

**USC – UNIVERSIDADE DO SAGRADO CORAÇÃO**

**MAURÍCIO GALLIS SAHÃO**

**PROTÓTIPO DE AUTOMATIZAÇÃO E  
INFORMATIZAÇÃO DO CONTROLE DE PRESENÇA  
DE ALUNOS COM O AUXÍLIO DA BIOMETRIA**

**BAURU**

**2011**

**MAURÍCIO GALLIS SAHÃO**

**protótipo de automatização e informatização do  
controle de presença de alunos com o auxílio da  
biometria**

Trabalho de conclusão de curso à  
Universidade Sagrado Coração, para a  
obtenção do título de Bacharel em Ciência  
da Computação, sob orientação da prof.  
Ms. Patrícia Bellin Ribeiro.

**BAURU**

**2011**

**MAURÍCIO GALLIS SAHÃO**

**PROTÓTIPO DE AUTOMATIZAÇÃO E INFORMATIZAÇÃO DO  
CONTROLE DE PRESENÇA DE ALUNOS COM O AUXÍLIO DA  
BIOMETRIA**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sónias Aplicadas como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação do Prof. Patrici Bellin Ribeiro.

Banca examinadora:

---

Prof. Ms. Patrícia Bellin Ribeiro  
Universidade Sagrado Coração

---

Prof. MS. Patrick Pedreira  
Universidade Sagrado Coração

---

João Marcelo de Mattos  
Analista de Sistemas

## RESUMO

Atualmente, é comum utilizar a biometria para o reconhecimento de pessoas, por tratar-se de uma tecnologia bastante difundida no mercado, ela permite o reconhecimento rápido e preciso de um indivíduo, podendo ser utilizada para uma infinidade de aplicações relacionadas a identificação, por isso, diversos órgãos estão utilizando a biometria, temos como exemplo o Governo Federal, Provedores de Saúde e Serviços Financeiros.

Este projeto vem com a proposta de facilitar o dia a dia do professor, informatizando a chamada escolar utilizando a biometria digital para o reconhecimento dos alunos. Ele também traz a proposta de computar as horas aulas dos alunos, considerando a carga horária, para obter a porcentagem final de presença. O projeto MyBioControl trata-se de um sistema on-line, contendo a parte cadastral, login, consultas e impressão de relatórios, o diferencial será o reconhecimento do aluno utilizando a biometria digital através de um leitor óptico. Para o desenvolvimento do projeto foi utilizado o padrão MVC (*Model View Control*); a linguagem de programação Java; o banco de dados PostgreSQL; as ferramentas (frameworks) Hibernate, PrimeFaces, FingerPrintSDK; o servidor de aplicação Tomcat.

**Palavras-chave:** Biometria, Digital, Template, Informatização, Automatização.

## **ABSTRACT**

Actually, It is common to use biometrics for recognition of people, because it is a widespread technology in the market, allows fast and accurate recognition of an individual, it can be used for a infinitude of applications related to identification by therefore, several agencies are using biometrics, we have the example of the Federal Government, Health Providers and Financial Services. This project comes with the proposal to facilitate the daily life of the teacher, computerizing the call using biometrics for recognition of digital students. It also brings the proposal to calculate classes the hours of students, working always on top of the workload to get the final percentage of presence. The MyBioControl project is an online system, containing the cadastre, login, queries and print reports, the gap will be the recognition of the student using biometrics through a digital optical reader. For the development of the project we used the standard MVC (Model View Control), the Java programming language, PostgreSQL database, tools (frameworks) Hibernate, PrimeFaces, FingerPrintSDK and the Tomcat application server.

## Sumário

<b>1 – Introdução.....</b>	<b>8</b>
1.1 - Problema de Pesquisa.....	9
1.2 – Justificativa.....	9
1.3 - Objetivo Geral .....	10
1.3.1 - Objetivos Específicos .....	10
<b>2 – Biometria.....</b>	<b>11</b>
<b>3 - Sistemas Biométricos.....</b>	<b>15</b>
<b>4 - Desenvolvimento de Software.....</b>	<b>17</b>
4.1 - UML (Unified Modelling Language).....	17
4.1.1 - Diagramas de Caso de Uso.....	17
4.1.2 – Exemplos de Diagrama de Caso de Uso.....	18
4.1.3 – Diagramas de Classes em UML.....	19
4.1.4 – Associações Simples.....	20
4.1.5 – Diagrama de Colaboração.....	22
4.2 – Sistema Gerenciador de Banco de Dados (SGDB).....	23
4.2.1 – Classificação dos Sistemas Gerenciadores de Banco de Dados.....	24
<b>5 – Materiais e Métodos.....</b>	<b>26</b>
5.1 – Linguagem de Programação.....	26
5.2 – Ambiente Integrado de Desenvolvimento (IDE).....	26
5.3 – Banco de Dados.....	27
5.4 – Ferramenta de Mapeamento Objeto/Relacional (Hibernate).....	27
5.5 – Java Server Faces 2.0 (JSF).....	28
5.6 – PrimeFaces 2.2.1.....	29
5.7 – Servidor de Aplicação Tomcat 6.0.....	29
5.8 – Fingerprint SDK e Leitor Biométrico.....	29
5.9.1 - Tipo de Pesquisa.....	30
5.9.2 – Organograma.....	31
<b>6 – Resultados. 32</b>	
6.1 – Padrão MVC.....	33
6.2 – Cadastros.....	34
6.3 – Ponto (Lançamento Manual, Abertura, Fechamento, Consulta).....	29
6.4 – Cadastro de digital e lançamento de ponto através do reconhecimento.....	43
6.5 – Banco de Dados.....	45
6.6 – Geração de Relatório.....	47
<b>7 – Conclusão.....</b>	<b>48</b>

