

**DIEGO D'LEON NUNES
DIÓGENES APARECIDO REZENDE**

APLICATIVO PARA CONSULTA DE NOTAS

**UNIVERSIDADE DO VALE DO SAPUCAÍ
POUSO ALEGRE – MG**

2015

SUMÁRIO

1	QUADRO METODOLÓGICO	2
1.1	Tipo de pesquisa	2
1.2	Contexto de pesquisa	2
1.3	Instrumentos	3
1.4	Procedimentos	4
1.4.1	Uml	4
1.4.2	Resto	4
REFERÊNCIAS.....		9

1 QUADRO METODOLÓGICO

Nesse capítulo serão apresentados os métodos adotados para se realizar a pesquisa, tais como tipo de pesquisa, contexto, procedimentos, entre outros.

1.1 Tipo de pesquisa

Uma pesquisa é ato de buscar e procurar pela resposta de algo. Marconi e Lakatos (2002, p. 15) definem pesquisa como “uma indagação minuciosa ou exame crítico e exaustivo na procura de fatos e princípios”.

Existem diversos tipos de pesquisa, no entanto percebeu-se que para o propósito desta, a mais indicada foi a pesquisa aplicada, pois foi desenvolvido um projeto real que poderá ser utilizado por qualquer instituição de ensino, mas que não mudará a forma com que as pessoas recebam suas informações, apenas acrescenta mais uma forma de consultá-las.

Segundo Marconi e Lakatos (2002, p. 15), uma pesquisa do tipo aplicada “Caracteriza-se por seu interesse prático, isto é, que os resultados sejam aplicados ou utilizados, imediatamente, na solução de problemas que ocorrem na realidade”.

Dessa maneira, percebeu-se que a pesquisa enquadra-se no tipo de pesquisa aplicada, pois resolveria um problema específico, e para isso foi criado uma aplicativo para dispositivos móveis que facilitará aos graduandos acessarem o sistema *web* de uma universidade.

1.2 Contexto de pesquisa

Essa pesquisa será benéfica a qualquer instituição educacional que possua um portal *online*, pois facilitará o acesso dos discentes às suas informações escolares.


Foi criado um aplicativo para dispositivos móveis, porém inicialmente apenas para a plataforma *Android*, o qual notifica os usuários quando há alguma mudança, como por exemplo, ao ser lançada uma nota.

O aluno consegue acessar o aplicativo com o mesmo *login* do sistema *web*. O utilitário acessa o *webservice* que é responsável por buscar as informações no banco de dados e apresentá-las no dispositivo móvel.

1.3 Instrumentos

Pode-se dizer que um questionário é uma forma de coletar informações através de algumas perguntas feitas a um público específico. Segundo Gunther (2003), questionário pode ser definido como um conjunto de perguntas que mede a opinião e interesse do respondente.

Foi realizado um questionário simples, que esta apresentado na figura 1, contendo quatro perguntas e enviado para *e-mails* de alguns alunos da universidade. O foco desse questionário era saber o motivo pelo qual os usuários mais acessavam o portal do aluno e se tinham alguma dificuldade em encontrar o que procuravam. Obteve-se um total de treze respostas, no qual pode-se perceber que a maioria dos entrevistados afirmam terem dificuldades para encontrar o que precisam e que o sistema não avisa quando ocorre alguma alteração. Sobre o motivo do acesso cem por cento respondeu que entram no sistema web para consultar suas notas.



Pesquisa sobre o portal do aluno

Qual é sua opinião sobre o portal do aluno?

☐ Ótimo
☐ Bom
☐ Ruim
☐ Péssimo

Qual é sua maior dificuldade para acessar o portal do aluno?

☐ Não tenho acesso a internet
☐ Demoro para encontrar o que preciso
☐ O sistema não avisa quando são lançadas as notas
☐ Outro:

A maior parte das vezes que acesso o portal do aluno é para?

☐ Ver minhas notas
☐ Ver provas agendadas
☐ Ver minhas faltas
☐ Buscar contatos dos professores
☐ Consultar financeiro
☐ Consultar material postado pelos professores
☐ Outro:

Você acha que um aplicativo para celular para acessar o portal seria?

☐ Ótimo
☐ Bom
☐ Ruim
☐ Péssimo

100% concluído

Figura 1 – Questionário Aplicado. **Fonte:**Elaborado pelos autores.

Outra forma utilizada para realizar a pesquisa foram as reuniões, ou seja, unir-se com uma ou mais pessoas em um local, físico ou remotamente para tratar algum assunto específico. Para Ferreira (1999), reunião é o ato de encontro entre algumas pessoas em um determinado local, com finalidade de tratar qualquer assunto.

Durante a pesquisa, foram realizadas reuniões entre os participantes com o objetivo de discutir o andamento das tarefas pela qual cada integrante ficou responsável. Além disso entravam em discussão, nessas reuniões, o cumprimento das metas propostas por cada participante e o estabelecimento de novas metas. Foi utilizada nessa pesquisa, referencias em livros, revistas, manuais e web sites.

