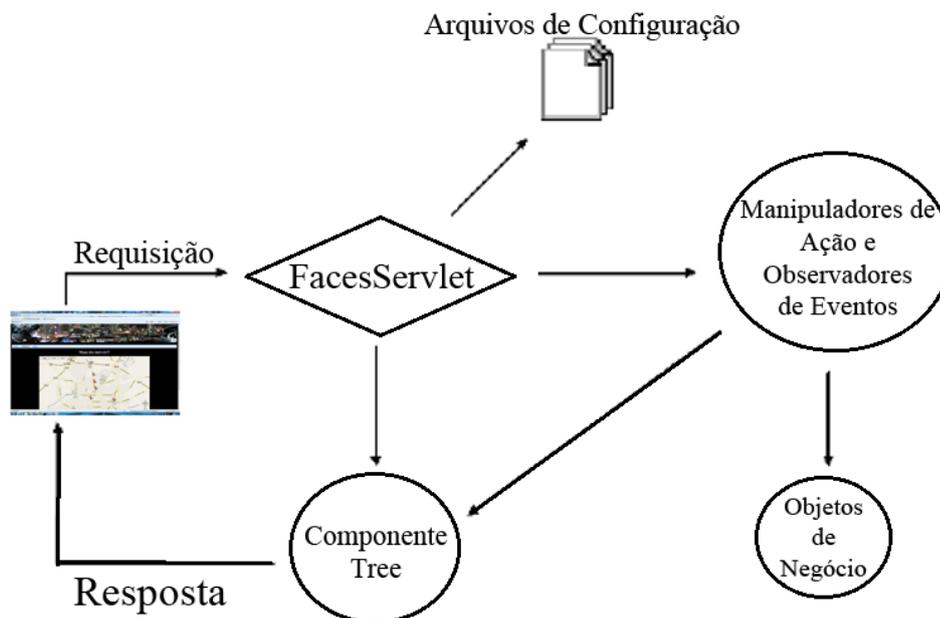


Figura 2- Arquitetura JSF baseada no modelo MVC. (SILVEIRA; COSENTINO, 2009).

A plataforma Java fornece suporte em diferentes camadas da aplicação, da interface com o usuário à comunicação com o banco de dados, do tratamento de solicitações e respostas aos usuários aos mecanismos de funcionamento do componente que distribui a aplicação pela *web*.

A necessidade que as empresas tinham em armazenar grandes quantidades de informação sobre temas diversos surgiu o XML (*Extensible Markup Language*) que provém de uma linguagem que a IBM inventou por volta dos anos 70 chamada *General Markup Language* (GML). O XML é uma tecnologia que tem ao seu redor outras tecnologias que a complementam e a faz muito maior e com possibilidades muito mais amplas. O XML é importante no mundo globalizado atual, pois pode compartilhar as informações de uma maneira segura, fácil e confiável.

Ao redor do XML está um mundo de tecnologias, facilidade no trabalho com os dados e um avanço na hora de tratar a informação. Enfim, o XML não é só uma linguagem e uma sintaxe, e sim várias linguagens com várias sintaxes (FURTADO JUNIOR, 2010, W3SCHOOLS, 2010 e ALVAREZ, 2010).

Outro fator importante no desenvolvimento de aplicações *web* é o servidor de aplicações *web*, e um dos utilizados é o *Tomcat*, desenvolvido pela Fundação *Apache*. Sua principal característica técnica é estar centrada na linguagem de programação Java, mais especificamente nas tecnologias de *Servlets* e de *Java Server Pages* (JSP). Ele está escrito em Java e, por isso, necessita que a versão *Java 2 Standard Edition* (J2SE) esteja instalada no mesmo computador onde ele será executado. No entanto, não basta ter a versão *runtime* de Java instalada, pois o *Tomcat* necessita compilar (e não apenas executar) programas escritos em Java. O servidor *Tomcat* tem a habilidade de converter automaticamente qualquer página JSP em um *servlet* equivalente. Em outras palavras, o *Tomcat* é capaz de criar código fonte Java a partir de um documento HTML (APACHE TOMCAT, 2010).

2.2 PostgreSQL

Um bom Sistema Gerenciador de Banco de Dados (SGBD) é essencial para o funcionamento de um sistema ou aplicativo, pois ele é quem garante que os dados sejam armazenados de maneira correta em relação ao domínio da aplicação. Uma questão importante que deve ser levada em consideração é a segurança, pois ela impede a violação de consistência dos dados, acesso indevido (usuários e aplicações), além de prover controle de concorrência (conflito de acesso a dados simultâneos). Isso tudo com total transparência ao usuário final.

Para esse projeto foi escolhido o PostgreSQL um banco de dados de nível corporativo com a vantagem de ser gratuito. O PostgreSQL dá total suporte a programação Java, pois trabalha com o *driver* JDBC (*Java Database Connectivity*) que permite que programas Java possam se conectar a um banco de dados PostgreSQL utilizando todos os recursos disponíveis pelo SGBD.

O PostgreSQL é um poderoso sistema gerenciador de banco de dados objeto-relacional de código aberto. Tem mais de 15 anos de desenvolvimento ativo e uma arquitetura que comprovadamente ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões. Roda em todos os grandes

sistemas operacionais, incluindo GNU/Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e MS Windows. (Sobre o PostgreSQL, PostgreSQL 2010).

2.3 Google Maps API

O *Google Maps* é um serviço gratuito fornecido pela empresa *Google*, usado para realizar pesquisas, visualização de mapas e imagens de satélite da Terra. O *Google Maps* possibilita a produção de um mapa personalizado completo, sendo possível marcar locais, adicionar vídeos, fotos e compartilhar todo este conteúdo na rede mundial de computadores (GOOGLE MAPS, 2010 e LIU, 2010). Quando o *Google Maps* foi lançado, em fevereiro de 2005, ainda em sua versão beta, tornou-se rapidamente uma referência em serviços de mapas na *Internet*. Com uma interface rica e intuitiva, a aplicação permitia acesso a uma enorme base de dados contendo inúmeras imagens de satélite, mapas de cidades, bairros, ruas e avenidas dos Estados Unidos. Com o tempo, novas localidades foram sendo adicionadas no sistema até que em meados de maio de 2007, a *Google* finalmente disponibilizou consultas de endereços no Brasil, sendo que, em outubro de 2007, uma versão estendida para o público brasileiro foi oferecida, com a possibilidade de se localizar restaurantes, hotéis, traçar rotas, dentre outras utilidades (AZEVEDO, 2010).

A API do *Google Maps* é uma interface de desenvolvimento para aplicações baseadas no próprio *Google Maps*, permitindo criar aplicativos inovadores de mapeamento para a plataforma da *Web*. Essa API ajuda a integrar mapas e geocodificação aos sites, possibilitando que as aplicações com conteúdo georeferenciado sejam facilmente apresentadas em qualquer navegador (GOOGLE DEVELOPER GUIDE, 2010).

Alguns dos recursos disponíveis são: zoom, pan, visualizar imagens de satélites de diversos locais do planeta, incluir marcadores de locais, realizar *overlay* de arquivos KML (*Keyhole Markup Language*), pesquisarem por locais, traçar rotas, buscar endereços, inserir linhas, polígonos e outros. Uma observação importante é que todo este procedimento se dá dentro da "filosofia" AJAX, o que significa dizer que o

usuário não tem que se preocupar com os detalhes (SOUZA NETO, 2009). O AJAX surgiu em fevereiro de 2005, como sendo um conceito de navegação e atualização de páginas *Web*. Antes do surgimento do AJAX, exigia-se que para cada solicitação em uma página *Web*, a página inteira fosse atualizada no navegador, ou seja, independentemente da atualização ser apenas de uma pequena parte da página, toda a página era recebida pelo servidor, redesenhada e retornada para o navegador, o que gerava mais tráfego de informações do que era necessário e conseqüentemente, tornando o processo mais lento. Sendo assim, com a utilização do AJAX pode-se trafegar apenas os dados que realmente foram atualizados em uma página *Web*, ganhando-se em qualidade e rapidez de processamento. AJAX também pode ser utilizado em conjunto com qualquer tecnologia Java para *Web* (LIMEIRA, 2006).

3 DESENVOLVIMENTO DO APLICATIVO

Esta seção dará ênfase a descrição e modelagem do problema, para facilitar o entendimento o processo foi dividido em etapas. Para obtenção de um bom desempenho e cumprimento de metas foram utilizadas metodologias da Engenharia de Software para especificação, implementação e documentação.

3.1 Descrição do Problema

Neste projeto foi desenvolvido um aplicativo *web* que se integra com a ferramenta *Google Maps*. A maior dificuldade para esse tipo de problema é determinar uma solução de fácil implementação e rapidez na resposta. O aplicativo *web* visa disponibilizar um serviço diferenciado para a área educacional e neste projeto os *markers* são dinâmicos, onde são inseridas informações e imagens da escola. *Markers* são figuras que marcam um determinado ponto no mapa. Foi incluída também no aplicativo a parte de gerenciamento de uma das escolas, para que no futuro todas as outras escolas sejam integradas no sistema. O aplicativo é um

sistema *web* com um servidor de serviços *web*, que será desenvolvido em JSF (*Java Server Faces*) com comunicação XML com o Servidor da Google através da *Google Maps API* para manipulação de dados espaciais e com o banco de dados PostgreSQL.

A figura 3 mostra a visão geral do Sistema.

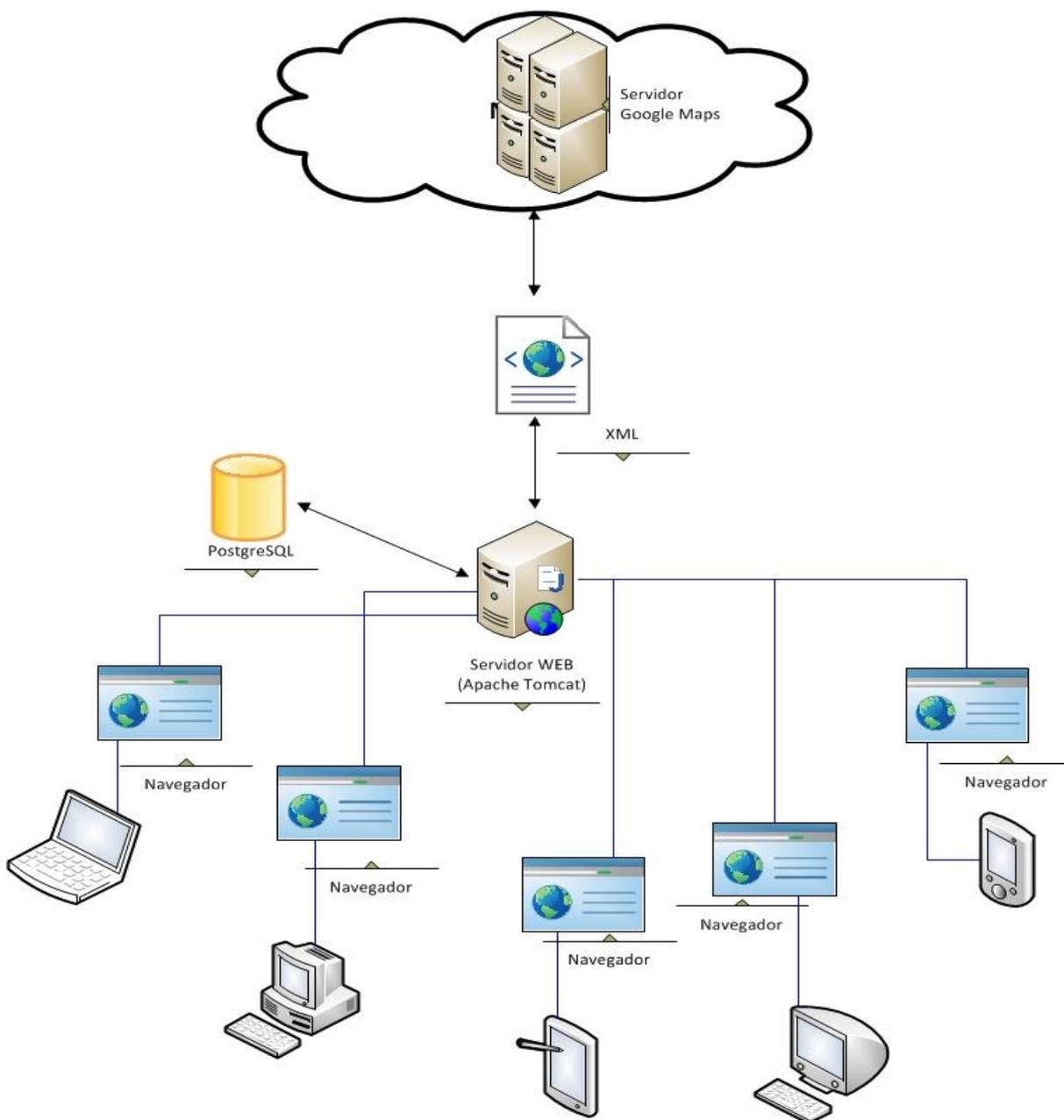


Figura 3 - Visão Geral do Sistema

3.2 Modelagem do Problema

A modelagem do problema é de suma importância, pois ela dá uma visão geral do que se pretende desenvolver. A figura 4 mostra esta modelagem e como o aplicativo *web* se integra com a API *Google Maps* utilizando o arquivo no formato XML para transporte dos dados.