## Lista de Imagens

Figura 1 - Características Biométricas .....	12
Figura 2 - Esquema de um sistema Biométrico .....	15
Figura 3 – Componentes de um Diagrama de Caso de Uso.....	18
Figura 4 – Exemplo Diagrama de Caso de Uso.....	18
Figura 5 – Exemplo de caso para telefone Celular.....	19
Figura 6– Exemplo de notação para classe em UML.....	20
Figura 7– Exemplo de Associação Binária Fonte: Reis,2000. ....	21
Figura 8– Exemplo de Associação Ternária.....	21
Figura 9– Exemplo de Diagrama de Colaboração.....	22
Figura 10 – Configuração de um sistema de banco de dados simplificado .....	24
Figura 11– Organograma do sistema de controle de chamadas escolares.....	31
Figura 12 – Reconhecimento da impressão Digital.....	32
Figura 13 – Estrutura do Padrão MVC (Model View Control).....	33
Figura 14 – Classes Mapeadas Para Persistirem na Base de Dados.....	34
Figura 15 – Classe Professor Mapeada. ....	36
Figura 16 – Cadastro de Professor (Listagem) .....	38
Figura 17 - Cadastro de professor (inclusão).....	39
Figura 18 – Lançamento de Ponto Manual .....	40
Figura 19 – Abertura de Ponto.....	41
Figura 20 – Fechamento de Ponto .....	42
Figura 21 – Consulta de Ponto .....	43
Figura 22 – Cadastro Aluno Digital.....	44
Figura 23 – Lançamento de Ponto .....	45
Figura 24 – Diagrama Banco de Dados.....	46
Figura 25 – Arquivo de configuração Hibernate. ....	47
Figura 26 – Impressão de Relatório .....	47

## **Lista de Tabelas**

Tabela 1 - Comparação das Características Biométricas.....	13
Tabela 2 - Distribuição Horizontal (por finalidades das principais aplicações biométricas).....	14

## 1 – Introdução.

Nos tempos de hoje é comum alunos de universidade, ensino médio e fundamental, não cumprirem a carga horária devida para um dia letivo de aula, por diversos motivos diferentes provocado por diversos fatores externos, acarretados quase sempre por três situações diferentes, são elas: atraso na entrada da aula, atraso na volta do intervalo de aula e saída antes do término da aula, isso provoca uma grande ineficiência no controle de presença dos alunos, muitas vezes a sala de aula tem um grande número de alunos e o professor não consegue ter o controle exato de quem não assistiu todas as aulas e de quem entra e sai da aula, o que poderia ser possível com o auxílio de um software para gerenciar o controle de carga horária com o reconhecimento dos alunos através da biometria digital.

Com base em Costa, Obelheiro e Fraga (2006) a biometria é o estudo do reconhecimento dos traços que podem ser medidos e computados na forma de um identificador único, difícil de forjar ou se alterar. Diversos tipos de tecnologias biométricas encontram-se difundidas no mercado, como por exemplo: reconhecimento das mãos, íris, faces e impressões digitais. Todas se encontram em um estágio bastante avançados e satisfatório de desenvolvimento, disponíveis em diversos softwares no mercado.

Presentemente, as sociedades atribuem relevância crescente a sistemas que contribuam para aumentar os níveis de segurança e fiabilidade, essencialmente devido a preocupações com o terrorismo ou outros atos extremistas. Neste contexto, o uso de sistemas biométricos tem sido crescentemente encorajado, quer por entidades públicas ou privadas, com vista a substituir ou aumentar os níveis de segurança tradicionais. Basicamente, o objetivo é estabelecer uma identidade para um indivíduo, baseado no que ele é em vez de o que ele possui ou o que ele sabe (por exemplo, um cartão de identificação ou uma palavra-passe). (Proença, 2006, p.7)

A biometria digital é amplamente conhecida como um método de verificação e reconhecimento biométrico de alta precisão, muito difundido no mercado ela tem sido a solução tecnológica para os mais diversos ramos de atividades, temos como exemplo várias empresas trocando o cartão de ponto pelo ponto biométrico através da digital, por oferecer maior confiabilidade, tanto para o empregado como para o empregador, outro exemplo que podemos citar é o sistema do Departamento de

Trânsito o DETRAN DIGITAL, o próprio citou como objetivo obter maior confiabilidade no sistema de emissão de Carteira Nacional de Habilitação (CNH), no efetivo reconhecimento do candidato à habilitação, maior agilidade no processo de emissão de CNH e a confirmação da identidade do cidadão já cadastrado sem a necessidade de apresentação de documentos.

Em estudos dermatológicos, a máxima diferença entre impressões digitais tem sido encontrada entre indivíduos de diferentes raças. Pessoas da mesma raça, porém sem grau de parentesco, possuem similaridade muito pequena nas digitais. Pai e filho possuem alguma similaridade, por compartilharem metade dos genes. Gêmeos monozigóticos (idênticos) possuem a máxima similaridade. Estima-se que 95% das características das digitais de gêmeos idênticos sejam iguais (Maltoni et al. Apput Costa, Obelheiro, Fraga 2003).

Um sistema biométrico pode ser encarado como um sistema de reconhecimento de padrões de propósitos específicos (Bolle et al. Apput Costa, Obelheiro, Fraga, 2002).

### **1.1 - Problema de Pesquisa.**

Como desenvolver um software para informatizar e automatizar o processo de controle de presença de alunos utilizando a biometria digital?

### **1.2 – Justificativa.**

Segundo Araújo (2006) a prática de se utilizar a impressão digital para o reconhecimento individual de uma pessoa é considerada um dos métodos mais eficazes e seguros para obter o reconhecimento, temos como exemplo a segurança nacional dos Estados Unidos da América, o FBI faz uso deste método e foi quem fundou a tecnologia de desenvolvimento para escaneamento de impressão digital em 1975, no início muito utilizada para desfechos criminais, porém hoje é utilizada para os mais diversos fins, tais como controle de ponto, fechaduras eletrônicas, caixas eletrônicos, alarmes automotivos, cofres, etc. Hoje esta tecnologia se encontra

difundida pelo mundo todo e não se questiona mais a viabilidade da biometria digital.

### **1.3 - Objetivo Geral.**

Desenvolver o protótipo de um software para informatizar e automatizar o controle de presença de alunos, implementando a biometria digital para realizar o reconhecimento dos alunos.

