

UNIVERSIDADE FEDERAL DE ALFENAS

INSTITUTO DE CIÊNCIAS EXATAS

Bacharelado em Ciência da Computação

Luiz Paulo Ferreira Bianco

GPSEARCH

**UM APLICATIVO PARA CONSULTA DE EVENTOS NO
CAMPUS DE ALFENAS DA UNIFAL-MG VIA
LOCALIZAÇÃO GLOBAL**

Alfenas, 27 de Março de 2013.

UNIVERSIDADE FEDERAL DE ALFENAS

INSTITUTO DE CIÊNCIAS EXATAS

Bacharelado em Ciência da Computação

GPSEARCH

**UM APLICATIVO PARA CONSULTA DE EVENTOS NO
CAMPUS DE ALFENAS DA UNIFAL-MG VIA
LOCALIZAÇÃO GLOBAL**

Luiz Paulo Ferreira Bianco

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Alfenas como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Orientador: Prof. Rodrigo Martins Pagliares

Alfenas, 27 de Março de 2013.

Luiz Paulo Ferreira Bianco

GPSEARCH
UM APLICATIVO PARA CONSULTA DE EVENTOS NO
CAMPUS DE ALFENAS DA UNIFAL-MG VIA
LOCALIZAÇÃO GLOBAL

A Banca examinadora abaixo-assinada aprova a monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

Prof. Humberto César Brandão
Universidade Federal de Alfenas

Prof. Douglas Donizeti De Castilho Braz
Universidade Federal de Alfenas

Prof. Rodrigo Martins Pagliares
Universidade Federal de Alfenas

Alfenas, 27 de Março de 2013.

RESUMO

O grande avanço dos computadores e da Internet tem proporcionado a integração em tempo real de várias aplicações. Este trabalho apresenta um sistema que permite consultas de forma interativa usando técnicas de geolocalização. O sistema foi construído para a plataforma Google Android e possibilita aos seus usuários consultar tudo o que está agendado em determinado dia e horário no prédio selecionado da Universidade Federal de Alfenas - UNIFAL-MG. Ressalta-se que o mesmo é capaz de detectar o prédio através de sua coordenada juntamente com a coordenada do usuário, e assim ativar a interface do sistema na tela do celular.

Palavras-Chave: Android, Geolocalização, Consulta de eventos.

ABSTRACT

The breakthrough of computers and the Internet has provided real-time integration of several applications. This work presents a system that allows queries interactively using geolocation techniques. The system was built for the Google Android platform and enables its users to see everything that is scheduled on a certain day and time selected in the building of the Federal University of Alfenas - UNIFAL-MG. It is noteworthy that it is able to detect the building through its coordinated together with the user's coordinates, and thus enable the system interface on the mobile screen.

Keywords: Android, Geolocation, Consultation Event

LISTA DE FIGURAS

FIGURA 1 - Arquitetura do sistema para consulta de eventos via geoposicionamento	14
FIGURA 2 - Trecho de código de uma classe Java marcada como entidade.....	15
FIGURA 3 - Ilustração do módulo web	19
FIGURA 4 - Arquitetura do módulo cliente e suas camadas	21
FIGURA 5 - Ilustração do módulo do cliente.	22

LISTA DE ABREVIACÕES

GPS Global Position System

UNIFAL Universidade Federal de Alfenas

JSF Java Server Faces

XML Extensible Markup Language

EJB Enterprise Java Beans

JPA Java Persistence API

REST Representation state transfer

URI Uniform Resource Identifier

API Application Programming Interface

POJO Plain Old Java Objects PDV

CRUD Create, Read, Update and Delete

SQL Structured Query Language

SUMÁRIO

1 Introdução.....	9
1.1 Justificativa e Motivação	10
1.2 Problematização	10
1.3 Objetivos	11
1.3.1 Gerais	11
1.3.2 Específicos	11
1.4 Metodologia	12
2 Referencial Teórico	13
2.1 Software para consulta de eventos via geoposicionamento.....	13
2.2 Camada de persistência	14
2.2.1 MySQL	16
2.3 Camada de negócio.....	16
2.4 Camada de serviço	17
2.5 O módulo web.....	18
2.5.1 GlassFish 3.1	19
2.6 O Módulo móbil.....	20
2.6.1 Atividades e Fragmentos	22
2.6.2 Controladores	23
2.6.3 Camada de Acesso Web Services	23
3 Resultados	24
4 Conclusões e Trabalhos Futuros	25
REFERÊNCIAS	26

1

Introdução

Este capítulo apresenta alguns detalhes pela razão o qual este trabalho foi desenvolvido, bem como seus objetivos, justificativa e motivação para a realização deste.

Atualmente, mais do que em qualquer outra época, a computação está sujeita à profundas mudanças devido a criação de novas tecnologias e à adaptação de novas realidades (KAMIMURA et al, 2012).

O avanço dos computadores e da Internet colaborou para a integração em tempo real de várias aplicações, como vídeos, jogos e ambientes interativos por meio da transferência eficiente de dados e informações.

Dada a disponibilidade de tais recursos, sistemas incorporados ao GPS (*Global Position System*) são cada vez mais comuns e de boa utilidade. O GPS foi colocado em produção em 1991 e no ano de 1993 a rede de satélites formadora do sistema finalmente foi concluída.