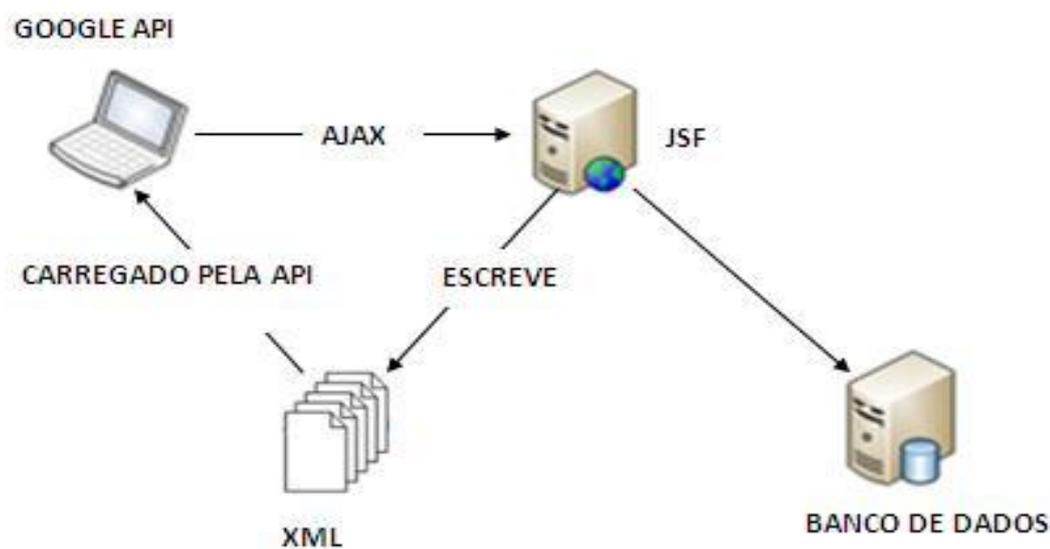


Figura 4 - Modelagem do Problema

O desenvolvimento do projeto foi dividido em módulos para facilitar a implementação.

O desenvolvimento do projeto foi dividido em módulos para facilitar a implementação.

Módulo 1: Criação da Base de Dados

Módulo 2: Criação da Interface

Módulo 3: Integração com a Api Google Maps

3.3 Especificação

As etapas utilizadas para o desenvolvimento do projeto baseiam-se em metodologias da Engenharia de Software, onde foi construída uma WBS (*Work Breakdown Structure*) para definir como essas etapas são inseridas no projeto. Também foi construído um diagrama que define o seqüenciamento das atividades estipulando o tempo médio para cada atividade a ser desenvolvida. O seqüenciamento de atividades é utilizado para que as etapas de desenvolvimentos não sejam “atropeladas”, e também para se obter uma média de tempo que será gasto no decorrer do projeto.

Para esse projeto não foi feito um planejamento de custos, pois a princípio o sistema não visa obter lucros, o desenvolvimento em questão busca, nesta primeira fase, adquirir conhecimentos de todas as ferramentas utilizadas no desenvolvimento. Todos os *softwares* e *hardwares* utilizados foram cedidos pela instituição que financiou este projeto.

A figura 5 mostra a WBS com as etapas do projeto.

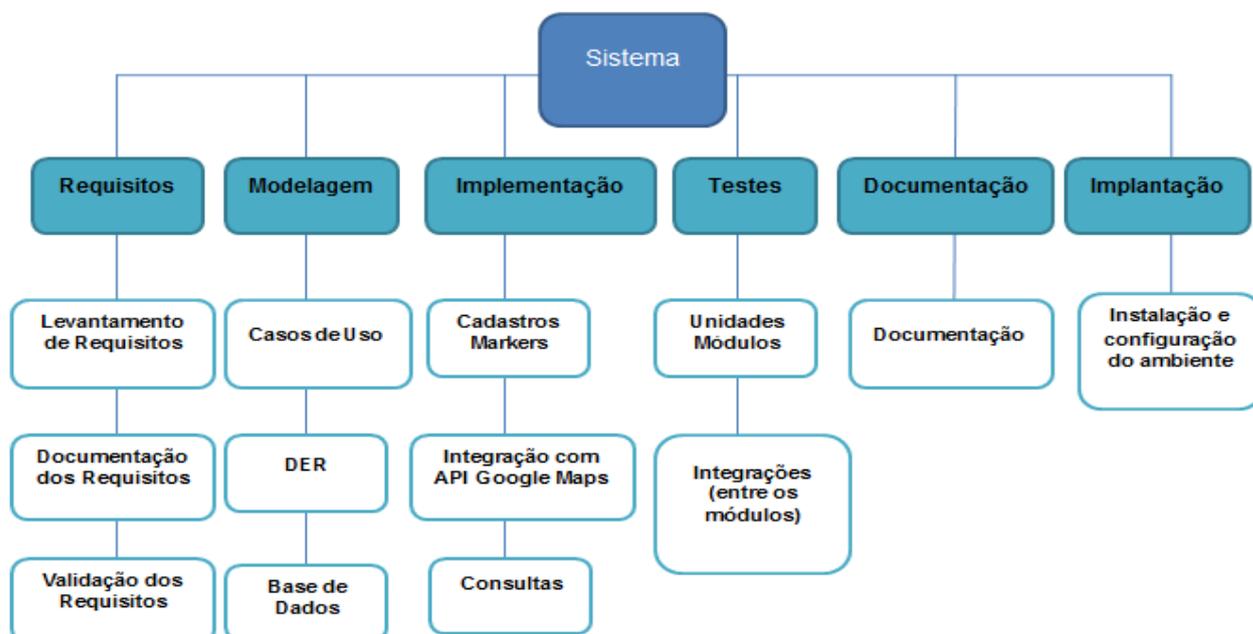


Figura 5 – WBS (Work BreakDown Structure)

A figura 6 mostra o sequenciamento de atividades que foi utilizado para este projeto.

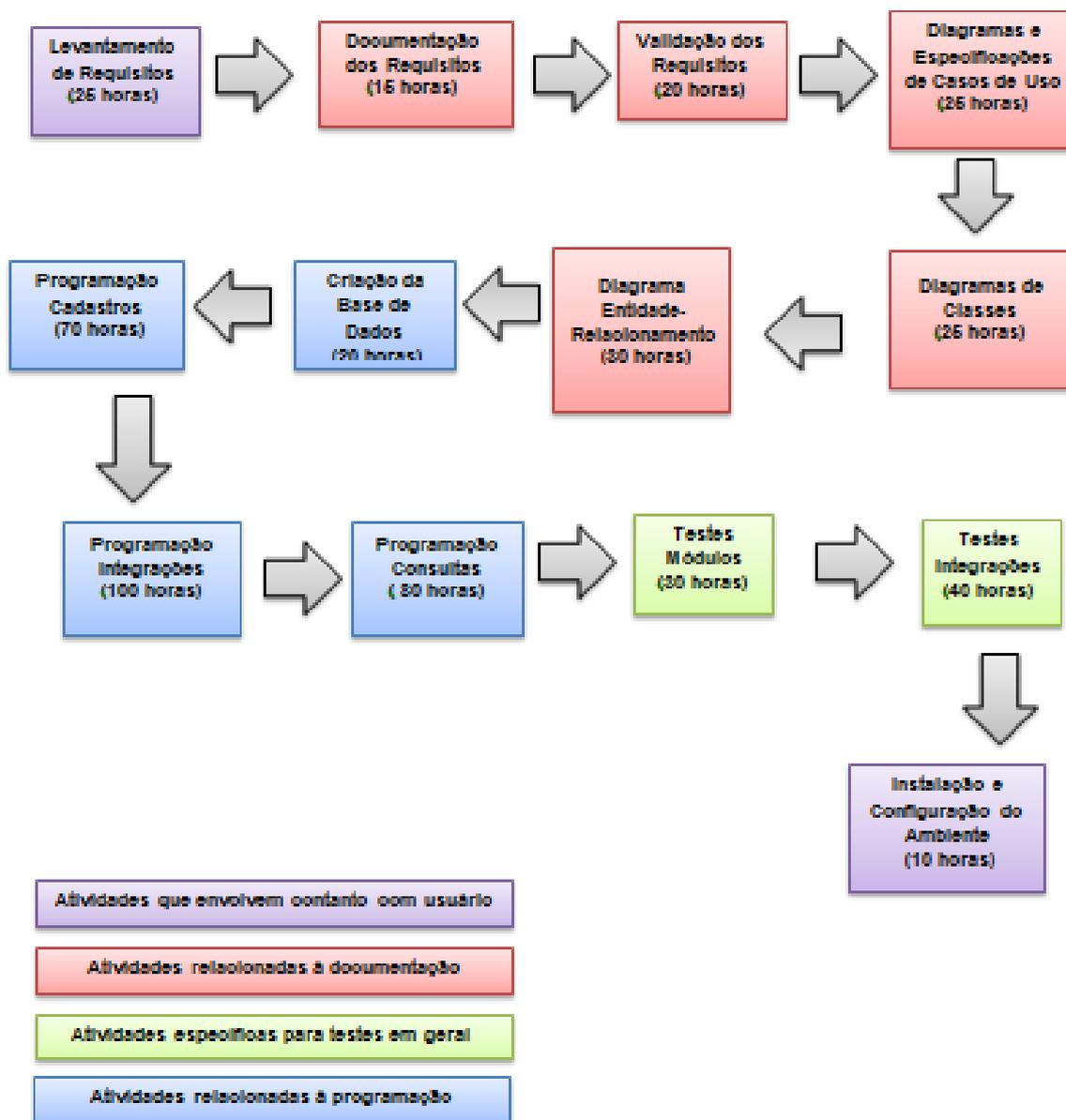


Figura 6 –Sequenciamento de Atividades

3.3.1 Levantamento de Requisitos

A etapa de levantamento de requisitos é uma das mais importantes para o desenvolvimento de um sistema, é nessa etapa que é definido o conteúdo e as funcionalidades do aplicativo. O aplicativo *web* que foi desenvolvido tem como objetivo prover alguns conteúdos da escola *online* para posteriormente possam ser consultados pelos alunos e também a integração com a *API* Google Maps, fornecendo serviços de localização. Uma etapa importante deste processo é a criação de um mapa da aplicação *web*, para facilitar o desenvolvimento e implementação do projeto.

A figura 7 mostra o mapa da aplicação *web*.

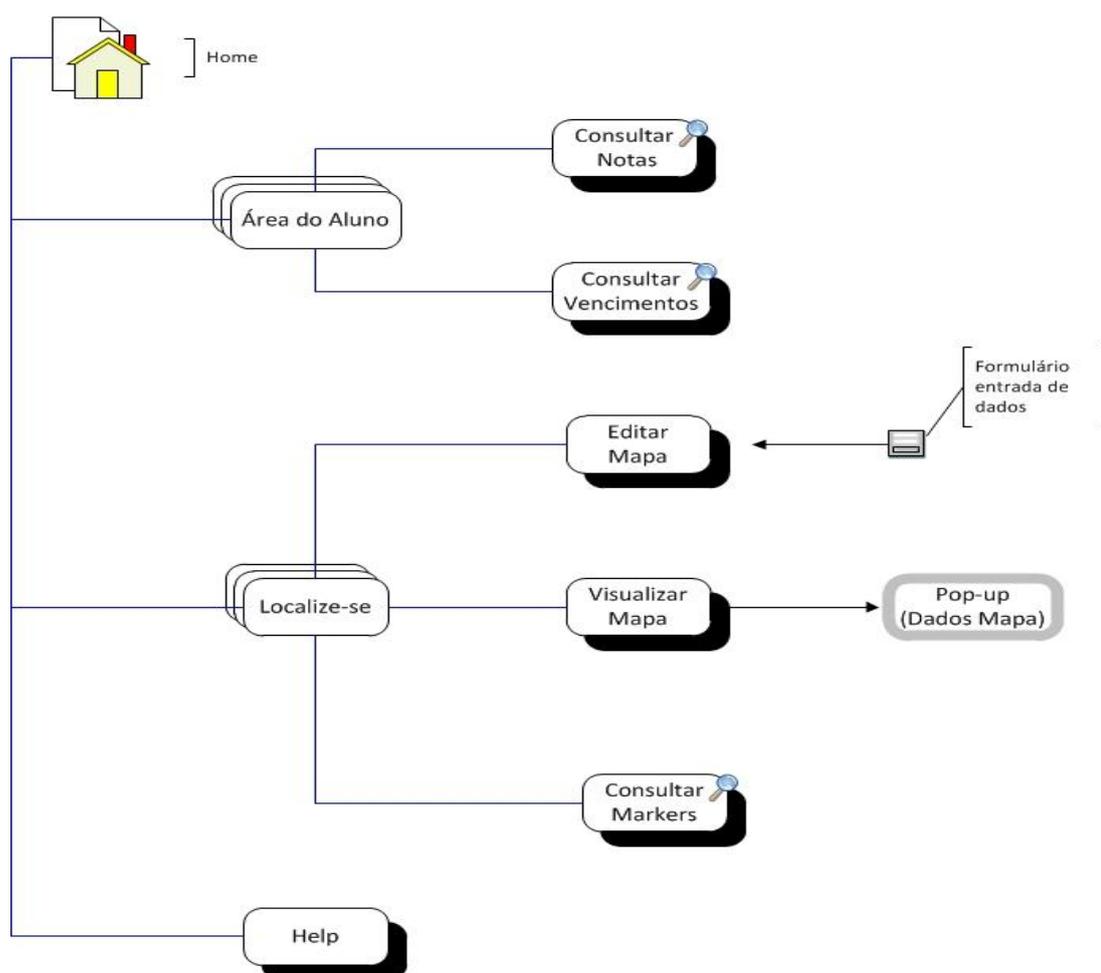


Figura 7 - Mapa do Site

Após o levantamento de requisitos e a certeza dos resultados que se deseja obter com o desenvolvimento do aplicativo é necessário realizar a especificação do modelo proposto no projeto.

3.3.2 Diagrama Entidade Relacionamento (DER)

O diagrama entidade relacionamento é de extrema importância para o bom funcionamento do aplicativo, ele é utilizado para fazer toda a modelagem da base de dados, através dele é possível saber exatamente quais tabelas, campos, seus respectivos tipos e tamanhos são necessários, além dos relacionamentos existentes entre as tabelas.

Este diagrama permite uma visão completa do Sistema de Cadastro de Alunos, destacando as tabelas que são utilizadas para inserção e consulta dos *Markers* e as tabelas utilizadas para consulta de Notas e Vencimentos dos alunos.

A figura 8 mostra o diagrama entidade relacionamento.