#### **1.3.1 - Objetivos Específicos.**

- Implementar a comunicação do protótipo de gerenciamento de chamadas, com um leitor biométrico digital, para o reconhecimento dos alunos.

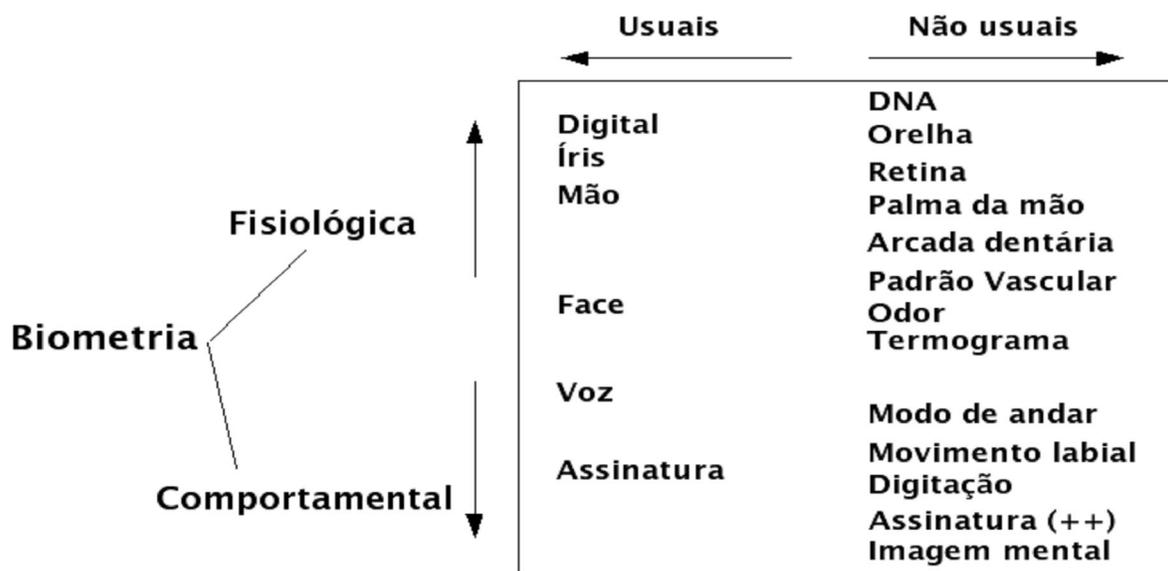
## 2 – Biometria.

Segundo Clark(1994) et all, apput. Costa, Obelheiro, Fraga, (2006) qualquer característica fisiológica ou comportamental humana pode ser considerada uma característica biométrica desde que atenda alguns requisitos, são eles:

- **Universalidade:** toda a população deve possuir a característica, tendo em vista que na prática existem pessoas que não possui certas características como por exemplos pessoas que não possuem impressões digitais.
- **Unicidade:** uma característica deve ser única para cada individuo ou seja não pode existir em hipótese alguma a possibilidade de duas ou mais características idênticas.
- **Permanência:** a característica deve ser imutável, porém na prática existem varias características que podem sofrer alterações, dependendo do clima, do estado de saúde do individuo, por estado emocional e outros diversos fatores externos.
- **Coleta:** a característica tem que ser adquirida e armazenada através de algum dispositivo específico, na prática todas as características biométricas utilizadas hoje de forma comercial, atendem a este requisito.
- **Aceitação:** a coleta da característica tem que ser permitida pelo individuo em questão, existem vários fatores que acabam impedindo com que o individuo aceite a coleta, como por exemplo a questão da higiene.

Porém na prática segundo Costa, Obelheiro e Fraga, (2006) nenhuma característica biométrica consegue atender com perfeição aos requisitos de uma característica biométrica ideal. Ao longo do tempo, diversas tecnologias biométricas

foram desenvolvidas. As tecnologias biométricas existentes são classificadas, por conveniência, em dois grupos. O primeiro grupo está baseado em características chamadas de fisiológicas ou estáticas. Essas características são traços fisiológicos, originários da carga genética do indivíduo, e essencialmente varia pouco (ou nada) ao longo do tempo. As principais características estáticas são a aparência facial, o padrão da íris, a geometria das mãos e as impressões digitais.



**Figura 1 - Características Biométricas**  
**Fonte: Costa, Obelheiro, Fraga, 2006.**

Segundo Costa, Obelheiro e Fraga, (2006) as seis características exibidas na imagem acima são consideradas as utilizadas com mais frequência, as características fisiológicas dependem principalmente das cargas genéticas e as comportamentais varia de acordo com a experiência e aprendizado.

O segundo grupo de tecnologias biométricas está baseado em características chamadas de comportamentais ou dinâmicas. São características aprendidas ou desenvolvidas ao longo da utilização constante, e que podem variar fortemente ao longo do tempo. Além disso, podem ser facilmente alteradas pela vontade ou estado do usuário. Assim, até mesmo duas amostras consecutivas podem mudar bastante. As principais características dinâmicas utilizadas são o padrão de voz e a dinâmica da assinatura (Costa, Obelheiro, Fraga, 2006)

Segundo Teixeira (2011) as características fisiológicas são mais confiáveis e precisas do que as comportamentais, por serem mais difíceis de serem copiadas ou falsificadas, e por não sofrerem variações com o passar do tempo. A tabela abaixo ilustra os níveis das características classificando em A (alto), M (médio), B (baixo).

**Tabela 1: Comparação das Características Biométricas. Legenda: A(Alto), M(Médio), B(Baixo).**

Fonte: Teixeira, 2011.