1.4 Procedimentos

1.4.1 Uml

O primeiro procedimento realizado para chegar ao resultado final da pesquisa proposta, foi delinear a arquitetura do *software* através da linguagem UML, fazendo uso de alguns diagramas oferecido pela mesma, como ferramenta de apoio. A construção desses diagramas só foi possível com a instalação da ferramenta Astah.

Para o aplicativo *Android*, fez-se necessários os diagramas de classe, de caso de uso principal e de atividade.

Para a construção do *software* proposto por essa pesquisa além do levantamento de requisitos que é peculiar da construção de qualquer *software*, foram também utilizados alguns diagramas da UML. Entre os diagramas usado estão:

- Diagrama de casos de uso;
- Diagrama de sequência;
- Diagrama de Entidade e Relacionamento (ou Modelo de Entidade e Relacionamento)

1.4.2 Resto

Para iniciar o desenvolvimento do aplicativo, primeiramente fez-se necessária a instalação e configuração da plataforma *Android Studio* e *Android SDK*. Ao concluir essa tarefa,

deu-se o início ao aplicativo *Android*.

O primeiro passo foi a criação de uma *activity* que é o *main*, ou seja a qual executará a aplicação quando for iniciada. A *activity* escolhida é denominada *Navigation Drawer Layout*. Com ela é possível criar, do lado esquerdo, uma lista que funcionará como um menu, com as opções *Home*, *Notas*, *Faltas*, *Provas Agendadas* e *Sair*. Essas opções ficam escondidas e só apareceram quando clicado no canto superior esquerdo.

Sempre que a aplicação é iniciada ou quando retorna-se para a *Home*, é mostrado ao usuário uma lista de *links* úteis. Para que essa lista aparecesse foi utilizada uma *activity* chamada de *Master/ Deital Flow* que já traz consigo o *widget* de *ListView*. Essa *activity* mostrara as seguintes opções: *Univás*, *MEC*, *FIES*, *Prouni* e *Google Acadêmico*. Ao clicar em um desses itens será chamada uma atividade que mostrará os detalhes do item escolhido, nesse caso na *activity* que detalhará o item foi utilizada uma *WebView* que mostrará a página web no espaço reservado. Dessa maneira, quando for clicado na opção *Univas* será carregada o site da universidade no aplicativo.

Logo após foram criadas três novas *activities* chamadas de *Blank Activity*, as quais listarão as matérias. Nelas foram inseridas o *widget ExpandableListView*, para que quando clicado em um item listado, ele expande e mostra mais informações, com isso quando se escolhe o menu *notas*, será mostrada uma *activity* que listará todas as matérias cursadas pelo aluno e ao clicar em uma delas mostrará as notas dessa disciplina.

Foi criada uma classe chamada *BuscaDados*, ela tem por finalidade receber os dados do *webservice*, salva-los no banco de dados local do aplicativo e os entregar para as classes que implementam o *Adapter* para listar as informações.

No arquivo *AndroidManifest.xml* foi necessário alterar a opção *Android:icon*, que define qual será o ícone do aplicativo, por padrão ele apresenta o mascote do *Android*, no entanto foi definida uma imagem do logo da universidade. Foi necessário também incluir uma *tag* de *users-permission* para *web*, que obriga o usuário a permitir o uso da *internet* pelo aplicativo.

O *Android Studio* tem uma facilidade para se trabalhar com controladores de versão, nesse caso foi escolhido o *GitHub*. Nele foi criado uma pasta e compartilhada entre os participantes e por fim configurado no IDE para que cada um possa ter a versão mais atualizada do projeto.

No que diz respeito à construção do *webservice*, foi necessário a instalação e configuração de um ambiente de desenvolvimento compatível com as necessidades apresentadas pelo *software* e que foram levantadas através dos requisitos. Foi instalado o *Servlet Container Apache Tomcat* em sua versão de número 7. O *Servlet Container* foi instalado para que o *Web Service*

pudesse fornecer os serviços necessários para o consumo de dados do Aplicativo *Android*, haja vista que *Apache tomcat* faz uso amplo do protocolo HTTP¹ e da plataforma *Java* de desenvolvimento.

Para armazenar os dados gerados e/ou recebidos foi necessário fazer a instalação do Sistema Gerenciador de Banco de Dados(SGBD) *PostGreSql* na sua versão de número 9.2. Através de um levantamento de requisitos parciais e das reuniões entre os participantes foi possível construir um Diagrama de Entidade e Relacionamento, no qual ficou definido a estrutura do banco de dados da aplicação. A figura 2 mostra o Diagrama de Entidade e Relacionamento concebido para esta pesquisa.