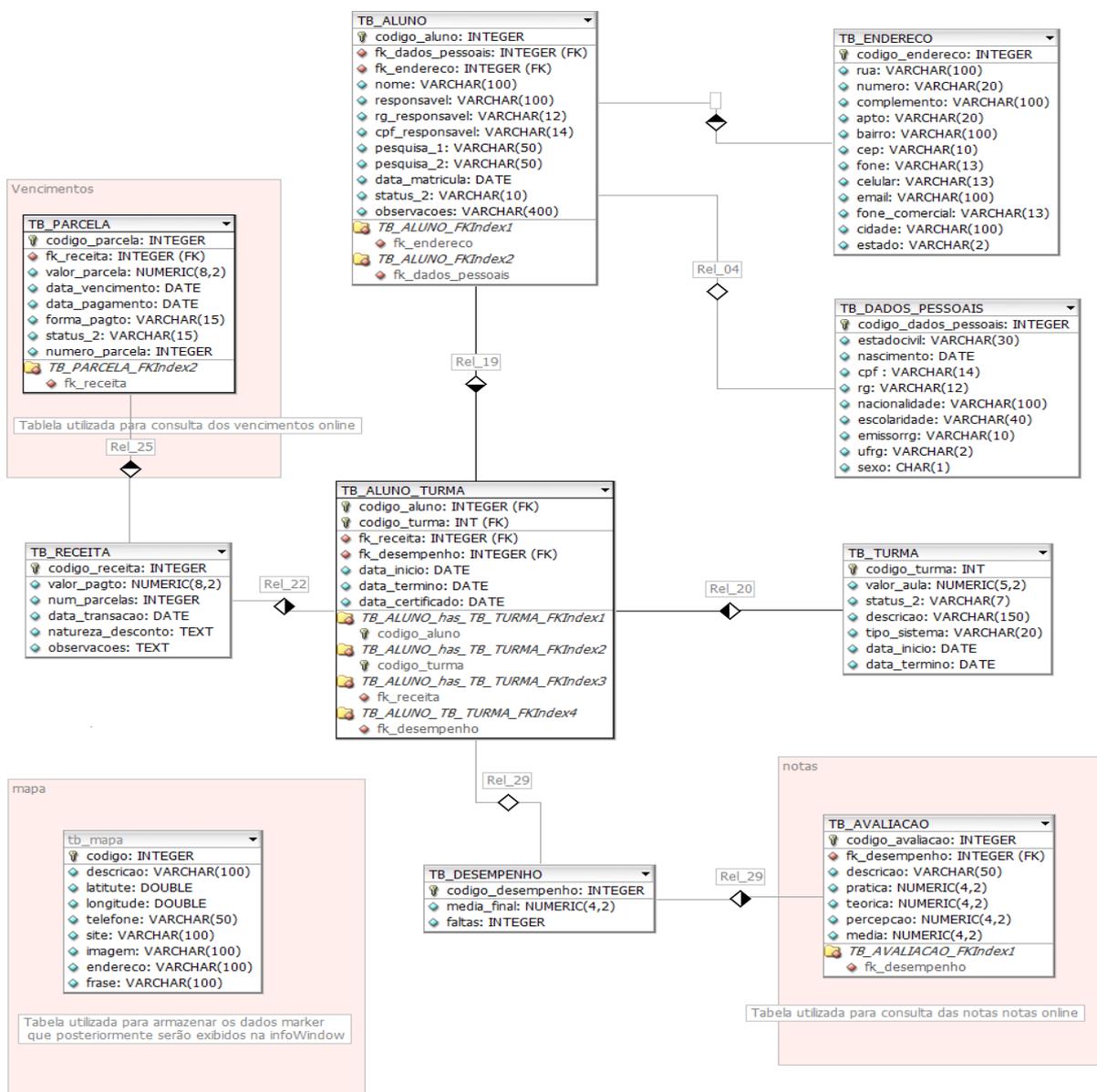


Figura 8 - Diagrama Entidade Relacionamento

3.3.3 Diagramas dos Casos de Uso

O aplicativo *web* possui seis casos de uso, sendo que somente um deles depende de outro caso de uso para executar. A figura 9 mostra o diagrama de casos de uso.

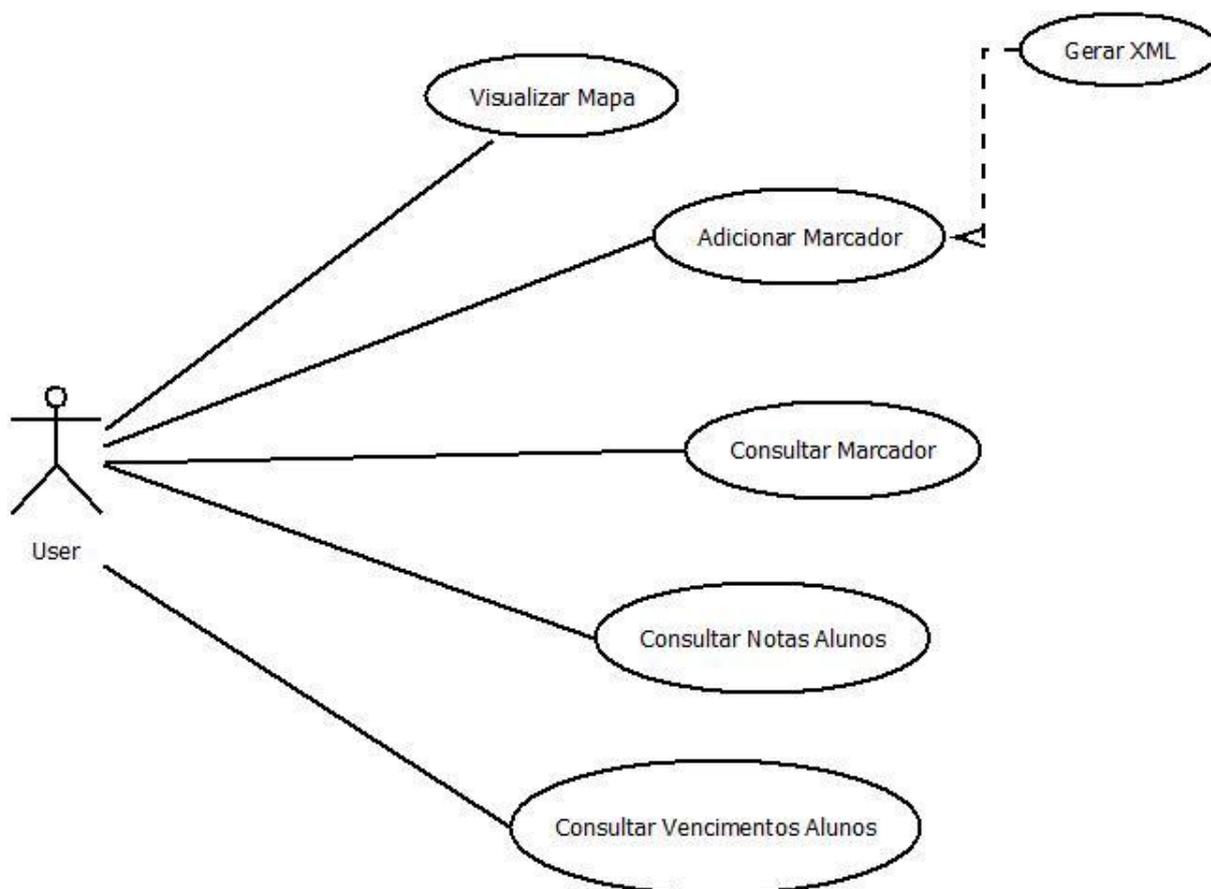


Figura 9 - Diagrama de Casos de Uso

Será feita uma breve descrição de cada um dos casos de uso:

- **Visualizar Mapa:** esse caso de uso é responsável por mostrar no mapa os *markers* cadastrados que ao receber o click do mouse exibem em uma janela pop-up os dados e imagem correspondentes aos mesmos;

- **Adicionar Marcador:** caso de uso que tem como objetivo adicionar um novo marcador recupera através do click do mouse latitude e longitude e fornece para o usuário um formulário para inserção de dados e imagem referente ao novo ponto que será cadastrado;
- **Gerar XML:** esse caso de uso depende da ação do caso de uso adicionar marcador para executar, a cada vez que o caso de uso adicionar marcador é acionado um novo arquivo XML é gerado contendo os novos dados que servirão posteriormente utilizados pela API do Google;
- **Consultar Marcador:** esse caso de uso mostra todos os *markers* já cadastrados e com seus respectivos dados;
- **Consultar Notas Aluno:** esse caso de uso permite que o aluno através de um número de matrícula consulte suas notas no aplicativo;
- **Consultar Vencimentos:** esse caso de uso permite que o aluno através de um número de matrícula consulte seus vencimentos (valores, data e situação) no aplicativo.

3.3.4 Diagrama de Classes

O diagrama de classes também é de suma importância para o entendimento e documentação de um sistema, para isso será feita uma descrição de todas as classes utilizadas nesse projeto, sendo que elas estão separadas em pacotes.

O pacote Bean consiste nas classes que tornam possível a integração entre a linguagem Java e as páginas *web*. A figura 10 mostra o pacote Bean.

- **Pacote Beans**

- **MapBean**: classe responsável pela comunicação da página JSF com os objetos do tipo marcadores;
- **TableMarkerBean**: classe responsável pela comunicação da página JSF com os objetos do tipo marcadores;
- **TableNotasBean**: classe responsável pela comunicação da página JSF com os objetos do tipo NotasFaltas;
- **TableVencimentosBean**: classe responsável pela comunicação da página JSF com os objetos do tipo vencimento.

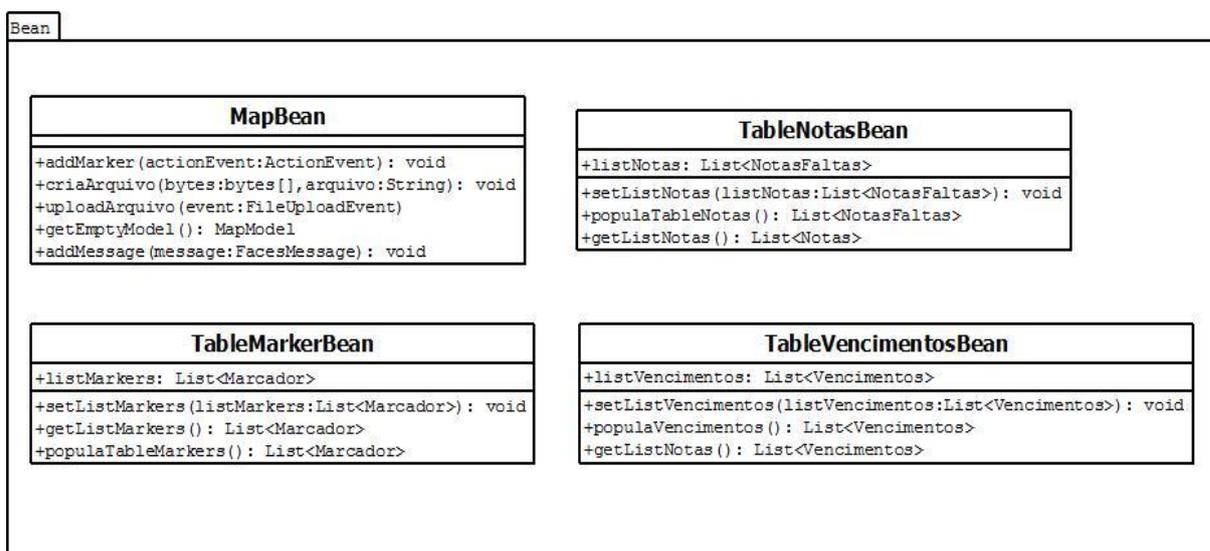


Figura 10 - Pacote Bean

O pacote Model mostra as classes responsáveis por instanciar os objetos que serão utilizados por outras classes. A figura 11 mostra o Pacote Model.

- **Pacote Model**

- **Marcador:** classe responsável por instanciar os marcadores;
- **Vencimentos:** classe responsável por instanciar os vencimentos;
- **Alunos:** classe responsável por instanciar os alunos;
- **NotasFaltas:** classe responsável por instanciar as notas e faltas.