Característica	Universalidade	Distinguíbilidade	Imutabilidade	Coletabilidade	Desempenho	Aceitabilidade	Circunvenção
DNA	A	A	A	B	A	B	B
Orelha	M	M	A	M	M	A	M
Face	A	B	M	A	B	A	A
Termograma facial	A	A	B	A	M	A	B
<b>Impressão digital</b>	M	<b>A</b>	<b>A</b>	M	<b>A</b>	<b>M</b>	<b>M</b>
Caminhar	M	B	B	A	B	A	M
Geometria da mão	M	M	M	A	M	M	M
Veias da mão	M	M	M	M	M	M	B
Íris	A	A	A	M	A	B	B
Teclar	B	B	B	M	B	M	M
Odor	A	A	A	B	B	M	B
Retina	A	A	M	B	A	B	B
Assinatura	B	B	B	A	B	A	A
Voz	M	B	B	M	B	A	A

A tabela 1 ilustra diversas formas de biometria e o grau de suas características, percebe-se que a Impressão Digital não oferece nenhuma característica de grau Baixo, e das 7 características apresentadas ela apresenta um Alto nível em 3 delas, podemos chamar a atenção para distinguibilidade e imutabilidade, pois com essas 2 características pode-se garantir um bom reconhecimento do indivíduo e garantir que esse reconhecimento não sofra perda de distinguibilidade com o passar do tempo.

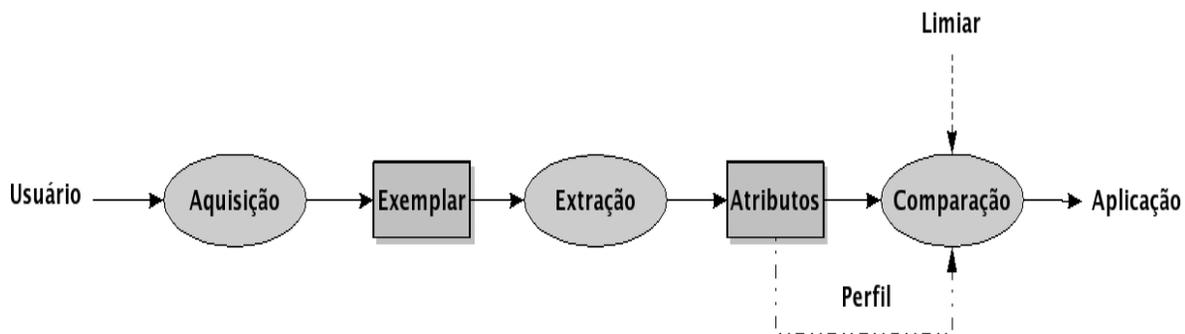
As tecnologias biométricas são utilizadas em vários tipos de aplicações na maioria das vezes visando o aumento da segurança e a confiabilidade da aplicação. Dentre os diversos conjuntos possíveis, dependendo do refinamento, pode-se citar a classificação de alto nível de sete grupos usada no relatório BITE Market Report (Bite 2005, apput Costa ,Obelheiro, Fraga, 2006), como mostra a tabela abaixo.

**Tabela 2 - Distribuição Horizontal (por finalidades das principais aplicações biométricas)**  
**Fonte: (Bite 2005, apput. Costa, Obelheiro, Fraga, 2006)**

<b>Finalidade</b>	<b>Utilização</b>
Identificação Criminal	28%
Controle de acesso e atendimento	22%
Identificação Civil	21%
Segurança de redes e computadores	19%
Autenticação em pontos de vendas, ATM's e varejo	4%
Autenticação telefônica e comércio eletrônico	3%
Vigilância e filtragem	3%

### 3 – Sistemas Biométricos.

Segundo Bolle et al, apud. Costa, Obelheiro, Fraga, (2006) seja qual for a característica biométrica utilizada, ela deve estar enquadrada em um sistema biométrico. Um sistema biométrico pode ser encarado como um sistema de reconhecimento de padrões de propósito específico. Num sistema biométrico, o usuário é previamente registrado, esse registro deve conter seu perfil biométrico (digital, iris, palma da mão, etc). Características particulares e únicas são retiradas desse perfil biométrico anteriormente armazenado, posteriormente é feita uma comparação com o perfil armazenado e o perfil apresentado momentaneamente, após a comparação o sistema avalia os dados e decide se os perfis são da mesma pessoa.



**Figura 2 - Esquema de um sistema Biométrico**  
**Fonte: Costa, Obelheiro, Fraga (2006).**

Modelo Simples de um Sistema Biométrico(Costa, et all. 2006):

- **Aquisição:** O processo de aquisição é onde o perfil biométrico é adquirido através de algum dispositivo, é um dos processo mais importantes pois a qualidade com que se adquire o perfil é essencial para a qualidade do reconhecimento, a maior dificuldade deste processo é balancear a qualidade da aquisição do perfil biométrico com a inconveniência para o usuário.

- **Extração:** Na extração é onde se adquire os atributos principais do perfil biométrico, pois várias informações do perfil são inúteis para a comparação, na extração é onde se adquire o conjunto de informações úteis conhecido como template.
- **Registro:** É o processo onde é feito o cadastramento do template no sistema, o perfil biométrico é salvo para uma futura comparação .
- **Comparação e decisão:** O processo de comparação, ou *matching*, verifica qual é o grau de similaridade entre as características extraídas da amostra do usuário e o perfil armazenado previamente. Este processo fornece um escore representativo da similaridade entre os dois conjuntos de dados. Caso a similaridade seja superior a um certo limite previamente determinado, conhecido como limiar, ou *threshold*, a decisão é aceitar o usuário, ou seja, uma autenticação válida. Caso a similaridade seja inferior ao limiar, a decisão é não aceitar o usuário, e então temos um usuário não autenticado.

## **4 – Desenvolvimento de Software.**

Estaremos discutindo sobre o desenvolvimento do software, partindo da modelagem do sistema utilizando UML (Unified Modelling Language), citando vários exemplos de diagramas que serão utilizados para o desenvolvimento e abordando como funciona um SGDB (Sistema de Gerenciamento de Banco de Dados), e sua fundamental importância para o software.

### **4.1 – UML (Unified Modelling Language).**

Segundo Medeiros (2004) UML consiste na principal linguagem para a descrição de sistemas Orientados a Objetos, tendo sido definida como padrão pelo organismo internacional a OMG, para definição de padrões da orientação objetos.