O desenho da constelação dos satélites permite que pelo menos quatro satélites estejam sobre o plano normal do observador (BLITZKOW, 1995).

Neste sentido, vale ressaltar que o GPS funciona da seguinte maneira: os satélites possuem relógios altamente precisos, que marcam a hora na casa dos nanosegundos com exatidão. Assim, quando o sinal é emitido até o observador, o satélite envia também o registro do momento em que o sinal foi disparado.

Calculando a diferença do tempo que ele chegou ao observador perante o horário de emissão, consegue-se calcular a distância do usuário do sistema diante do satélite. Multiplicando essa distância pela velocidade do sinal torna-se possível encontrar a coordenada requerida (MONICO, 2000).

Esta tecnologia associada a diferentes dispositivos tem causado grande mudança na vida das pessoas, proporcionando facilidades e contribuindo de diferentes maneiras para obtenção de informações através da captura de sua posição no globo terrestre, seja

para a produção de algum dado comparativo ou também para o avanço da área acadêmica.

1.1 Justificativa e Motivação

Desde a viabilização dos primeiros receptores de sinais georreferentes no mercado, tem ocorrido um grande aumento no número de aplicações dependentes de dados coletados via levantamento topográfico (BERNARDI; LANDIM, 2002).

Propriedades como, alta precisão na localização de coordenadas e uma boa interatividade, tornam o sistema de posicionamento global uma alternativa muito interessante para ser integrada em novos projetos. Assim sendo, propõem-se uma aplicação que servirá de complemento para o sistema de busca já existente na instituição.

O aplicativo proposto permitirá que portadores de um dispositivo móvel com suporte ao Sistema Operacional Android tenham acesso a todos os eventos realizados em tempo real nos prédios da Universidade de forma rápida e interativa.

1.2 Problematização

Atualmente o Porteiro Web é o programa em uso no campus da Universidade Federal de Alfenas/UNIFAL-MG para a realização de consultas sobre horários de eventos, como aulas e exposições. Nota-se, porém, que este programa é pouco divulgado entre os alunos e ainda não possui o cadastro de cômodos, como sala dos professores, cantina entre outras coisas.

Partindo-se da base de dados que o Porteiro Web possui hoje em dia e pensando no grande potencial oferecido por novas tecnologias em ascensão, como a de aplicativos para dispositivos móveis, surge a ideia de criação de um sistema complementar ou independente que seja mais viável, dinâmico e moderno que o adotado atualmente.

1.3 Objetivos

Com o grande avanço da computação, e de outras tecnologias em geral, pode-se notar que o uso do Sistema de Posicionamento Global está cada vez mais presente na vida das pessoas, sendo através de software munidos de alguma tarefa que exige o posicionamento do usuário no globo terrestre ou até mesmo por meio do aparelho GPS utilizado como guia em nosso dia-a-dia.

1.3.1 Gerais

Objetivou-se com este trabalho, permitir que pessoas portadoras de dispositivos móveis com suporte à tecnologia Android, pudessem obter informações sobre serviços prestados nos prédios da UNIFAL-MG, tais como aulas ministradas, reuniões agendadas, dentre outras diversas informações.

1.3.2 Específicos

Para se conseguir sucesso nos objetivos gerais do projeto proposto, foi necessário atender a uma lista específica de objetivos:

- Criar a visão do projeto.
- Estudar sobre GPS e localização em qualquer parte do globo terrestre.
- Estudar Android, Java Server Faces, PrimeFaces, JPA, Web Services RESTful e SOAP.
- Selecionar um prédio da UNIFAL para os testes.
- Implementar um módulo web com informações sobre o prédio escolhido para este trabalho.
- Implementar um módulo mobile que se comunique com o módulo web via web services.
- Realizar testes na aplicação desenvolvida.

1.4 Metodologia

A metodologia de pesquisa que tornou os objetivos deste projeto possíveis foi a pesquisa-ação.

Segundo Kincheloe (1997), a pesquisa-ação, rejeita as noções positivistas de racionalidade, pressupondo a exposição entre valores pessoais e práticos. Isso ocorre em parte porque a pesquisa-ação não pretende apenas entender ou mostrar o mundo da prática, mas sim transformá-la.

Foi criado o documento visão do projeto, para tornar mais claro os primeiros requisitos, restrições e riscos, possibilitando o planejamento das ações para o começo da pesquisa de tecnologias.

O prédio V da UNIFAL-MG foi escolhido para o início do desenvolvimento do projeto. Após a escolha do prédio, iniciou-se a implementação do sistema. A princípio foi criado um Módulo Web que serviu para os cadastros de novos prédios e manutenção do sistema. Outra camada, Mobile também foi desenvolvida, com a função de exibir os dados a serem consultados, sendo sua funcionalidade restrita primeiramente à plataforma Mobile Android.

Um meio de comunicação entre as demais plataformas já mencionadas precisou ser criado, para que uma passasse informações para a outra via internet.

Por fim, foram realizados testes no sistema para que o mesmo pudesse ser colocado em produção na Universidade Federal de Alfenas.

2

Referencial Teórico

Este capítulo apresenta uma revisão dos conceitos necessários para um bom entendimento deste trabalho.