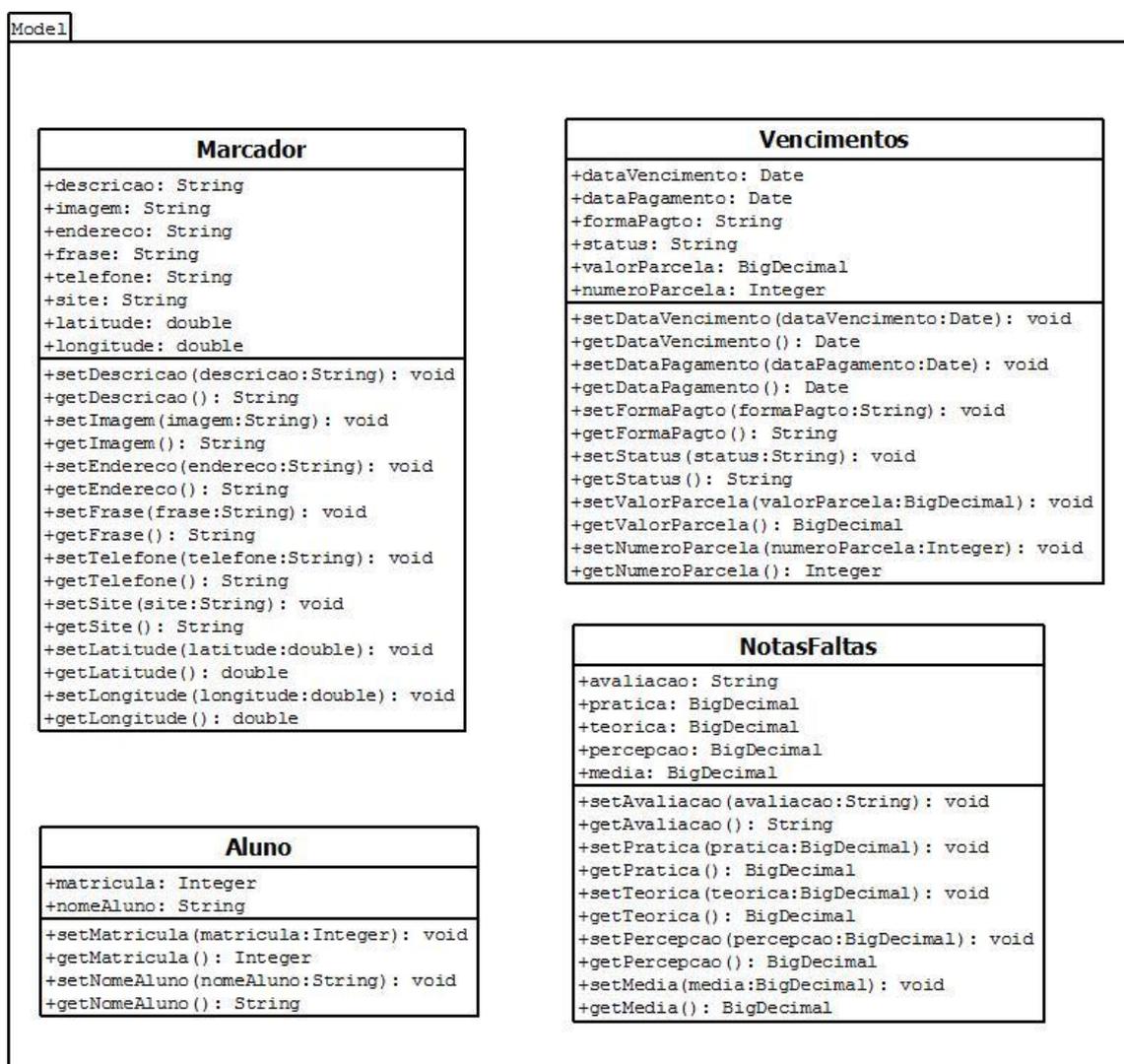


Figura 11 - Pacote Model

O pacote Control traz a classe responsável pela geração do arquivo XML que controla os dados que se integram com a API do Google. A figura 12 mostra o Pacote Control.

- **Pacote Control**

- **MarkerXML:** classe que através dos objetos do tipo marcador gera um arquivo XML.



Figura 12 - Pacote Control

O pacote Dao contém as classes responsáveis pela comunicação entre as classes do pacote Bean e a base de dados. A figura 13 mostra o pacote Dao.

- **Pacote Dao**

- **Conexao:** classe responsável por prover a conexão com a base de dados;
- **MapDao:** classe que contém os métodos para inserir e consultar os marcadores na base de dados;

- **NotasFaltasDao:** classe que contém método responsável pela consulta das notas e faltas na base de dados;
- **VencimentosDao:** classe que contém o método responsável pela consulta de vencimentos.

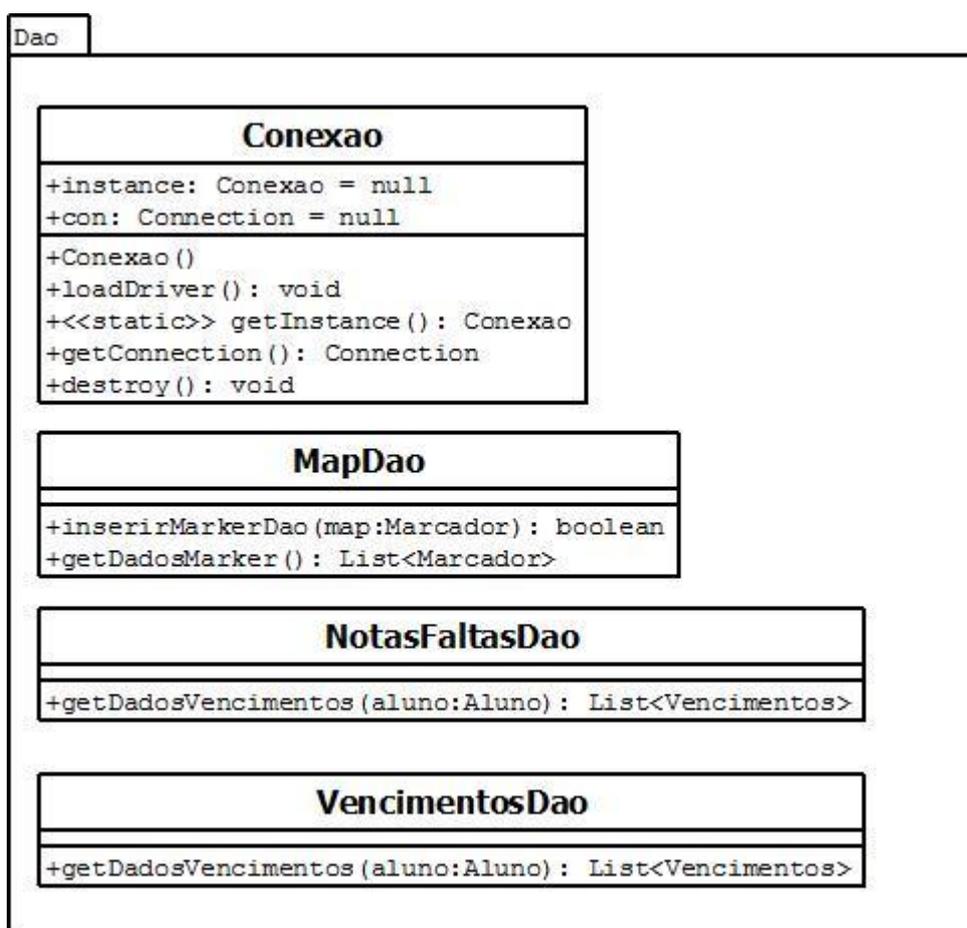


Figura 13 - Pacote Dao

3.3.5 Diagrama de Atividades

Esse tipo de diagrama representa o controle de fluxo de dados de aplicativo, e tem como característica as possíveis tomadas de decisão que possam vir a ocorrer. A figura 14 mostra o diagrama de atividades do aplicativo.

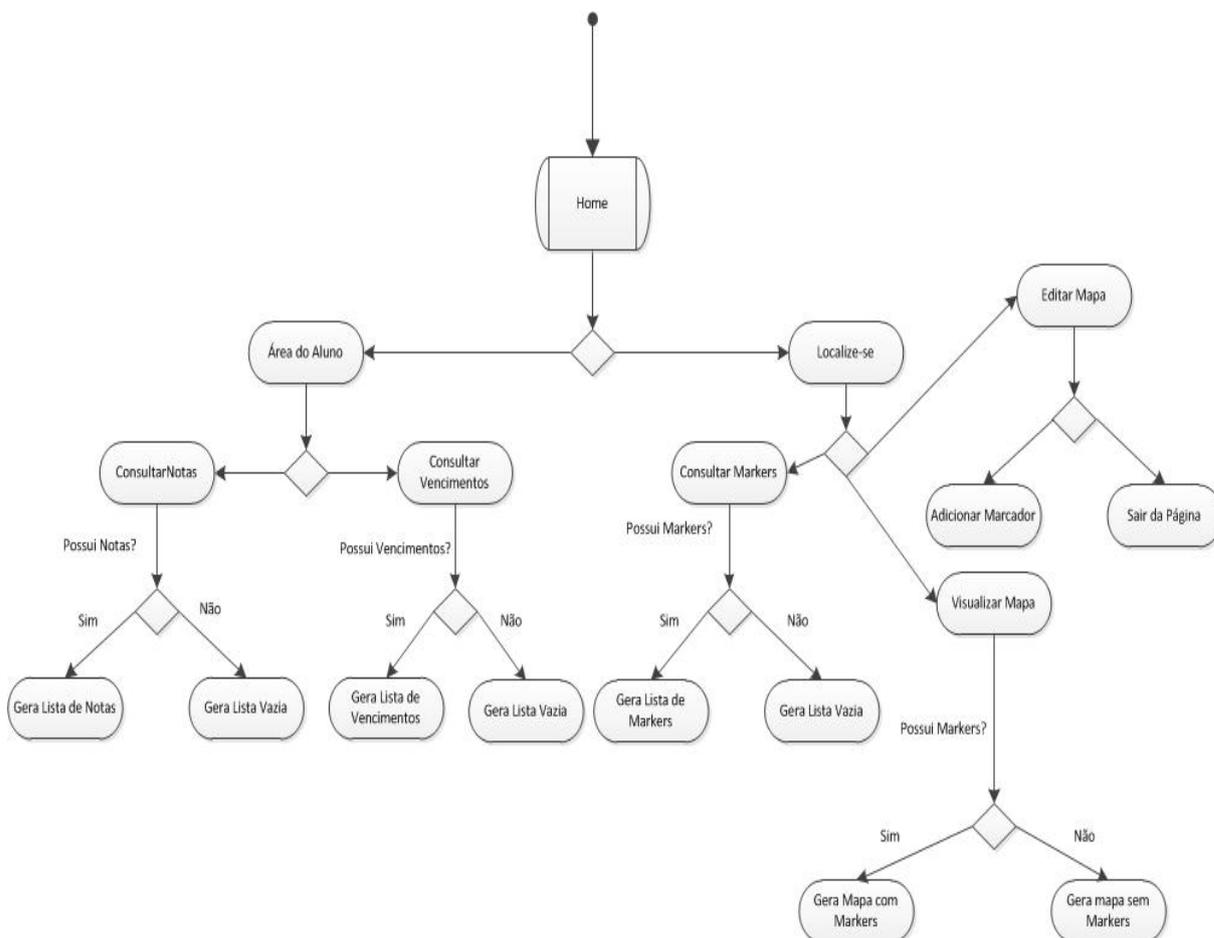


Figura 14 - Diagrama de Atividades

3.3.6 Diagramas de Sequência

Os diagramas de sequência são utilizados para ilustrar a realização dos casos de uso, representando como os objetos interagem para executar o comportamento total

ou parcial destes casos de uso. Para cada caso de uso é necessário um diagrama para representar as possíveis ações.

3.3.7 Adicionar Marcador

O diagrama da figura 15 mostra as possíveis ações do caso de uso adicionar marcador, como este caso de uso possui um caso de uso dependente do método do caso de uso GerarXML também foi incluído nesse diagrama.

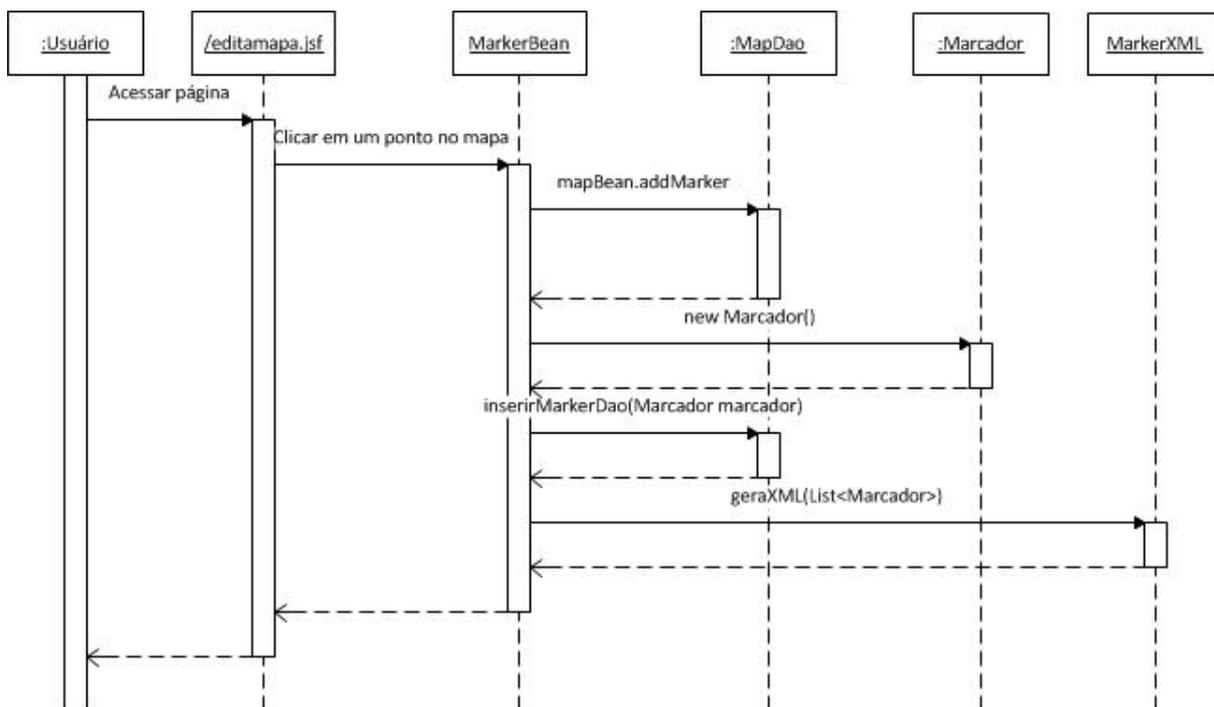


Figura 15 - Diagrama de Sequência Adicionar Marcador

3.3.8 Visualizar Mapa

O diagrama da figura 16 mostra as possíveis ações que podem ser executadas pelo caso de uso Visualizar mapa. Nesse caso de uso ocorre especificamente a integração com a API Google Maps, onde o método GDownloadUrl() que é capaz de ler arquivos ou URL's(desde que em formatos suportados) faz a leitura do arquivo XML possibilitando a visualização dos marcadores previamente cadastrados.