#### **4.1.1 – Diagramas de Casos de Uso.**

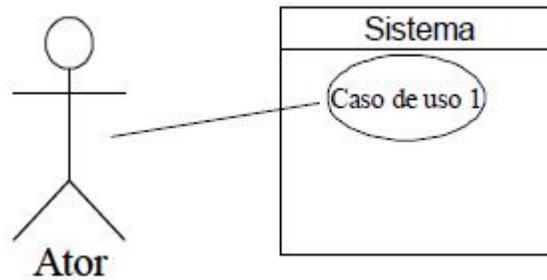
Segundo Reis (2003) os diagramas de caso de uso, são aqueles que descrevem exatamente a visão externa do sistema e suas interações com o mundo exterior.

A modelagem de caso de uso é uma técnica utilizada para descrever a funcionalidade de um sistema através de atores externos interagindo em casos de uso. Atores representam um papel e iniciam o caso de uso que, por sua vez, deve entregar um valor tangível de retorno ao ator. Atores e casos de uso estão conectados através de associações e podem ter relacionamentos de generalização que descreva o comportamento comum em superclasses herdadas por uma ou mais subclasses especializadas. (Reis, 2003, p4).

Segundo Reis (2003) os principais objetivos dos diagramas de caso de uso são:

- Descrever os requisitos funcionais do sistema de maneira uniforme para usuários e desenvolvedores.
- Descrever de forma clara e consistente as responsabilidades a serem cumpridas pelo sistema.
- Oferecer possíveis situações do mundo real para a fase de testes do sistema.

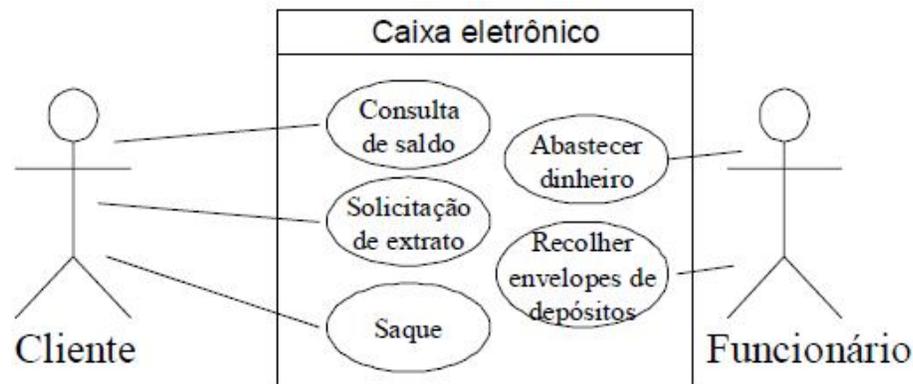
Os elementos básico de um diagrama de caso de uso são : ator, caso de uso, interação e sistema, como mostra figura abaixo.



**Figura 3 – Componentes de um Diagrama de Caso de Uso.**  
Fonte:Reis, 2003

#### 4.1.2 – Exemplos de Diagrama de Caso de Uso.

O exemplo abaixo mostra um diagrama de caso de uso que ilustra os serviços tipicamente fornecidos por um Caixa eletrônico Bancário. O diagrama distingue dois grupos de serviços(Reis, 2003).



**Figura 4 – Exemplo Diagrama de Caso de Uso**  
Fonte: Reis 2003, apput. Fur, 1998.

O exemplo abaixo demonstra o diagrama de caso de uso de um celular, deve se observar que o serviço “Faz ligação ” faz uso de “Identificação destinatário” e opcionalmente utiliza o serviço “Fazer Ligação e Conferência” (Reis, 2003).

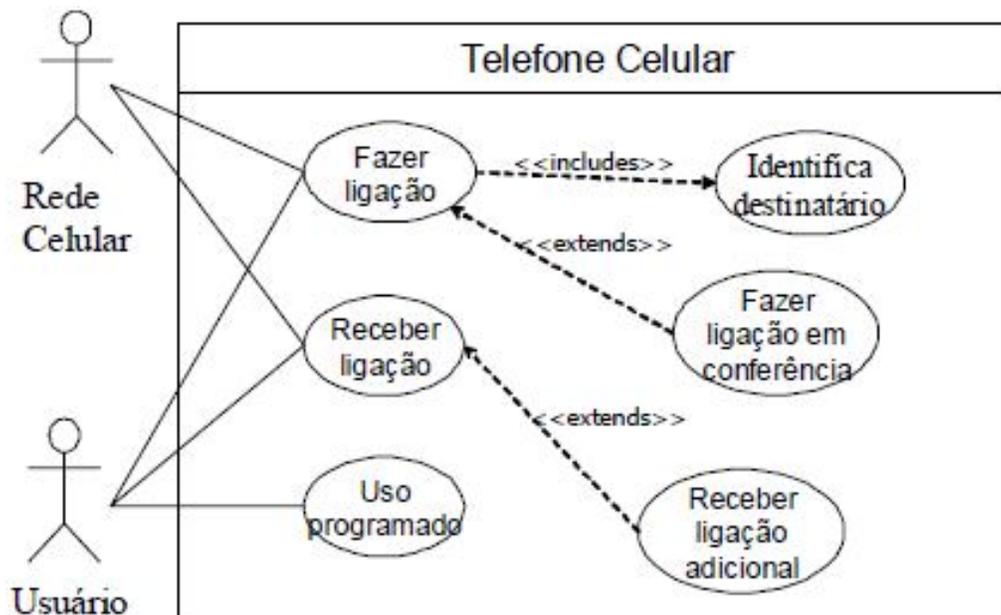
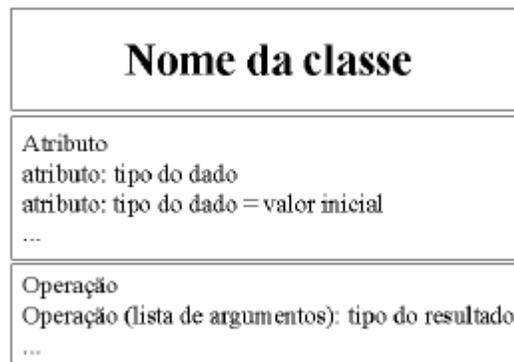


Figura 5 – Exemplo de caso para telefone Celular.  
Fonte: Reis, 2003 apud Boo, 2000.