2.1 Software para consulta de eventos via geoposicionamento

A ideia principal para a criação do presente sistema é atacar a interatividade e a mobilidade, colocando em prática tecnologias modernas como a geolocalização e a plataforma Android.

Pensando nisto, foi proposta uma aplicação "mobile" destinada à plataforma Android, com o objetivo de consultar todos os eventos a serem realizados em qualquer prédio da Universidade Federal de Alfenas - UNIFAL via localização por GPS.

Para a identificação do prédio será estipulada um raio de ativação, definido usando a coordenada do interior do edifício juntamente com a localização do cliente.

Dividiu-se o projeto em dois módulos principais. O Módulo Web foi responsável pelo cadastro dos prédios a serem consultados e suas demais informações, como salas de aula, sala dos professores, banheiros, escadas, elevadores, dentre outras coisas. Já o Módulo Mobile se responsabilizou pela integração entre aplicação e usuário, onde o mesmo pôde acessar o menu de consultas de eventos, quando se encontrava dentro da região de identificação.

O Módulo Web foi subdividido em três pequenas camadas, sendo elas: a camada de negócio, a camada de persistência e a camada de serviço. A camada de negócio foi responsável por incorporar toda lógica do sistema. A camada de persistência se encarregou de tudo que dizia respeito à persistência de dados, implementando métodos para remoção, atualização, consulta e cadastro. Já a camada de serviço fez a distribuição das operações entre os módulos Web e Móvel como mostrado na Figura 1. Cabe ressaltar que tais módulos possuem uma única URI do tipo "*http://localhost/rest/exemplo*" sendo definida no Web Service RESTful. Os módulos Web e Mobile contêm recursos REST para executar as operações. Como servidor de

aplicação foi utilizado o glassfish 3.1 e como sistema gerenciador de banco de dados o MySQL.

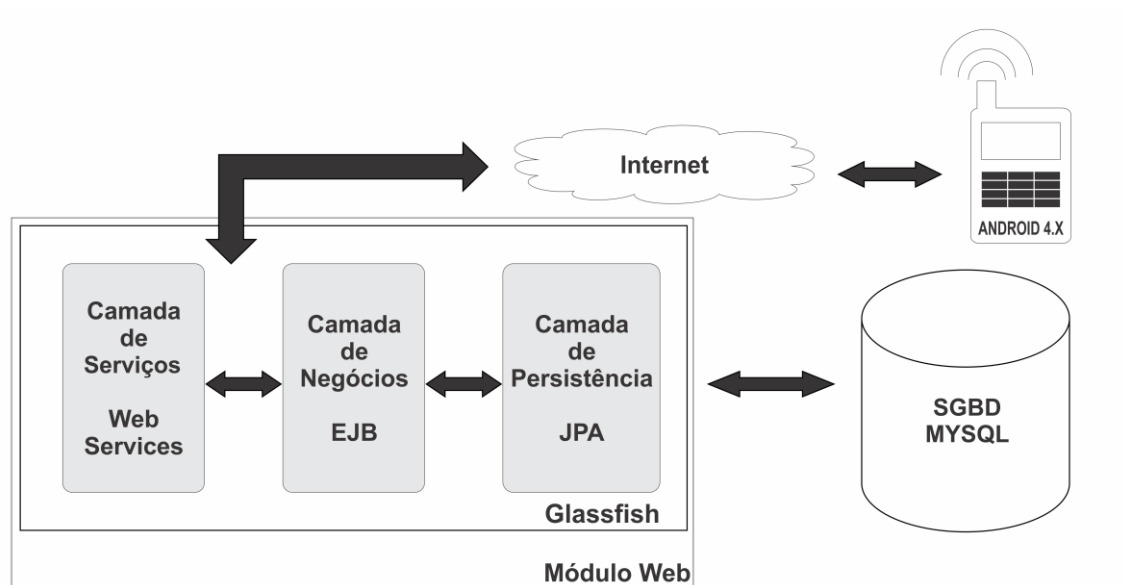


Figura 1 - Arquitetura do sistema para consulta de eventos via geoposicionamento.

2.2 Camada de persistência

A persistência de dados é um assunto sempre em destaque e se constitui em uma das partes mais importantes de um sistema.

A JPA (*Java Persistence API*) é uma interface de programação aplicada ao padrão Java usado no controle da persistência de dados. Com esta API é possível obter queries (buscas) dinâmicas baseadas em um objeto exemplo. A API adiciona as queries construindo um tipo de grafo, conforme os atributos dos respectivos objetos encontrados (TELLIER; BONIN, 2011).

A JPA é um modelo definido para *frameworks* com mapeamento objeto/relacional, que contribui para alterar a estrutura de seu código sem precisar ter qualquer tipo de preocupação em adaptar as entidades. Um *framework* nada mais é do que um arcabouço em desenvolvimento de software, capaz de unir códigos comuns entre vários projetos (TELLIER; BONIN, 2011).

A Entidade representa a tabela e seus atributos representam as colunas. Para esta ocasião usamos a anotação `@Entity`, responsável por marcar uma determinada classe com tal característica.

Existem muitas outras anotações como `@id` que marca um atributo como chave primária da entidade e `@mappedSuperClass` que se responsabiliza por gerar a herança entre as classes de entidade. Anotações funcionam como metadados, isto é, como um forma de se passar informações a um compilador ou a um framework.