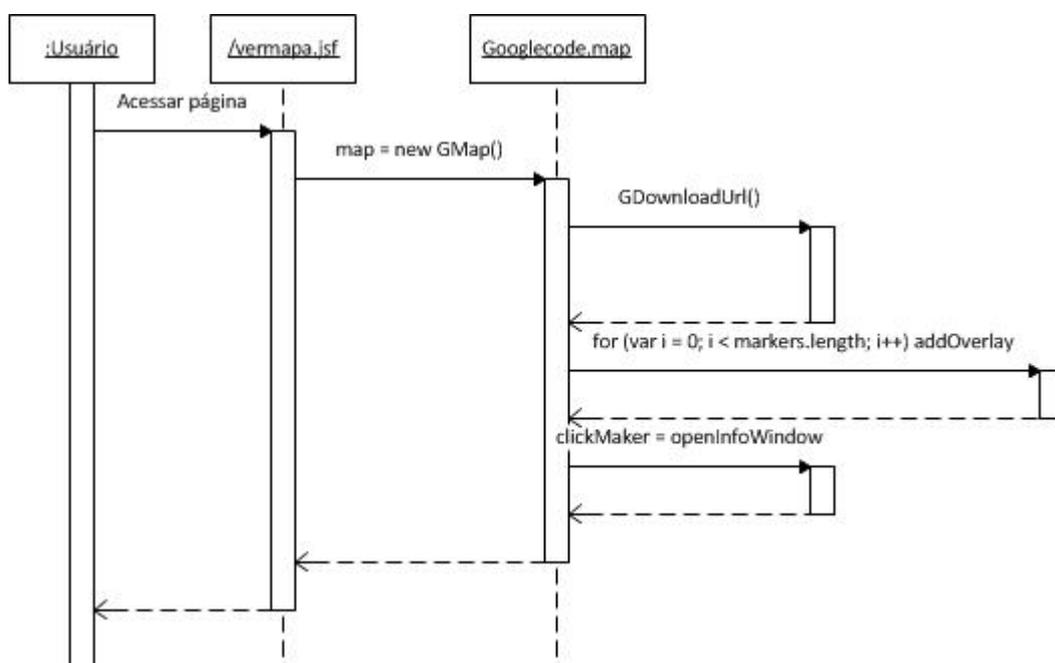


Figura 16 - Diagrama de Sequência Visualizar Mapa

3.3.9 Consultar Marcador

O diagrama da figura 17 mostra as possíveis ações que podem ser executadas pelo caso de uso Consultar Marcador.

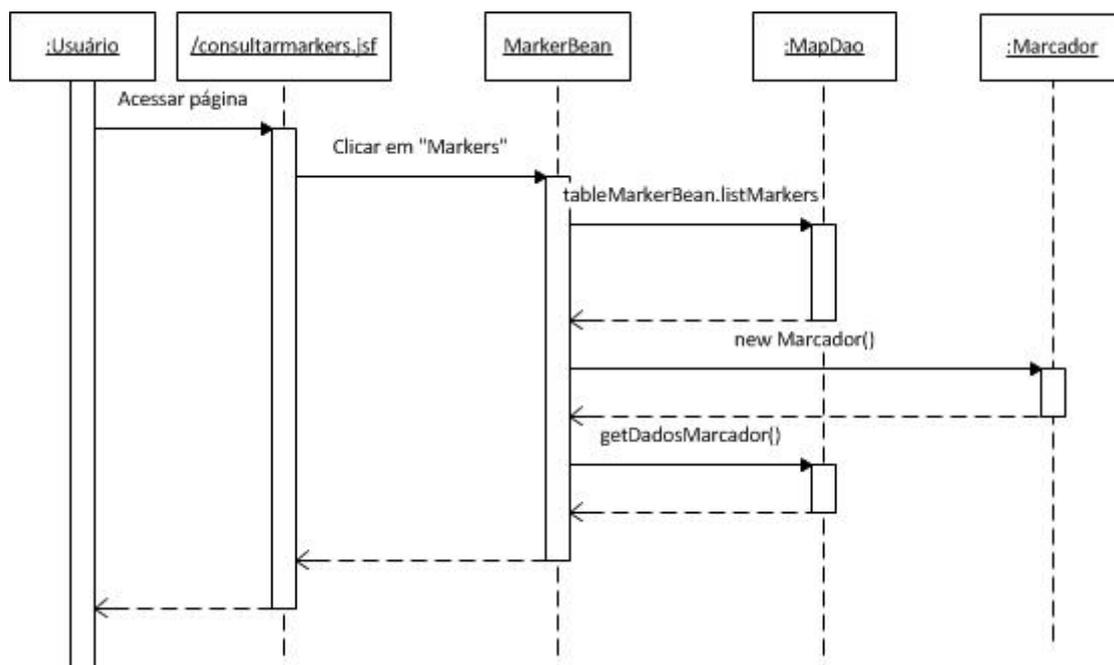


Figura 17 - Diagrama de Sequência Consultar Marcador

3.3.10 Consultar Notas Alunos

O diagrama da figura 18 mostra as possíveis ações do caso de uso Consultar Notas Alunos.

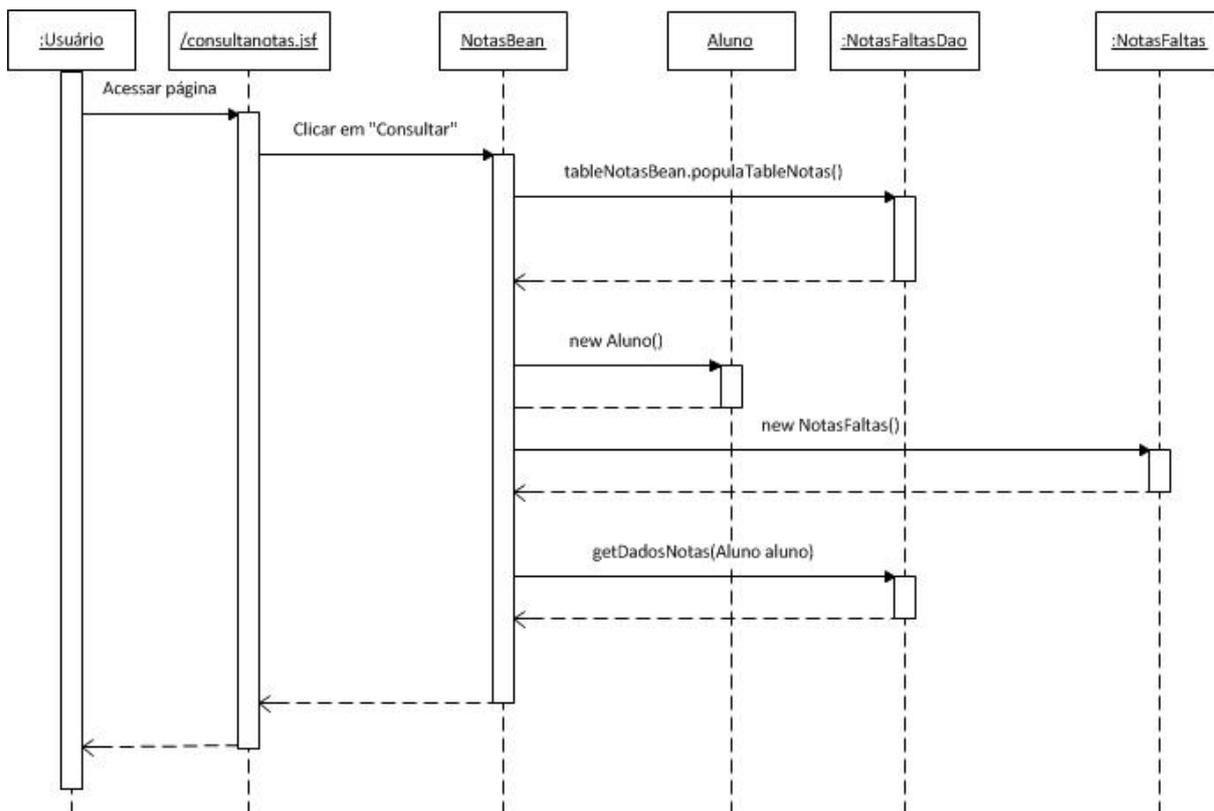


Figura 18 - Diagrama de Sequência Consultar Notas Alunos

3.3.11 Consultar Vencimentos

O diagrama da figura 19 mostra as possíveis ações que podem ser executadas pelo caso de uso Consultar Vencimentos.

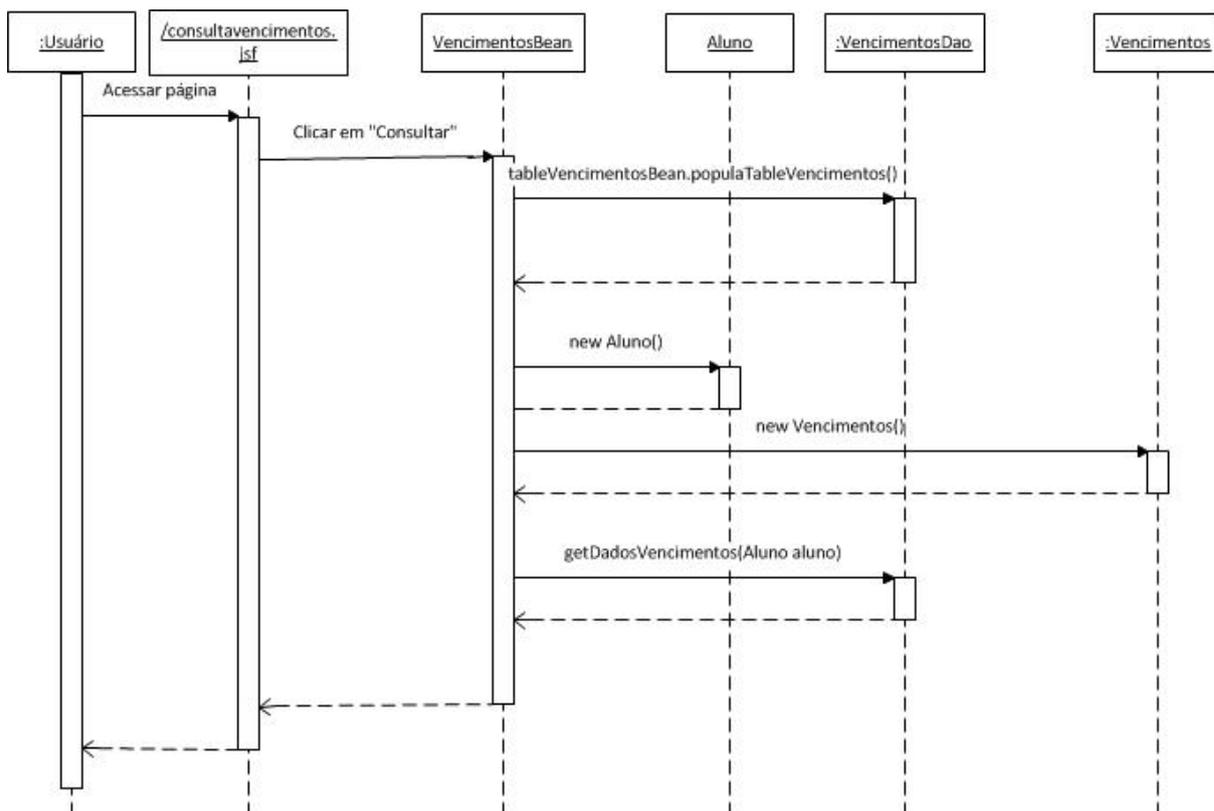


Figura 19 - Diagrama de Sequência Consultar Vencimentos

3.4 IMPLEMENTAÇÃO DO APLICATIVO

Nesta seção será feita a descrição da implementação do aplicativo *web* integrado com a Google Maps API.

3.4.1 Criação da Base de Dados

Para esse projeto foi utilizado o PostgreSQL um SGBD de nível corporativo, gratuito e totalmente compatível com a linguagem de programação Java, a criação da base de dados foi feita respeitando o diagrama entidade relacionamento da figura 8.

3.4.2 Criação de Interface

A interface de comunicação com o usuário foi desenvolvida utilizando os recursos de JSF (Java Server Faces) e Ajax, a biblioteca Primefaces também foi adicionada ao projeto com o intuito de trazer um aplicativo com visualização mais agradável ao usuário. Toda estrutura de programação foi desenvolvida na IDE Eclipse Helios que provê vários recursos ao programador e também é gratuita.

3.4.3 Integração com a Api Google Maps

A Google fornece um API que possibilita visualização de mapas de todas as regiões do mundo, a API é desenvolvida em JavaScript. A Google permite que uma parte “externa” desse conteúdo seja manipulada, possibilitando através de funções fornecidas pela própria API a inserção de marcadores e outras funções personalizando assim os mapas. A ideia dessa integração em suma é a utilização da API da Google mais com uma base de dados própria em um sistema intuitivo onde o usuário final possa utilizar recursos da API sem possuir conhecimentos avançados. Para isso foi integrado ao aplicativo páginas que carregam os mapas da API e possibilitam essas alterações. A comunicação com a API do Google é feita através de um arquivo XML que transporta os dados armazenados em uma base, esse arquivo é recebido como um parâmetro em função da API (`GDownloadUrl`) é feito então um *loop* iterando a quantidade de vezes necessárias e lançando os marcadores no mapa.

3.4.4 Funcionamento do Aplicativo

O objetivo desta seção é mostrar o funcionamento do aplicativo, explicando cada funcionalidade e mostrando a interface de comunicação com o usuário em ação. Ressaltando que o aplicativo foi desenvolvido no ambiente de programação Eclipse Helios 3.5.

3.4.4.1 Interface Principal

A interface principal do aplicativo (*Home Page*) disponibiliza os menus:

- Home
- Área do Aluno
- Localize-se
- Help

A figura 20 traz a interface principal do aplicativo.