#### 4.1.3 – Diagramas de Classes em UML.

Segundo Medeiros (2004) um diagrama de classes denota a estrutura estática de um sistema e suas classes representam coisas que são manipuladas por esse sistema. O que se utiliza para a notação que representa o diagrama de classes em UML é baseado na notação de Diagramas Entidade-Relacionamento, e no Modelo de Objetos.

Uma classe é representada por um retângulo sólido, dividido em três partes na horizontal, o primeiro para o nome da classe, o segundo para os atributos das classes, e a terceira parte para a declaração das operações definidas para a classe.



**Figura 6– Exemplo de notação para classe em UML**  
**Fonte: Reis,2003**

Segundo Reis(2003) os principais tipos de relacionamento entre classes são:

- **Generalização e Especialização (Herança):** Indica o relacionamento de um elemento mais geral e um elemento mais específico como superclasse e subclasse, a subclasse só contém informações adicionais acerca da superclasse.
- **Agregação:** Usada para denotar relacionamento todo/parte. Como por exemplo um pedido é parte de uma compra.
- **Associação (Simples):** Usada para representar o relacionamento entre as classes (por exemplo, um cliente pode alugar várias fitas de vídeo).
- **Dependência:** Um relacionamento entre um elemento dependente e um independente, onde a mudança do elemento independente acarreta em mudanças no elemento dependente.

#### 4.1.4 – Associações Simples

Uma associação descreve o vínculo dos objetos das classes relacionadas, existem 3 tipos de associação são elas:

**Associação Unária:** Relacionamento entre uma classe e ela mesma, bastante conhecida como associação recursiva, cujo seu relacionamento pode conectar dois diferentes objetos de uma mesma classe(Reis, 2000).

**Associação Binária:** Expressa o relacionamento entre duas classes distintas.

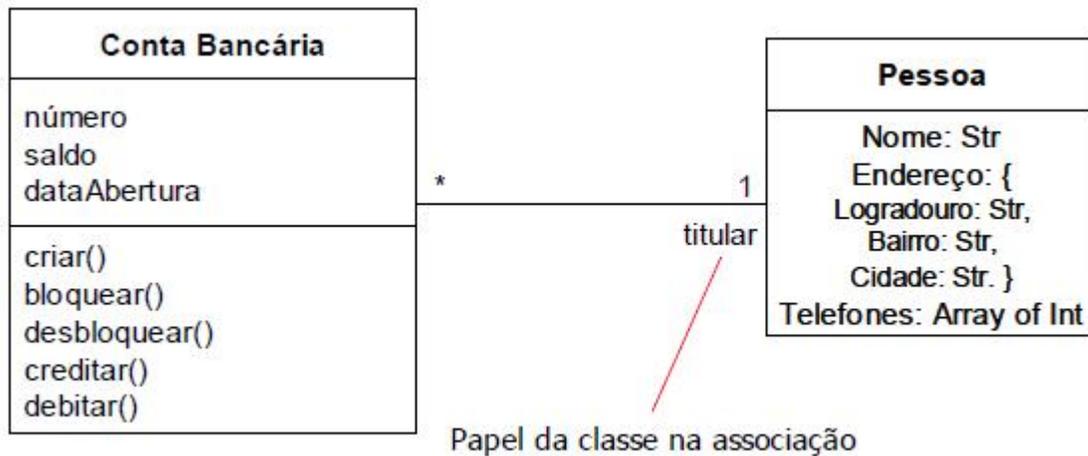


Figura 7– Exemplo de Associação Binária Fonte: Reis,2000.

**Associação n-ária:** Associação entre três ou mais classes. Neste caso a notação inclui um losango para representar a associação.

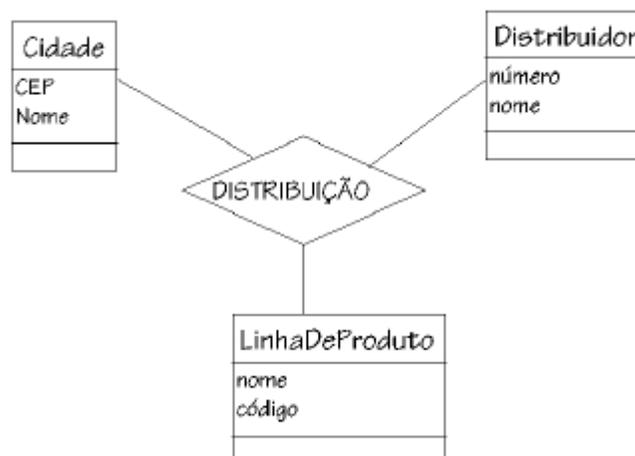


Figura 8– Exemplo de Associação Ternária Fonte: Reis,2003.

#### 4.1.5 – Diagrama de Colaboração.

Segundo Reis(2003) um diagrama de colaboração mostra uma interação organizada em torno dos objetos e seus vínculos formando padrões.

O diagrama de seqüência e o diagrama de colaboração expressam as mesmas informações, porém apresentam de formas diferentes. O primeiro exhibe uma seqüência explícita de mensagens e é melhor para especificações de tempo real (dimensão tempo) e para cenários complexos, enquanto o segundo mostra os vínculos entre objetos e é melhor para entender os efeitos num determinado objeto (dimensão espaço). Para decidir qual diagrama deve ser utilizado para estudar uma interação, uma regra geral é escolher o diagrama de colaboração quando o objeto e seus vínculos facilitam a compreensão da interação, e escolher o diagrama de seqüência somente se a seqüência necessita ser evidenciada.(Reis, 2003, p24)