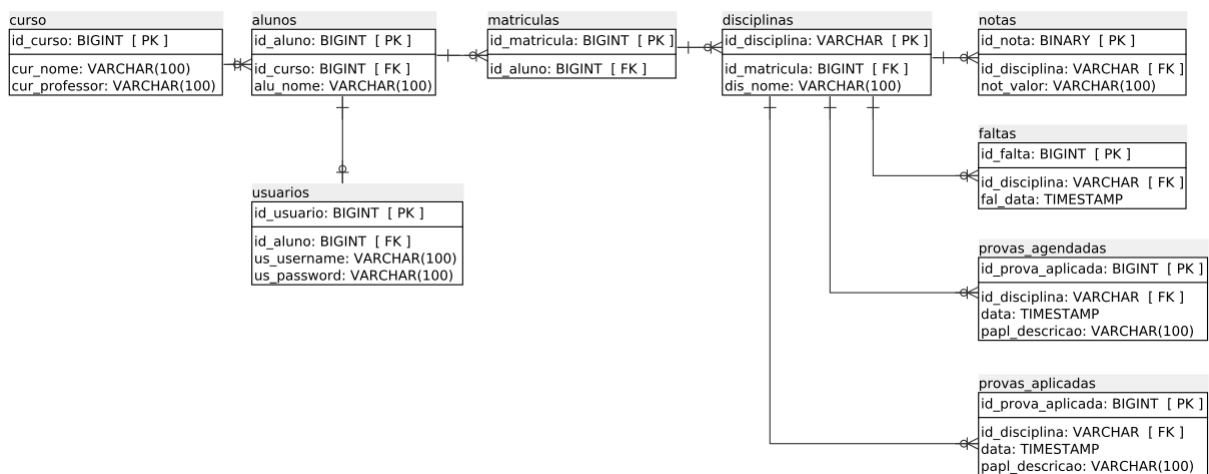


Figura 2 – Diagrama de Entidade e Relacionamento. **Fonte:**Elaborado pelos autores.

Fazendo uso desse diagrama foi possível criar todas as classes *Java* que representam as entidades do mapeamento objeto-relacional. Essas classes foram criadas fazendo uso de anotações próprias do *Hibernate*, que é um *framework* que implementa a especificação JPA². Essas classes fazem parte dos mecanismos de persistência de dados, e são simplesmente POJO's³ ou seja objetos simples que contêm somente atributos privados e os métodos *getters* e *setters* que servem apenas para encapsular estes atributos. Uma das classes criadas, foi a classe *Curso.java* que representa a tabela *cursos* no banco de dados e está representada na figura 1.1.

¹ HTTP - Hypertext Transfer Protocol


² JPA - *Java Persistence API*

³ POJO - *Plain Old Java Object*

Código 1.1 – Classe Curso. Fonte: Elaborado pelos autores.

```
1  public class Curso {
2
3      private Long idCurso;
4      private String nome;
5      private String professor;
6
7      @Id
8      @GeneratedValue
9      @Column(name = "id_curso")
10     public Long getIdCurso() {
11         return idCurso;
12     }
13
14     public void setIdCurso(Long idCurso) {
15         this.idCurso = idCurso;
16     }
17
18     @Column(length = 100, nullable = false)
19     public String getNome() {
20         return nome;
21     }
22
23     public void setNome(String nome) {
24         this.nome = nome;
25     }
26
27     @Column(length = 100, nullable = false)
28     public String getProfessor() {
29         return professor;
30     }
31
32     public void setProfessor(String professor) {
33         this.professor = professor;
34     }
35
36     /**
37      * hashCode e Equals
38      */
39 }
```

Foram criadas outras classes *Java* com a mesma finalidade da anterior, porém com pequenas diferenças no que diz respeito à atributos, métodos e anotações. Essas outras classes estão representadas na figura 3.



./imagens/imagem6.png

Figura 3 – Classes que Representam Entidades. Fonte:Elaborado pelos autores.

Em seguida à criação das entidades, foi necessário configurar o arquivo `persistence.xml` que fica dentro do *classpath* do projeto *Java* ou seja, dentro da mesma pasta onde estão contidas as classes do projeto. Esse arquivo é extremamente importante pois, é nele que estão todas as configurações relativas à conexão com o banco de dados, configurações referentes ao Dialeto

SQL que vai ser usado para as consultas e configurações referentes ao *persistence unit* que é o responsável direto por conversar com o banco de dados. O arquivo `persistence.xml` está exposto na

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
  xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="WsUnivas">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/wsunivas" />
      <property name="javax.persistence.jdbc.user" value="postgres" />
      <property name="javax.persistence.jdbc.password" value="2289cpm22" />
      <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver" />
      <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQLDialect" />
      <property name="hibernate.show_sql" value="true" />
      <property name="hibernate.format_sql" value="true" />
      <property name="hibernate.hbm2ddl.auto" value="update" />
    </properties>
  </persistence-unit>
</persistence>
```

criação do `JpaUtil`

criação e disponibilização do primeiro serviço

REFERÊNCIAS

FERREIRA, A. B. H. : **Novo Aurélio Século XXI**: o dicionário da língua portuguesa. 3^a. ed. Rio de Janeiro: Nova Fronteira, 1999.

GUNTHER, H. : **Como Elaborar um Questionário**. 2003. Disponível em: <http://www.dcoms.unisc.br/portal/upload/com_arquivo/como_elaborar_um_questionario.pdf>. Acesso em: 15 de Abril de 2015.

MARCONI, M. A.; LAKATOS, E. M. : **Técnicas de pesquisas**: planejamento e execução de pesquisas, amostragens e técnicas de pesquisas, elaboração, análise e interpretação de dados. 5^a. ed. São Paulo: Atlas, 2002.