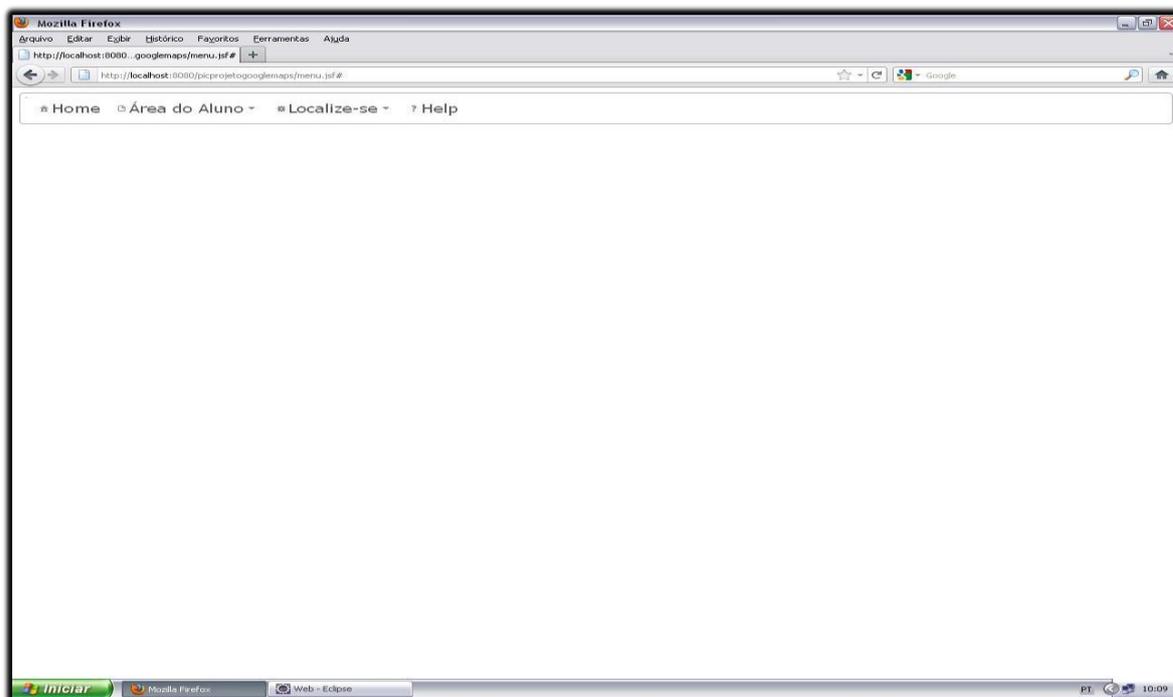


Figura 20 - Interface Principal do Aplicativo

3.4.5 Área Do Aluno

A Interface principal apresenta o item Área do Aluno, onde se encontram dois submenus, que servem para que os alunos através de um número de matrícula possam consultar suas notas e a situação de seus vencimentos. Os submenus que se encontram na área do aluno são:

- Notas
- Vencimentos

A figura 21 mostra os itens da área do aluno.

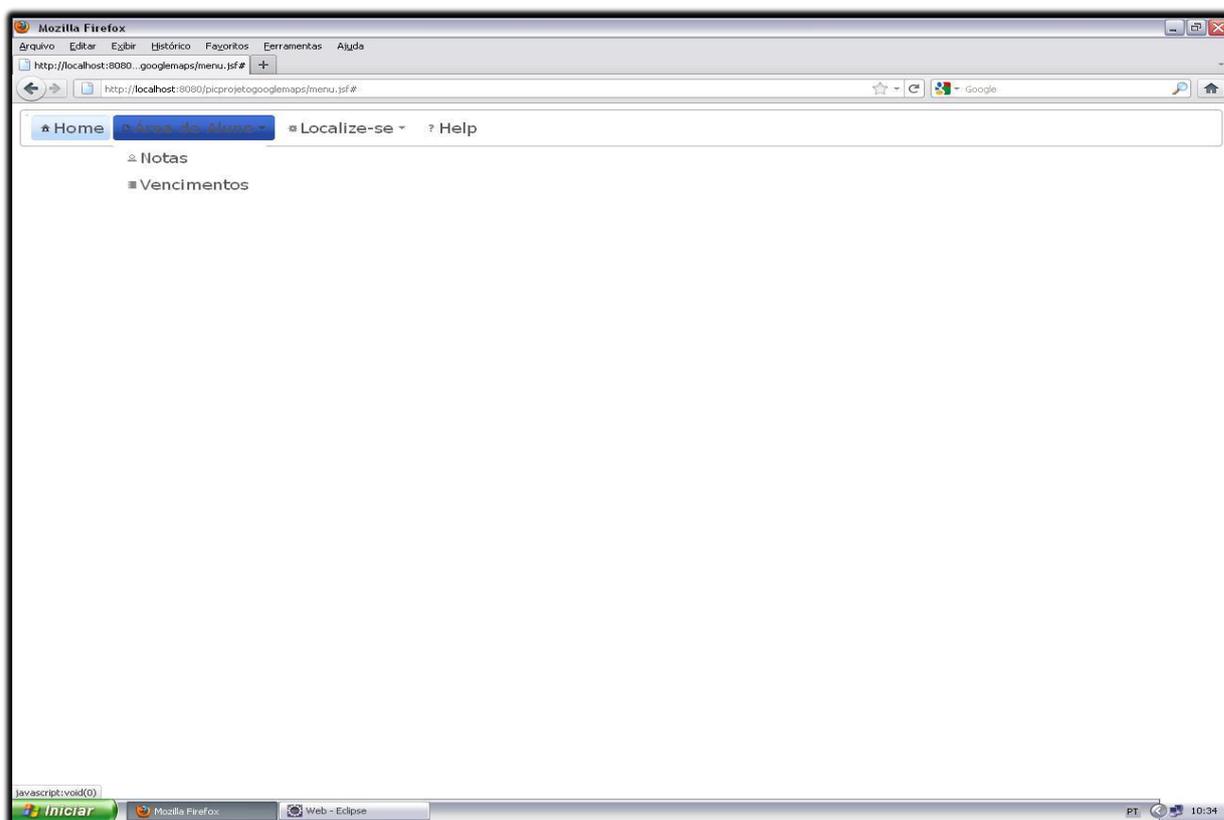


Figura 21 - Menu Área do Aluno

3.4.5.1 Consulta de Notas

No submenu Notas, o aluno insere seu número de matrícula e ao selecionar consultar acionando o caso de uso Consultar Notas Alunos, uma tabela com os dados das avaliações é exibida. A figura 22 mostra a consulta de notas em ação.

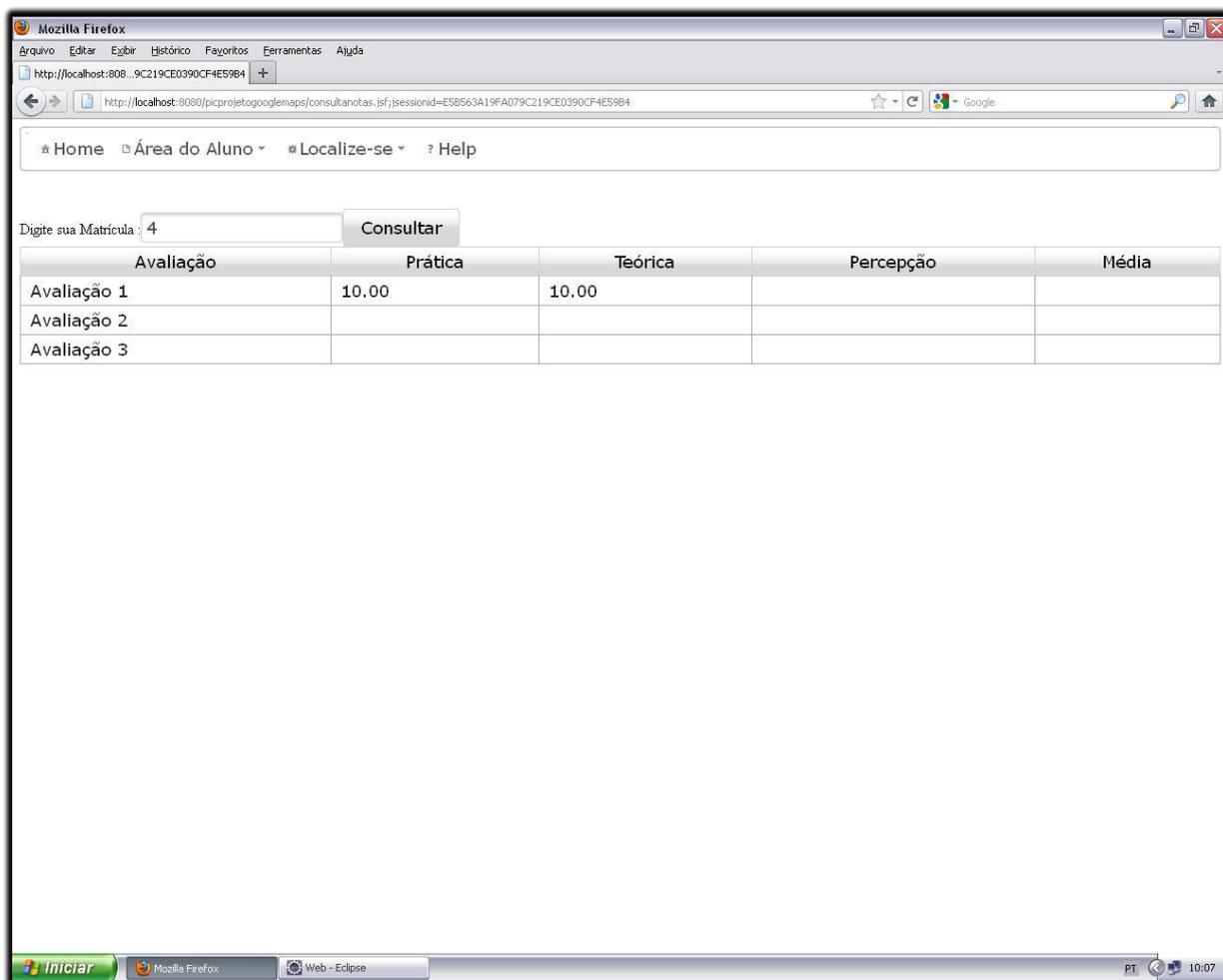
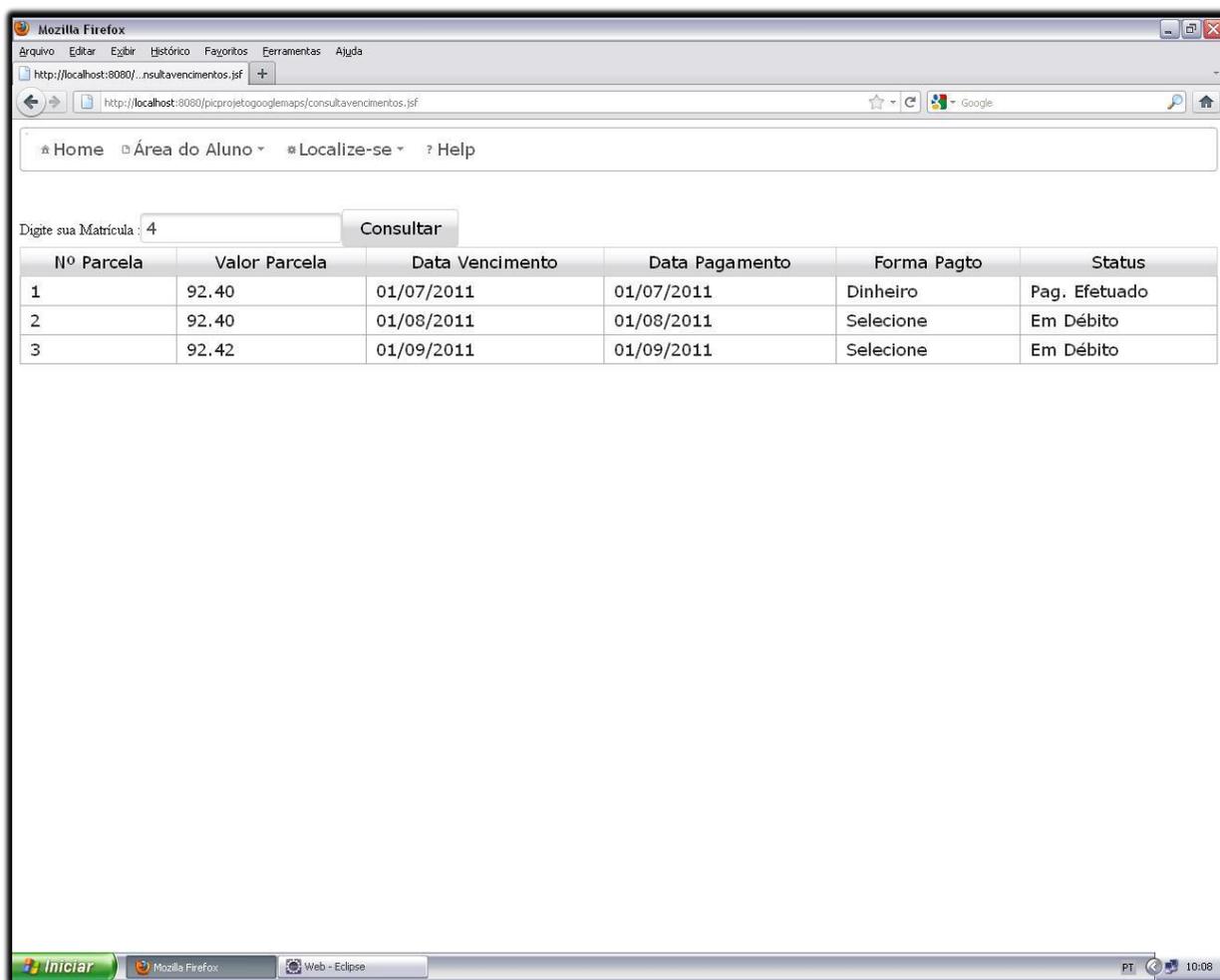


Figura 22 - Área do Aluno - Consulta de Notas

3.4.5.2 Consulta de Vencimentos

No submenu Vencimentos o aluno insere seu número de matrícula e ao selecionar consultar acionando o caso de uso Consultar Vencimentos, uma tabela com as parcelas e seus respectivos dados é exibida. A figura 23 mostra a consulta de vencimentos em ação.