Além do mais, esta API oferece diferentes tipos de relacionamentos entre as entidades, também representadas por anotações como `@OneToOne`, `@ManyToOne`, `@OneToMany` e `@ManyToMany`. Tais anotações possuem parâmetros muito importantes que são: o `cascade` e o `fetch`, responsáveis por passar instruções na hora de se alterar os dados persistidos através de alguma operação de inserção por exemplo.

Ao optar-se pela JPA, economizou-se: esforço de desenvolvimento, com a construção de queries para o CRUD (*criar, remover, atualizar, deletar*) e preocupações comuns ao acesso ao banco de dados, uma vez que grande parte de seu código é gerado automaticamente pelo IDE Netbeans.

A Figura 2 apresenta um trecho da classe marcada como uma entidade. Na linha 01, a anotação `@Entity` define a classe como entidade a ser mapeada no banco de dados. Na linha 02, `@NamedQueries` abre o escopo para declaração das `@NamedQuery` que definem consultas declaradas estaticamente.

Outras anotações como `@id` na linha 07 marca o primeiro atributo como chave primária e `@GeneratedValue` na linha 08 define a forma de geração da chave.

`@ManyToOne` na linha 18 define um relacionamento de muitos para um, com a classe de entidade `Prédio` e `@JoinColumn` define a chave estrangeira.

```

01.  @Entity
02.  @NamedQueries({
03.      @NamedQuery(name = "Andar.predios", query = "select a from Andar a
04.                  where a.idPredio.idPredio = :Predio"))
05.  public class Andar implements Serializable {
06.      private static final long serialVersionUID = 1L;
07.      @Id
08.      @GeneratedValue(strategy = GenerationType.IDENTITY)
09.      private Integer idAndar;
10.      private String informacoes;
11.      private Integer nome;
12.      private String tipo;
13.      @JoinColumn(name = "id_predio", referencedColumnName = "id_predio")
14.      @ManyToOne
15.      private Predio idPredio;

```

Figura 2 - Trecho de código de uma classe Java marcada como entidade.

2.2.1 MySQL

O MySQL é um sistema gerenciador de banco de dados, que utiliza a linguagem SQL. Trata-se de um software cujo objetivo principal é gerenciar as entidades criadas pela JPA.

Suporta vários recursos, que o fazem se destacar como um dos melhores SGBD's (Sistema Gerenciador de Banco de Dados) disponíveis atualmente, além de ser um software livre. (TECH REPUBLIC, 2005).

Devido a tais peculiaridades, o SGBD MySQL foi escolhido para constituir a camada de recursos da aplicação desenvolvida.

2.3 Camada de negócios

Para concentrar as operações principais do sistema faz-se necessária a existência de uma camada de negócios na qual se localizará os EJB's (*Enterprise JavaBeans*) da aplicação. Este tipo de componente é utilizado para monitorar serviços que representam as regras de negócio.

A tecnologia Enterprise JavaBeans (EJB) é a arquitetura existente no servidor para a plataforma Java, que permite um desenvolvimento rápido e bem simplificado entre sistemas distribuídos, transacionais e portáteis.

Um EJB constitui-se basicamente em um componente gerenciado que é criado, e distribuído pelo container gerenciador do Java. Esse controle dá ao container o poder de manipular o número de EJBs existentes e recuperá-los, para o uso com o banco de dados ou para uma memória de conexão. Um container nada mais é, que um objeto que comporta outros objetos e podem ser adicionados ou excluídos dinamicamente em tempo de execução. (KEITH et al, 2006)

A vantagem de se utilizar EJB's consiste no grande potencial que os mesmos possuem em centralizar os serviços oferecidos de um container, como por exemplo, a segurança ao código. Para que isso seja possível, o EJB deve estar corretamente implantado ao sistema, com regras que permitam identificá-lo como um EJB e conseqüentemente, usufruir de seus benefícios (TELLIER; BONIN, 2011). Existem três tipos de EJB's, que possuem configurações distintas para diferentes propósitos:

- Session EJB - Uma sessão possui a função de mapear processos. Os EJB's devem representar a ideia de sua funcionalidade.
- Entity EJB - Responsável por mapear uma combinação de dados e métodos associados aos mesmos. Geralmente são utilizados no conhecimento de dados-objetos que resultam em algum processo de um banco de dados.
- Message-driven EJB - Possui um conceito muito parecido com o Session EJB, porém é apenas ativado quando chegam mensagens assíncronas.

Portanto, pode-se afirmar que o uso de EJB's, contribui para um código mais seguro e confiável, visto que o mesmo padroniza objetos Java de forma simples, conferindo-lhes escalabilidade e um ciclo de vida gerenciável.

2.4 Camada de serviços

A camada de serviços é responsável por distribuir as operações entre os módulos web e mobile, sendo realizada no *Web Service*. Cabe lembrar que o *Web Service* é um tipo de sistema web, oferecido por meio do protocolo HTTP (*Hypertext Transfer Protocol*), que permite uma aplicação distribuída, onde os seus componentes podem ser executados em programas distintos (TELLIER; BONIN, 2011).

Esta conexão que o *Web Service* cria entre módulos heterogêneos é feita através de arquivos XML, que representam as mensagens a serem enviadas de um módulo para o outro.