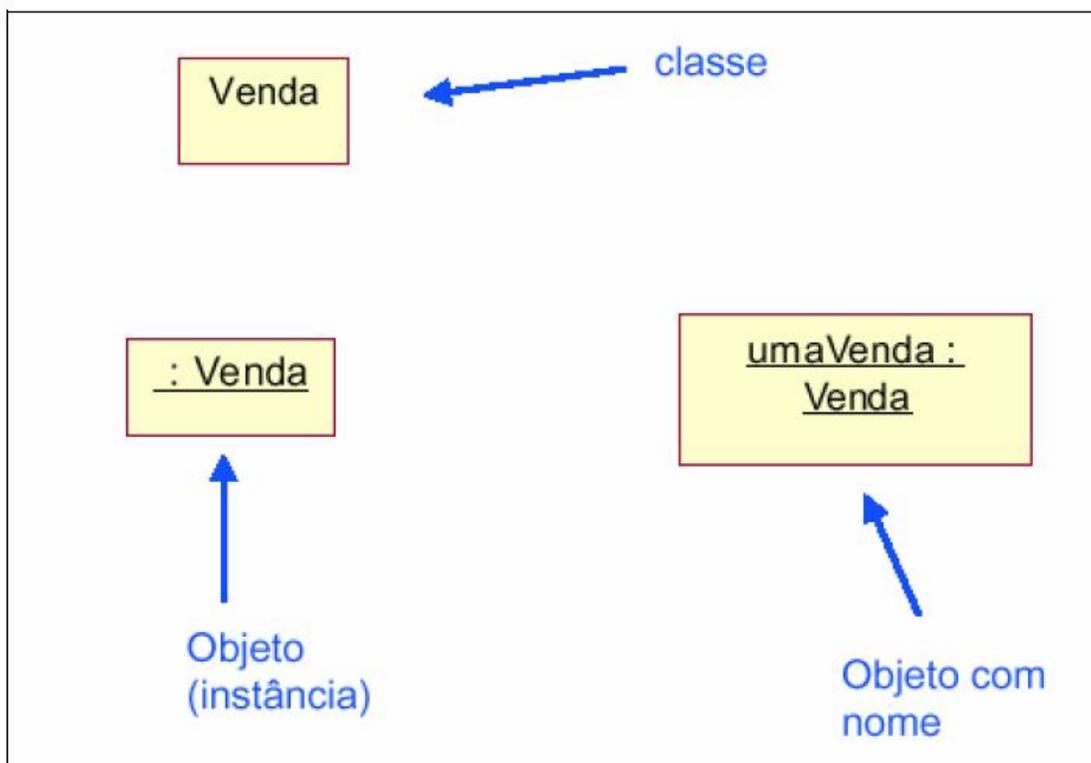
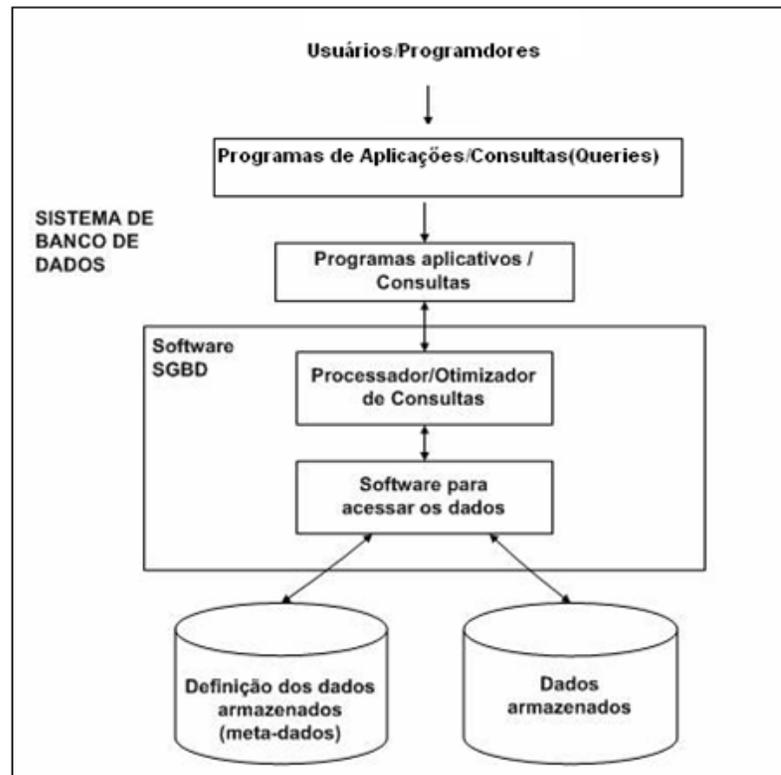


Figura 9– Exemplo de Diagrama de Colaboração  
Fonte: Reis, 2003.

## **4.2 – Sistema Gerenciador de Banco de Dados (SGDB).**

Segundo Ramez, Heuser (2010) um SGDB é uma coleção de programas que permite aos usuários criar e manter uma base de dados, portando um SGDB é um software de propósito geral que facilita os processos de definição, construção e manipulação de banco de dados entre vários usuários e aplicações . A definição de um banco de dados implica em especificar os tipos de dados, suas estruturas e restrições para os dados que serão inseridos na base de dados. A construção da base de dados é o processo de armazenar os dados em alguma mídia apropriada que vai ser controlada pelo SGDB, a manipulação inclui diversas funções, como consultas na base de dados para recuperar dados específicos, atualizações do banco para refletir nas mudanças necessárias para geração de algum relatório específico. O compartilhamento permite aos múltiplos usuários e programas acessar, de forma concorrente, o banco de dados. Outras funções importantes são a segurança e manutenção de uma base de dados por um longo período, a proteção seria a proteção de sistemas contra o mau funcionamento ou falhas do hardware ou do software, e segurança contra acessos não permitidos ou maliciosos. Um banco de dados típicos pode ter um ciclo de vida muito longo, por isso um SGDB tem que ser capaz de manter um sistema de banco de dados e que permita a evolução dos requisitos que se alteram com a evolução das tecnologias da área.



**Figura 10 – Configuração de um sistema de banco de dados simplificado**  
 Fonte: Ramez,2010.

Segundo Ramez (2010) apesar das diversas vantagens em se utilizar um SGDB, existem algumas situações em que esse sistema pode envolver custos altos e desnecessários, que normalmente não ocorreriam no processamento tradicional de arquivos. Os altos custos de utilizar o SGDB são devidos a:

- Investimentos iniciais altos, em softwares hardwares e treinamentos.
- Generalidade que o SGDB fornece para definir e processar os dados.
- Custos elevados para oferecer a segurança necessária para uma base de dados, controle de concorrência, recuperação e funções de integridade.