Nº Parcela	Valor Parcela	Data Vencimento	Data Pagamento	Forma Pagto	Status
1	92.40	01/07/2011	01/07/2011	Dinheiro	Pag. Efetuado
2	92.40	01/08/2011	01/08/2011	Selecione	Em Débito
3	92.42	01/09/2011	01/09/2011	Selecione	Em Débito

Figura 23 - Área do Aluno - Consulta de Vencimentos

3.4.6 Localize-se

No menu principal encontramos o item Localize-se, nesse menu estão as páginas integradas com a API do Google Maps, que possibilitam o cadastro e visualização dos marcadores no mapa. O item localize-se disponibiliza os seguintes submenus:

- Visualizar Mapa
- Editar Mapa
- Markers

A figura 24 mostra os itens do Menu Localize-se.

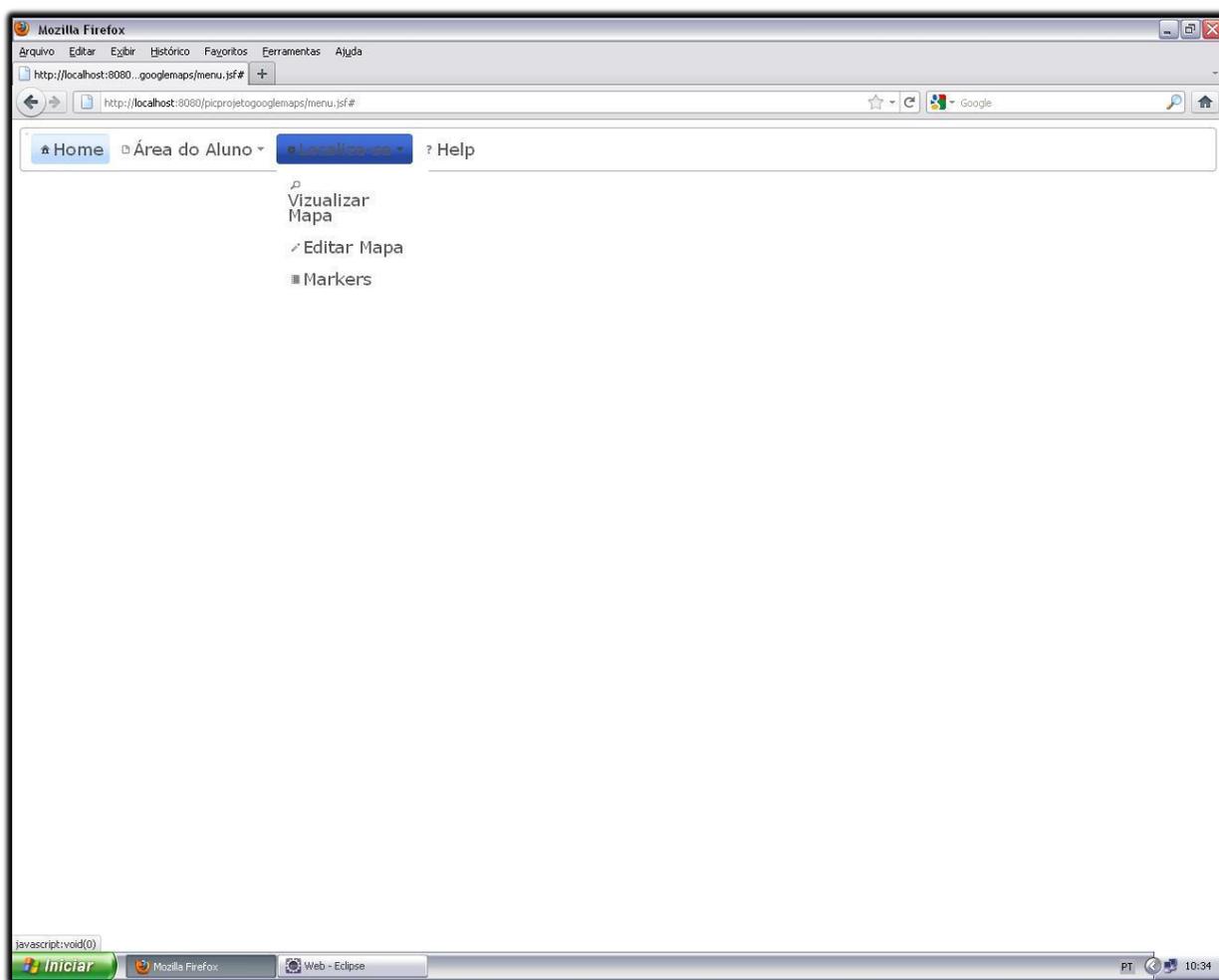


Figura 24 - Menu Localize-se

3.4.6.1 Editar Mapa

Ao selecionar no item editar mapa, uma página com o mapa da API Google Maps é carregado, nesse mapa o usuário seleciona em um determinado ponto a latitude e longitude e estas são recuperadas de maneira transparente ao usuário e um formulário para inserção de dados é carregado na interface. A figura 25 mostra o cadastro de um marcador em funcionamento.

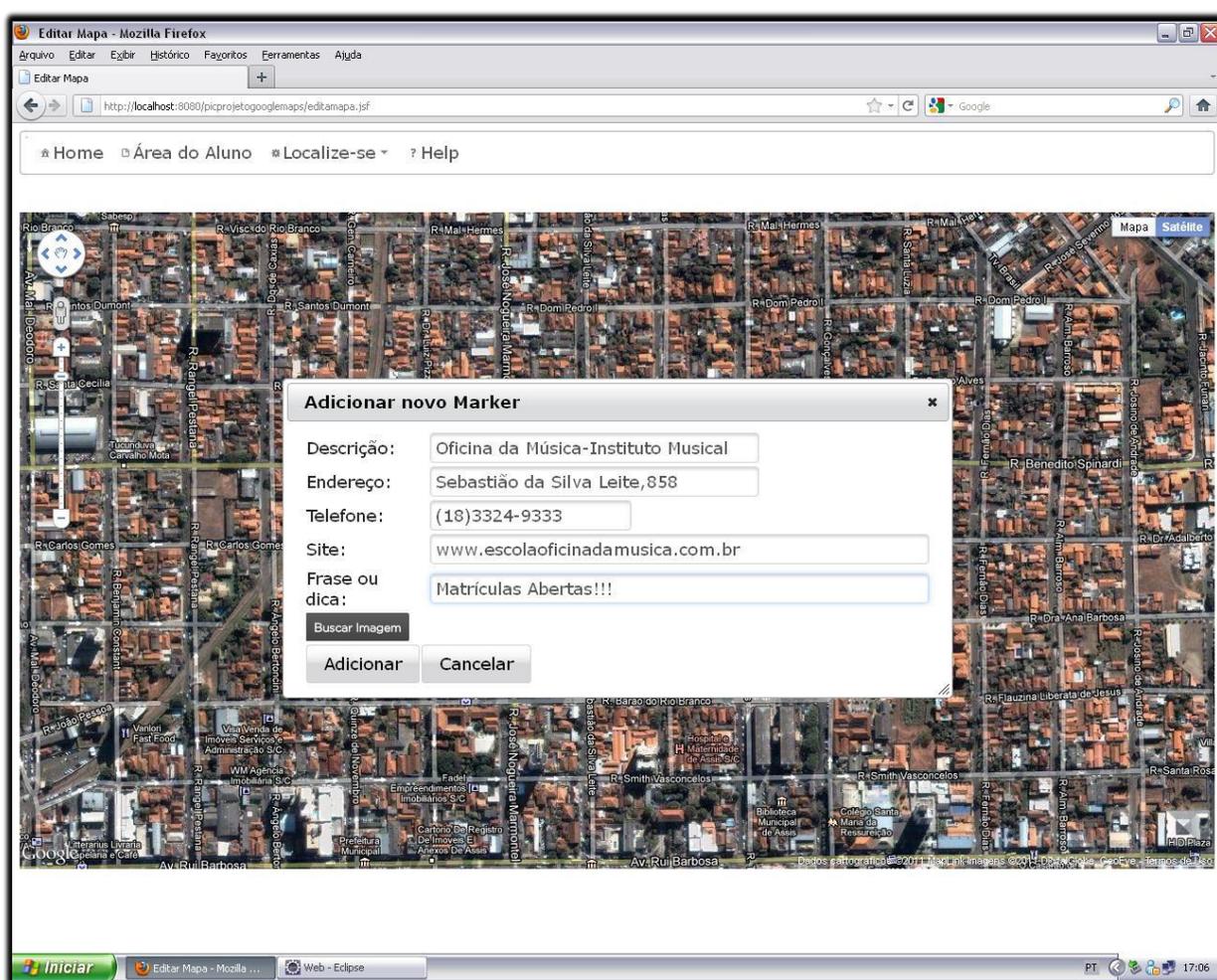


Figura 25 - Inserção de um novo Marcador

3.4.6.2 Visualizar Mapa

Ao selecionar no item visualizar mapa, caso haja marcadores cadastrados os mesmos são exibidos no mapa, ao selecionar um marcador os dados que foram previamente cadastrados são exibidos em uma *infoWindow* (pop-up fornecido pela API). A figura 26 mostra a exibição de um marcador e sua respectiva *infoWindow* em ação.

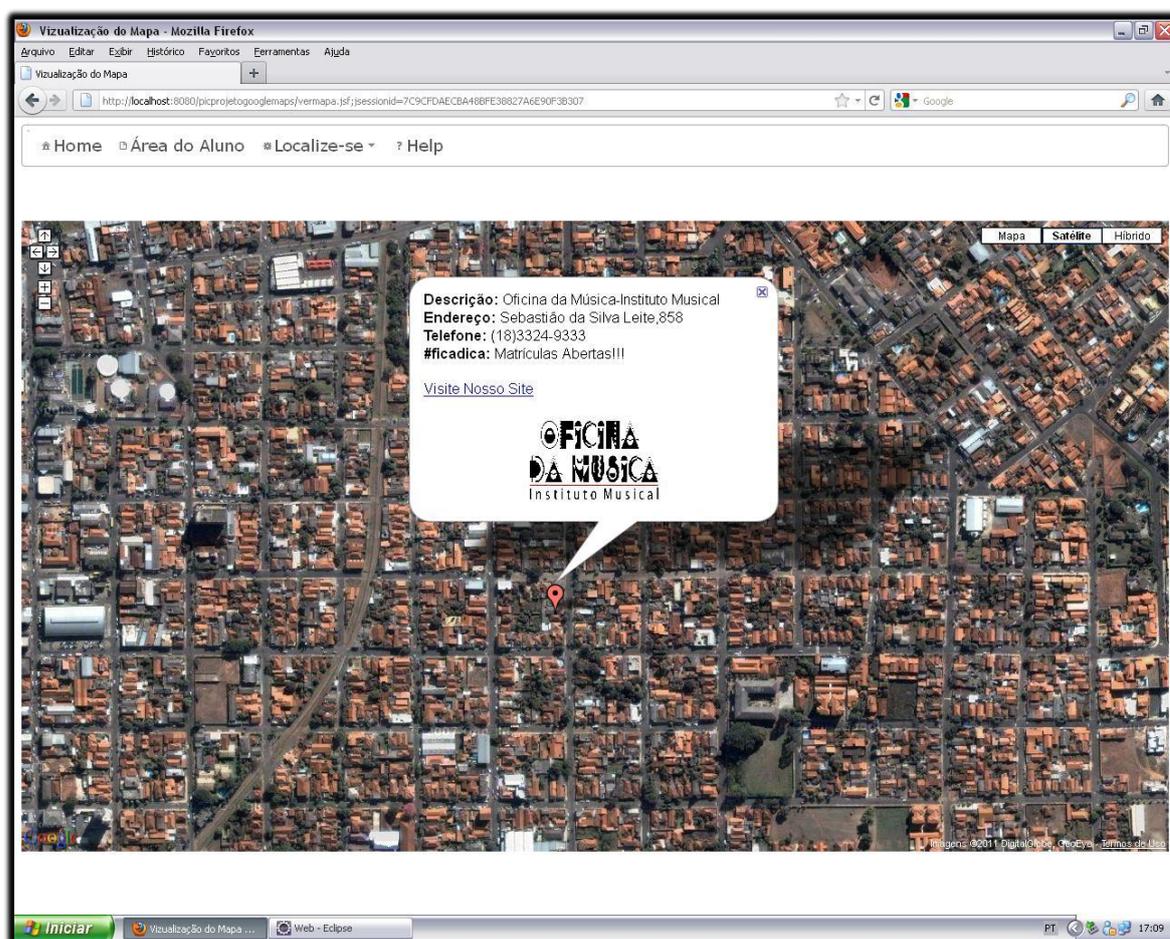


Figura 26 - Visualização de Marcador com infoWindow

Na figura 27 é mostrada a localização da Prefeitura de Assis com o mapa no modo “satélite” .

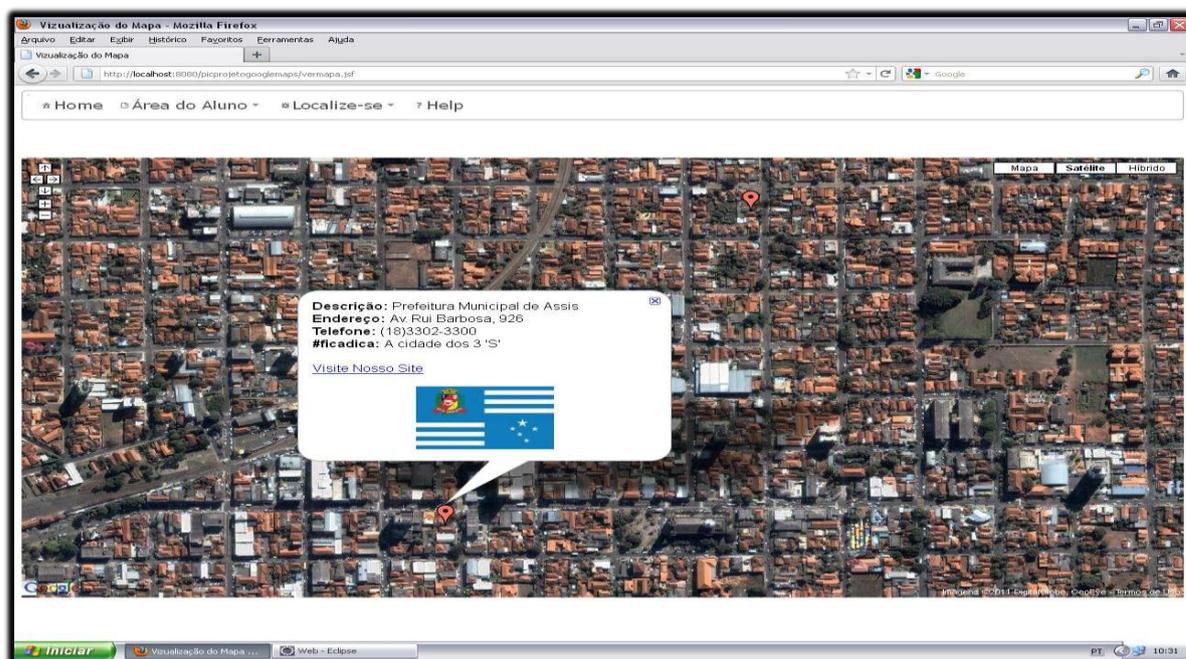


Figura 27 - Localização Prefeitura de Assis modo Satélite

Na figura 28 a mesma localização com o mapa no modo “mapa” .