Um *Web Service* pode possuir dois paradigmas diferentes: estilo SOAP e REST. O primeiro se caracteriza por ser uma especificação de protocolos para troca de dados através de pacotes de mensagem entre dois pontos de extremidade. Já o segundo é mais um estilo de arquitetura utilizado para criar aplicativos cliente-servidor.

Na prática, o recurso REST (*Representational State Transfer*) gera URIs (*Uniform Resource Identifier*) e funciona como um identificador capaz de satisfazer os requisitos necessários para a implementação do recurso.

Ressalta-se que um Recurso se constitui em um item que possui hyperlinks, ou seja, que possui o caminho para algum outro item, e esses caminhos são as URIs (KALIN, 2010).

Como todas as outras tecnologias já mencionadas, o *Web Service* também faz uso de anotações. A anotação `@webServicesProvider` por exemplo sinaliza que as mensagens trocadas serão documentos XML de algum tipo (KALIN, 2010).

As vantagens de se usar um serviço web são claras. Sistemas modernos são escritos em diferentes linguagens hoje em dia, e essas aplicações podem ficar em uso por muito tempo, tornando-se sistemas legados, como por exemplo, sistemas bancários. Como toda aplicação exige, sempre será necessária a introdução de novas funções com o passar do tempo, o que pode ser uma tarefa bem dispendiosa se tratando de grandes programas. O *Web Service* resolve esse problema, por ser imune a qualquer tipo de linguagem e plataforma, uma vez que, se constitui em sistemas distribuídos que se comunicam através do protocolo HTTP.

2.5 O módulo web

O módulo web foi responsável pelo cadastro dos prédios pertencentes à Universidade Federal de Alfenas - UNIFAL-MG, para que cada um pudesse apontar os eventos a serem realizados na data consultada. Compôs-se pelas camadas já mencionadas anteriormente, ou seja, camada de persistência, camada de negócio e camada de serviços. O Módulo Web foi desenvolvido em Java Server Faces (JSF), por se tratar do *framework* mais utilizado para o desenvolvimento de aplicações web com Java. O mesmo possibilita criar sistemas de forma mais rápida e ágil, permitindo uma fácil manutenção e integração com outras tecnologias.

Além de todas essas facilidades, o JSF ainda possui algumas ferramentas para o auxílio de criação de telas, das quais se destacam: o PrimeFaces e o RichFaces (GATTI; PEREIRA, 2011).

O RichFaces é uma biblioteca para sistemas web que utiliza o JSF, possuindo um bom suporte ao AJAX e temas para padronização de interfaces sem necessitar de nenhuma linha de Java Script.

O PrimeFaces, assim como o RichFaces, também oferece um conjunto de funcionalidades para sistemas web. Este framework possui muitos componentes e os disponibiliza na forma de um showcase com todas as estruturas e códigos à vista (MOTA; WERNER, 2011).

As vantagens de se utilizar o PrimeFaces foi o que determinou sua escolha para integrar o projeto, entre elas estão:

- Excelente conjunto de componentes HTML e gráficos.
- Ajax API baseado no padrão JSF 2.
- Excelente documentação.
- 30 temas incorporados, e suporte a ferramentas de designer.

A Figura 3 abaixo ilustra o módulo web em sua forma completa, com a tabela de horários no centro, botão de logout no canto superior direito e o menu localizado na parte inferior da página.

Segunda	Terça	Quarta	Quinta	Seita	Sala	Horário	
Ecologia	Embriologia Basica	UAN II (2)	OP UNIT	Parasito	Sala V202 Andar: 2 Prédio: Prédio V	09:00 - 11:00	Visualizar - Editar - Deletar
Quim Analtit	Toxico	Fisio	TCC I	Fisio	Sala V202 Andar: 2 Prédio: Prédio V	13:00 - 15:00	Visualizar - Editar - Deletar
Farmaco	Evolucao		Fisio AN	Nut Social I	Sala V202 Andar: 2 Prédio: Prédio V	15:00 - 17:00	Visualizar - Editar - Deletar
Fisiologia	Bromato	UAN II (3)	Farmaco I	Q ORG	Sala V202 Andar: 2 Prédio: Prédio V	07:00 - 09:00	Visualizar - Editar - Deletar

Cadastrar novo Horário

Figura 3 - Ilustração do módulo web.

2.5.1 Glassfish 3.1

O GlassFish 3.1 é um servidor de aplicações que implementa todos os padrões Java EE 6. Um servidor de aplicações é um sistema que oferece alguns serviços comuns, considerados serviços de infra-estrutura, como por exemplo transformar as

requisições HTTP em objetos Java, conferir segurança de autenticação, entre outros (LLOPART et al, 2007).

Diante de tais capacidades, o mesmo foi escolhido como o servidor de aplicação para rodar o sistema.

2.6 O módulo mobile

Pesquisas recentes mostram que mais de três bilhões de pessoas hoje no mundo possuem um aparelho celular. Estes dados deixam mais do que evidente como o forte mercado de dispositivos mobile está em ascensão (KAMIMURA et al, 2012).

Com tamanho crescimento, a disputa pelo mercado se torna cada vez mais acirrada (CRUZ, 2011).

Tendo em vista que a disputa é mais pela aparelhagem do que pelos software que embarcam os aparelhos, surge a necessidade de desenvolver um sistema operacional mobile de código aberto que seja comum à maioria dos aparelhos.