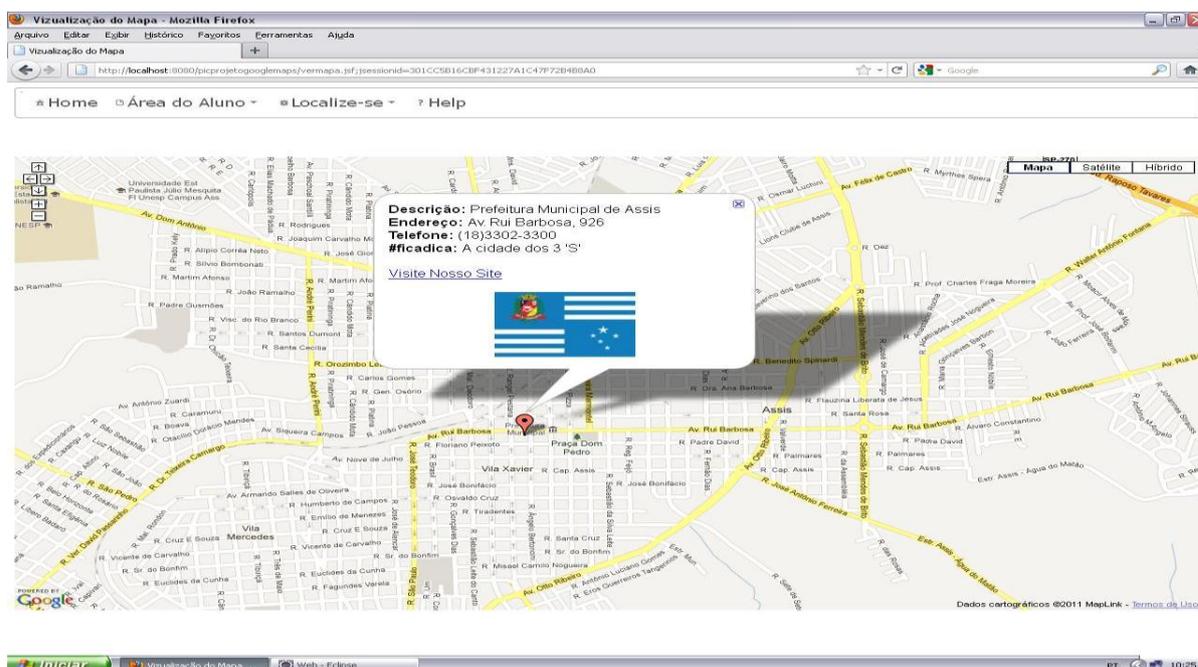
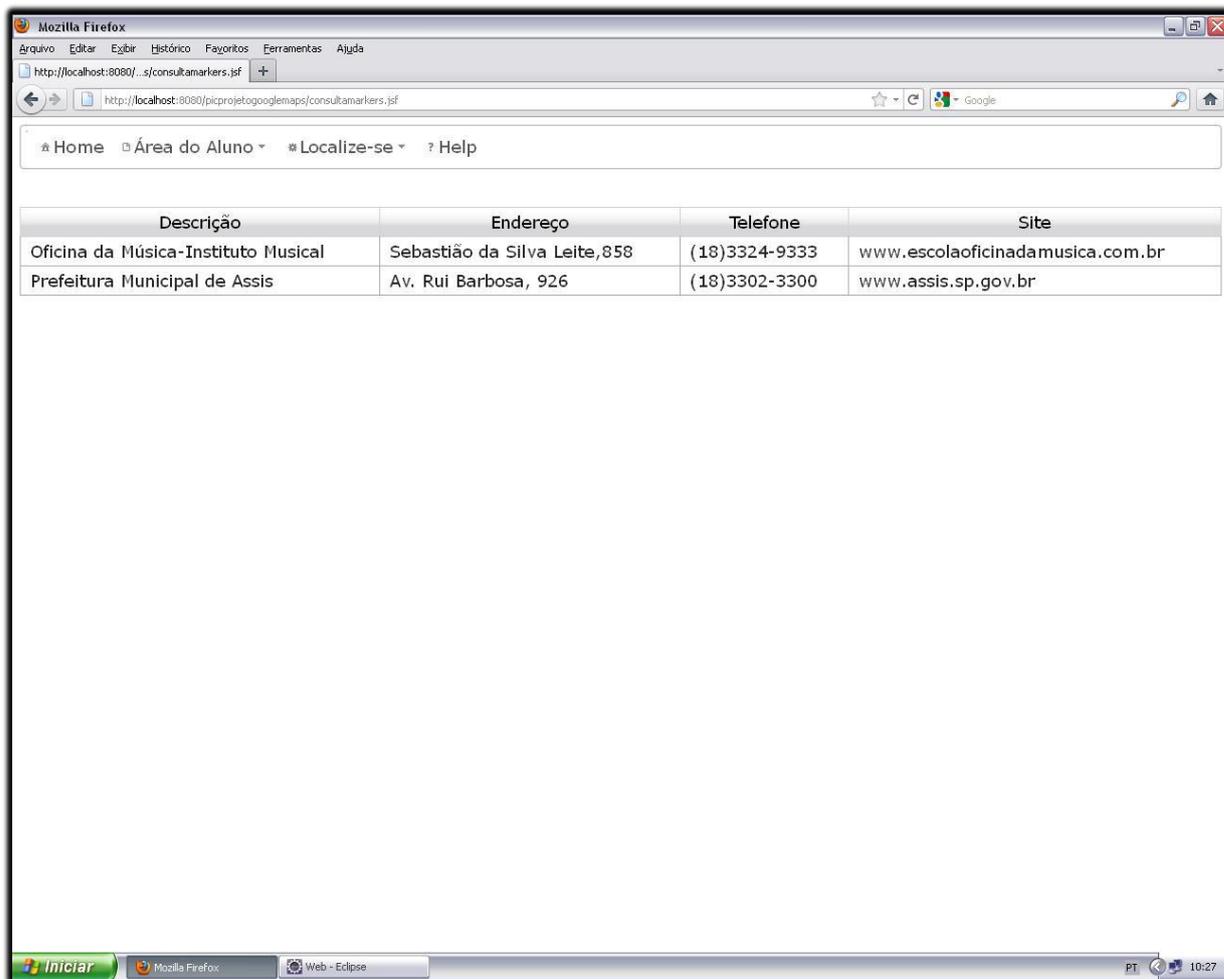


Figura 28 - Localização Prefeitura de Assis modo mapa

3.4.7 Markers

Ao selecionar no item *markers* o caso de uso Consultar Marcador é acionado, uma tabela com os dados referentes aos marcadores cadastrados é exibida. A figura 29 mostra a consulta dos marcadores.



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://localhost:8080/picprojeto/googlemaps/consultamarkers.jsf`. The page content includes a navigation menu with 'Home', 'Área do Aluno', 'Localize-se', and 'Help'. Below the menu is a table with the following data:

Descrição	Endereço	Telefone	Site
Oficina da Música-Instituto Musical	Sebastião da Silva Leite, 858	(18)3324-9333	www.escolaoficinadamusica.com.br
Prefeitura Municipal de Assis	Av. Rui Barbosa, 926	(18)3302-3300	www.assis.sp.gov.br

Figura 29 - Localize-se Consulta Marcadores

4 CONCLUSÃO

A conclusão desse projeto possibilita que o aplicativo desenvolvido realize a localização das escolas no mapa de maneira automática e em tempo real e os serviços oferecidos na proximidade do local, bem como outros serviços *on-line* como a consulta de notas e vencimentos dos alunos. Para facilitar a reutilização dos algoritmos em futuros projetos, o aplicativo foi construído de maneira modular.

Durante o desenvolvimento deste projeto foram obtidos conhecimentos sobre várias áreas tecnológicas. Para por em prática o andamento e conclusão do projeto foram necessário estudos sobre desenvolvimento *web*, geocodificação, Linguagem de Programação Java para Web, estrutura de arquivo XML e um estudo profundo sobre o funcionamento da API Google Maps para usufruir da melhor maneira os recursos fornecidos pela API da Google.

Também foram realizadas pesquisas para utilização dos Frameworks e IDE's de desenvolvimento e gerenciamento utilizados no projeto.

É inegável a importância da realização desse projeto de pesquisa no que diz respeito ao crescimento e amadurecimento acadêmico e também na utilização dos conhecimentos obtidos para um futuro uso profissional.

Por fim, deseja se que essa aplicação, sirva de base para eventuais produtos na área de localização e determinação de rotas, que é uma área que vem sendo cada vez mais estudada e incentivada por iniciativas públicas e privada, além é claro de servir como base para futuros estudos que envolvam os conhecimentos aqui apresentados.

REFERÊNCIAS

ALONSO, G; CASATI, F; KUNO, H. e MACHIRAJU, V.: **Web Services Concepts – Architectures and Applications**. Springer-Verlag, 2004.

ALVAREZ, M. A.: **Objetivos e usos do XML**. Disponível em: <<http://www.criarweb.com/artigos/431.php>>. Acesso em setembro de 2011.

APACHE TOMCAT.:**Website oficial Apache Tomcat**. Disponível em: <<http://tomcat.apache.org/>>. Acesso em setembro de 2011.

ARMSTRONG, E; BALL, J; BODOFF S.: **The J2EE Tutorial**. Sun Microsystems, 2005.

AZEVEDO, C.: **Imasters: Por Uma Internet Mais Criativa e Dinâmica**, Disponível em: <http://imasters.uol.com.br/artigo/7832/programacao/google_maps_api/>. Acesso em maio de 2011.

BARALE, R. F.: **Desenvolvimento de um sistema de Vendas na web**. Uberlândia, 2007.

BONFIN JUNIOR, F.: **JSP –Java Server Pages – A tecnologia Java na Internet**. 2ed. São Paulo: Editora Érica, 2002.

DEITEL, H.M., **Java como programar**, trad. Carlos Arthur Lang Lisbôa, 4ed. Bookman, Porto Alegre, 2003.

DRJT.: **DeveloperResources for Java Technology**. Disponível em: <<http://java.sun.com/>>. Acesso em setembro de 2011.

FURTADO JUNIOR, M. B.: **Tutorial XML - ExtensibleMarkupLanguage**. Universidade Federal do Rio de Janeiro, Disponível em: <http://www.gta.ufrj.br/grad/00_1/miguel/index.html>. Acesso em outubro de 2011.

GIGAFLOPS.: **Java Server Pages**. Disponível em: <<http://gigaflops.tripod.com/page/lp/jsp/jsp.html> >. Acesso em agosto de 2011.

GINIGE, A., MURUGESAN, S., **Web Engineering: an Introduction**, IEEE Multimedia, Vol. 8, Issue: 1, 2001.

GLORIA, A.I.; CAMPELO, S.S. e RIBEIRO, J.A., **Como utilizar as API's do Google Maps**, Anais do XIII Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, 2007.

GOOGLE DEVELOPER GUIDE.:**Google Maps Java API Data**, Disponível em: <http://code.google.com/intl/pt-BR/apis/maps/documentation/mapsdata/developers_guide_java.html>. Acesso em junho de 2011.

GOOGLE MAPS. Disponível em: <<http://maps.google.com.br/maps?ct=reset>>. Acesso em setembro de 2011.

HALL, M.; BROWN, L., **Core Servlets and Java Server Pages**. 2^a ed. Sun Microsystems Press/Prentice Hall PTR Book, 2003.

KURNIAWAN, B., **Java para a Web com Servlets, JSP, EJB**, 1a edição, 2002.

LEMAY, L. e PERKINS, C.L., **Teach Yourself Java in 21 Days**, Sams.net Publishing, 1996.

LIMEIRA, J. L. S.. **Utilização de AJAX no desenvolvimento de sistemas Web**. Porto Alegre: 2006.

LIU, S.: **Getting Started with the Google Data Java Client Library**, 2007, Disponível em: <http://code.google.com/intl/pt-BR/apis/gdata/articles/java_client_lib.html>. Acesso em agosto de 2011.

PostgreSQL.: **Sobre o PostgreSQL** Disponível em: <<http://www.postgresql.org.br/sobre>>. Acesso em setembro de 2011.

PRESSMAN, R.**Engenharia de Software**, Rio de Janeiro, McGraw Hill, 2002.

RODRIGUES, M. C.: **Encapsulando a NavigationPlan Tool como Serviço Web**. IME – USP, 2008.

SILVEIRA, P. E. A. e COSENTINO, R. A.: **FJ-21 Java para desenvolvimento web**. Caelum Ensino e Soluções em Java, 2009.

SOUZA NETO, W. P.: **Usando Api do Google Maps para criar um mapa interativo**, Universidade Federal de Viçosa, 2009.

W3SCHOOLS.: **XML Tutorial**. Disponível em: <<http://www.w3schools.com/xml/>>. Acesso em setembro de 2011.