A ideia é tornar as aplicações desenvolvidas por terceiros comuns a todos os dispositivos que possuam o Sistema Operacional móvel. Embasando-se nisso, surge o Android.

O Android é uma plataforma de desenvolvimento para dispositivos móveis, baseado no sistema operacional Linux e que vem cada vez mais se popularizando no mundo da aparelhagem móvel (CRUZ, 2011).

É essencial ressaltar que o Android apresenta componentes atraentes para o desenvolvimento de softwares em dispositivos móveis, além de implementar conceitos eficientes de integração e flexibilidade (TELLIER; BONIN, 2011).

Pode-se afirmar que o sistema operacional móvel Android foi selecionado para incorporar o módulo do cliente, devido às características que o mesmo possui e também pelo seu menu interativo e de fácil manuseio.

O módulo mobile da aplicação de consulta de eventos via localização global consiste apenas em uma tela de interface que será ativada conforme a localização do usuário for aceita pelo sistema. Tal aceitação será feita por meio de coordenadas via geoposicionamento do usuário e do prédio.

Para melhorar o entendimento, será feita uma breve descrição das sub-divisões das camadas do módulo cliente, conforme mostrado na Figura 4. A primeira camada é

representada pelas atividades e fragmentos, já a segunda se refere aos controladores e a terceira tem como responsabilidade permitir o acesso remoto ao web service, ambas serão descritas em maiores detalhes nas próximas seções deste trabalho.

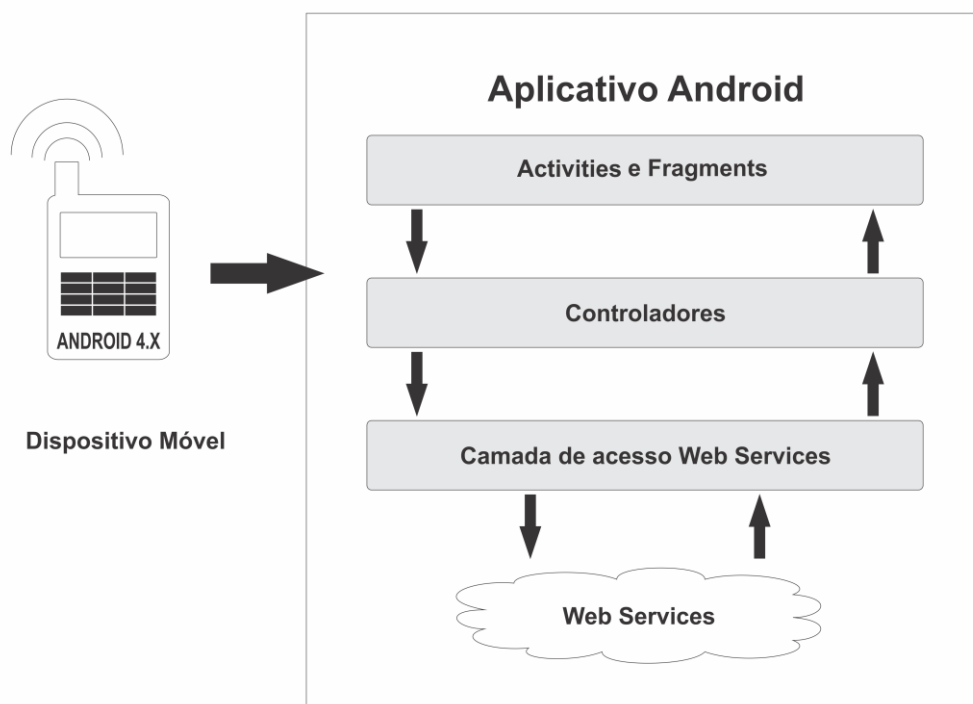


Figura 4 - *Arquitetura do módulo cliente e suas camadas.*

Na Figura 5, são demonstradas duas telas presentes no módulo do cliente. Na tela da esquerda é exibido o menu de andares a serem selecionados na parte superior da interface gráfica.

Na tela da direita é evidenciado o mapa do segundo andar do prédio V, onde cada container representa um botão que resultará na descrição da sala com opções para seus horários.

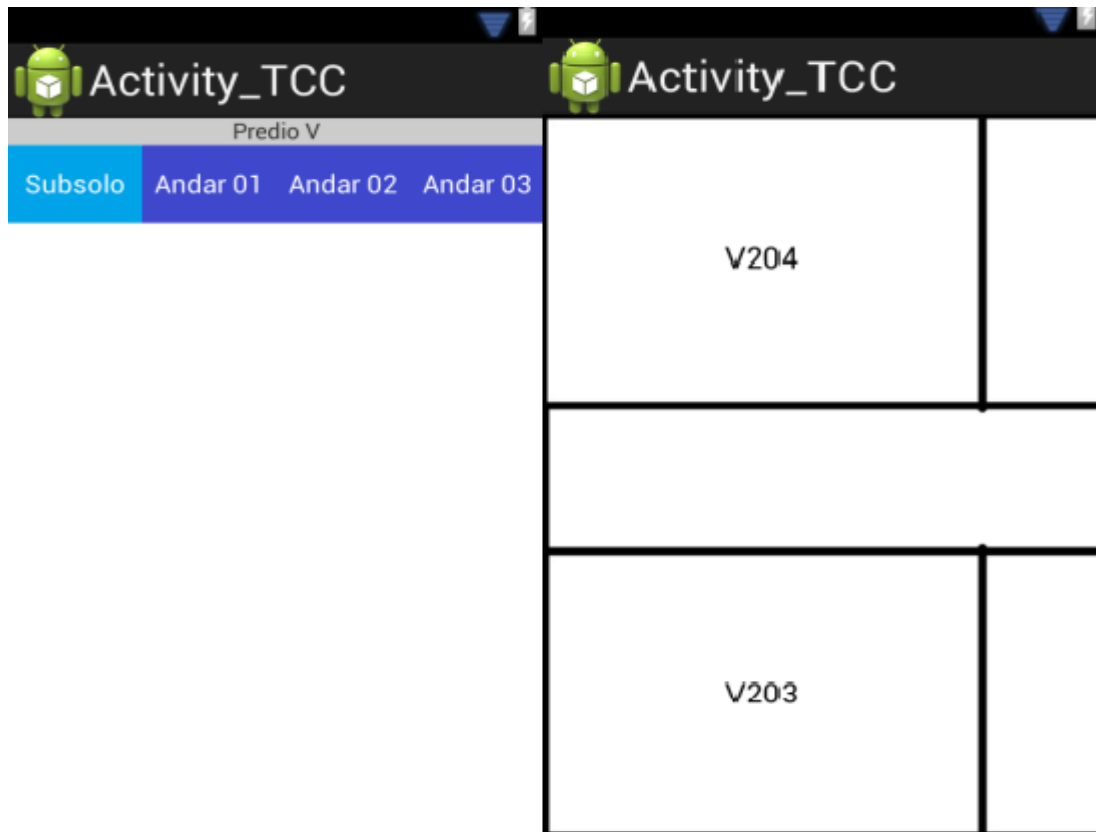


Figura 5 - Ilustração do modulo do cliente.

2.6.1 Atividades e Fragmentos

Uma atividade constitui-se em um componente que nos fornece uma tela, servindo de veículo para a integração do usuário com o sistema (LECHETA, 2008).

Uma aplicação geralmente possui uma quantidade razoável de atividades, que estão ligadas entre si representando uma grande função a ser executada ou um conjunto destas.

Os fragmentos denotam o comportamento da interface com o usuário sobre uma atividade. Na maioria das vezes, combinam-se vários fragmentos a cada atividade e assim se constrói uma tela multi-painel (LECHETA, 2011).

O ciclo de vida de um fragmento é altamente dependente do ciclo de vida da atividade que o hospeda. Por exemplo, quando uma atividade está pausada, os fragmentos que nela estão também serão pausados (DEVELOPERS, 2013).

Vale destacar aqui o funcionamento do sistema de geolocalização que se situa nestas camadas. Conforme já mencionado, o programa exibirá o prédio para consulta quando o mesmo se encontrar dentro de uma determinada área de ativação.

O que ocorre é o seguinte: através do sistema GPS do celular e de uma coordenada coletada no interior do edifício, é estipulado um raio, chamado "área de ativação". Desta forma, se o cliente estiver dentro deste espaço o sistema exibirá a lista de prédios detectados para que seja realizada a determinada consulta.

Mas como é estipulado e calculado este espaço?

Por meio de uma coordenada geográfica capturada no interior de algum prédio, é estipulada uma distancia fixa, definida no próprio código como vinte metros, e assim calculado um perímetro.

A validação para a consulta é feita usando-se a coordenada do usuário, obtida em tempo real, através do próprio Android e feita a subtração com a coordenada do edifício presente na base de dados. Se o módulo da subtração for menor ou igual à distância fixa estipulada, os devidos itens serão listados.

2.6.2 Controladores

Um controlador é o primeiro objeto além da camada de interface gráfica com o usuário que é responsável por receber ou tratar uma mensagem de operação do sistema e encaminhá-la para a lógica de negócios da aplicação (LARMAN, 2002).

No módulo cliente estes serão os responsáveis pela comunicação direta com a aplicação fornecedora do web service, encapsulando e fornecendo todas as chamadas de métodos obtidos (LECHETA, 2008).

2.6.3 Camada de acesso web service

Esta camada é a responsável por realizar a requisição assíncrona através do *loader*, que nada mais é que um carregador.

As mensagens podem ser requisitadas de duas formas diferentes, como XML ou JSON, constituindo-se em um formato especial para intercâmbio de mensagens.

O *loader* agiliza o processo, tornando mais fácil o carregamento dos dados, e entre suas facilidades estão: (DEVELOPERS, 2013):

- Encontram-se disponíveis para cada atividade e fragmento.
- Fornecem o carregamento assíncrono de dados.
- Monitoram as fontes de seus dados.

3

Resultados

Este capítulo apresenta os resultados relacionados á este trabalho.

Após o estudo e definição da arquitetura do sistema e entendimento das tecnologias propostas para o desenvolvimento, foi possível construir uma aplicação mobile, capaz capturar os eventos do prédio V da UNIFAL-MG através dos recursos de geolocalização.

Analisando o sistema como um todo, pode-se notar que referente à sincronia e a taxa de acerto ele se encontra bem fiel, se tratando da conectividade a mesma esta sendo realizada de forma correta e satisfatória.

Apesar da aplicação não estar disponível para todos os prédios, fica evidente a gama de opções e ideias para que se possa evoluir ainda mais o sistema, e habilitá-lo para funções ainda mais surpreendentes.

4

Conclusões e Trabalhos Futuros

Este capítulo apresenta as conclusões relacionadas á este trabalho.

O trabalho aqui exposto teve o intuito de apresentar os passos para a criação de uma aplicação mobile integrada a um módulo web de administração.

O projeto em produção proporcionou um alto ganho na interatividade para realização de consultas, além de colocar em prática tecnologias em alta no mercado.

Entretanto vale ressaltar que o mesmo possui algumas limitações como, por exemplo, a integração com o Porteiro Web, site para reservas de salas e consulta de horários da universidade.

Pensando nas suas limitações e em possíveis trabalhos futuros propõe-se:

- Integrar a aplicação desenvolvida com o porteiro web;
- Ampliar o sistema para os edifícios restantes da UNIFAL-MG;
- Criar uma versão para o iOS (Sistema operacional móvel da apple).

Tais requisitos irão conferir ao sistema novas funcionalidades e versões ainda mais abrangentes.

REFERÊNCIAS

BLITZKOW, D. **Navstar/GPS: um desafio tornado realidade**. In: SIMPÓSIO BRASILEIRO DE GEOPROCESSAMENTO, 3, 1995, São Paulo. Anais. São Paulo, 1995

CRUZ, B. et. al. **Desenvolvimento de uma aplicação embarcada em celular visando controle de robô via Wi-Fi**. 2010. Disponível em:

<http://www.upf.br/seer/index.php/rbca/article/view/1301/1071>

Acessado: 14 de Novembro de 2012.

DEVELOPERS. **API guias**. 2013. Disponível em:

<http://developer.android.com/guide/components/index.html>. Acessado: 13 de

março de 2013.

GATTI, M.; PEREIRA, J. **Richfaces & Primefaces utilizados no JSF 2.0**. 2011.

Disponível em: <http://web.unipar.br/~seinpar/artigos/Marcos-Gatti.pdf>

Acessado: 09 de Novembro de 2012.

KALIN, M. **Java Web Services implementando**. Alta Books, Rio de Janeiro, Brasil 2010

KAMIMURA, Q. P. et al. Inovação tecnológica: início de novos negócios em TI - estudo de caso numa empresa de bens de capital do Vale do Paraíba. **Latin American Journal Of Business Manegement**, v.3, n.2, p. 174-187, 2012.

KEITH, M.; HALEY, J.; SCHINCARIOL, M. **Pro EJB 3.0 Persistence**. Press, Nova York, Estados Unidos da América 2006. Disponível em:

<http://books.google.com.br/books?hl=pt>

[BR&lr=&id=fVCuB_Xq3pAC&oi=fnd&pg=PR1&dq=EJB+3.0&ots=Bh7w7h-F_Z&sig=HHuxX9ANJeysWIXrek8XvpKaEpM#v=onepage&q=EJB%203.0&f=false](http://books.google.com.br/books?hl=pt&lr=&id=fVCuB_Xq3pAC&oi=fnd&pg=PR1&dq=EJB+3.0&ots=Bh7w7h-F_Z&sig=HHuxX9ANJeysWIXrek8XvpKaEpM#v=onepage&q=EJB%203.0&f=false).

Acessado: 07 de Novembro de 2012.

KINCHELOE, J. L. **A formação do professor como compromisso político: mapeando o pós-moderno**. Porto Alegre: Artes Médicas, 1997. Acessado: 16 de Outubro de 2012.

LARMAN, C. Applying UML and patterns: an introduction to object oriented analysis and design and the unified process. **Prentice Hall**, 2002.

LECHETA, R.R. Google Android para Tablets. **NovaTec**, 2011.

LECHETA, R.R. Google Android. **NovaTec**, 2º Ed, 2008.

LLOPART, E.; YOSHIDA, Y.; POUCHKIN, A. **The GlassFish Community Delivering a Java EE Application Server**. 2007. Disponível em:

<http://glassfish.java.net/faq/v2/GlassFishOverview.pdf>

Acessado: 10 de Novembro de 2012.

MONICO, J.F.G. **Posicionamento pelo NAVSTAR-GPS Descrição, fundamentos e aplicações**. 1. ed. Presidente Prudente: Editora UNESP, 2000.

MOTA, R.; WERNER, C. **Utilizando framework Java Server Faces(JSF), Hibernate e PrimeFaces para desenvolvimento de software para web**. 2011. Disponível em:

<http://web.unipar.br/~seinpar/artigos/Roberto-Dias-Mota.pdf>. Acessado: 09 de

Novembro de 2012.

RIPPER, M.D. **Universalização do Acesso aos Serviços de Telecomunicações: o Desafio Atual no Brasil**. Ciclo de Seminários Brasil em desenvolvimento, 2003.

TELLIER, C.; BONIN, O. Uma aplicação completa passo a passo integrando Android + javaEE + JavaSE. **Java de ponta a ponta**, São Paulo, Ed. 48, p. 08-23, Jul/Ago. 2011.

TECH REPUBLIC. **MySQL - Administrator's Guide**. Sams Publishing, 2005.