Com base em Ramez (2010) problemas adicionais ainda podem surgir se os projetistas do banco de dados e o DBA não projetarem o banco de forma adequada ou se a aplicação não for implementada de forma adequada.

#### **4.2.1 – Classificação dos Sistemas Gerenciadores de Banco de Dados.**

Segundo Ramez, Heuser (2010) diversos critérios são utilizados para classificar um SGDB, o primeiro é o modelo de dados no qual o SGBD é utilizado, atualmente o principal modelo de SGBD utilizado é o modelo de dados relacional, o modelo de dados de objeto foi implementado em alguns sistemas comerciais, mas seu uso não foi muito difundido. Ainda existe o modelo de dados hierárquicos e de rede, se tratam de modelos muito antigos e que atualmente existem em aplicações legadas(antigas). Os SGDB's relacionais estão evoluindo continuamente e, em especial, tem incorporado muitos do conceitos que foram desenvolvidos nos bancos de dados de objetos. Essa evolução gerou uma nova categoria de SGDB's conhecida hoje como SGDB's objeto-relacional. Sendo assim podemos classificar os SGDB's com base no modelo de dados: relacional, objeto, objeto-relacional, hierárquico, de rede e outros.

## 5 – Materiais e Métodos.

Este capítulo contém todos os materiais que serão utilizados para o desenvolvimento do software, e os métodos de como esses materiais foram utilizados.

### 5.1 – Linguagem de Programação.

A linguagem que foi utilizada para o desenvolvimento do sistema foi o Java, segundo Cesta (1996) Java é uma linguagem compilada, ou seja um programa em Java é compilado para o chamado byte-code, que é próximo as instruções de máquina, porém não de uma máquina real, trata-se de um código de uma máquina virtual (*Java Virtual Machine*) idealizada pelos criadores da linguagem e por isso Java pode ser mais rápida do que se fosse interpretada. Java foi criada para ser portátil, o seu byte-code é gerado pelo compilador para a sua aplicação específica pode ser transportado entre plataformas distintas que suportam Java (Windows, Linux, Mac/Os), e um grande diferencial é que não é preciso recompilar um programa para que ele rode em uma plataforma diferente, ao contrário do que acontece por exemplo com a linguagem C e outras linguagens. Java é uma linguagem orientada a objetos, que suporta concorrência, ou seja permita a criação de maneira fácil de vários *threads* de execução, Java também oferece facilidade para a programação com *sockets*, *method call*, sem contar as diversas ferramentas (*frameworks*) gratuitas para o auxílio do desenvolvimento, tornando o desenvolvimento muito mais ágil, fácil e prazeroso.

### 5.2 – Ambiente Integrado de Desenvolvimento (IDE).

As IDE's (*Integrated Development Environment*) ou Ambiente Integrado de Desenvolvimento, são programas criados para auxiliar os desenvolvedores em diversos processos, agilizando e auxiliando na correção de erros com depuradores, na formatação do código com editores, entre outros diversos processos.

A *IDE* escolhida para o desenvolvimento do sistema de controle de chamadas foi o *NetBeans* 6.9.1, trata-se de uma *IDE* livre (gratuita) muito utilizada pelos desenvolvedores que trabalham com a linguagem Java, ela visa simplificar o desenvolvimento e aumentar a produtividade, totalmente escrita em Java ela suporta também outras diversas linguagens como PHP, *Ruby*, C, C++.

### **5.3 – Banco de Dados.**

O banco de dados escolhido para o desenvolvimento do sistema foi o PostgreSQL (Versão 9.0), segundo ([postgresql.org](http://postgresql.org)) trata-se de um SGBD (Sistema Gerencial de Banco de Dados) objeto-relacional de código fonte aberto, possui mais de quinze anos de desenvolvimento ativo e uma arquitetura que comprovadamente ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões. O PostgreSQL roda em todos os grandes sistemas operacionais, é totalmente compatível com ACID, tem suporte completo a chaves estrangeiras (*joins*), visões, gatilhos, e procedimento armazenados (em múltiplas linguagens). Inclui a maior parte dos tipos de dados do ISO SQL:1999, suporta também o armazenamento de objetos binários, incluindo figuras, sons ou vídeos. Possui *interfaces* nativas de desenvolvimento para C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre outros além de possuir uma excepcional documentação. O PostgreSQL também possui uma comunidade bastante ativa tanto no Brasil como no exterior, é um líder em tecnologia e é conhecido como o banco de dados de código aberto mais avançado no mundo, possui uma excelente performance e segurança.

### **5.4 – Ferramenta de Mapeamento Objeto/Relacional (Hibernate).**

Foi utilizado o *Framework* Hibernate para agilizar o desenvolvimento referente aos processos relacionados ao banco de dados.

Segundo Linhares (2008) o Hibernate é uma grande ferramenta de mapeamento objeto/relacional para a linguagem Java. Ela transforma os dados tabulares de um banco de dados em um grafo de objetos definido pelo desenvolvedor. Utilizando o

Hibernate, o desenvolvedor ganha agilidade e se livra de escrever muito do código de acesso a banco de dados e de SQL que ele escreveria não usando a ferramenta, acelerando a velocidade do seu desenvolvimento de uma forma fantástica.

### 5.5 – Java Server Faces 2.0 (JSF).

Segundo Nunes, Magalhães (2010) Java Server Faces é uma especificação técnica do Java Community Process que foi publicada em 2004, ele tem o objetivo de padronizar um framework para desenvolvimento de aplicações web, visando sempre maximizar a produtividade e minimizar a complexidade de manutenções, facilitando e evoluindo o uso de tecnologia Ajax, além de proporcionar melhores integrações com diversas tecnologias web.

JSF já é o framework Java mais usado para desenvolvimento de aplicações web com Java. Seu excelente conjunto de funcionalidades faz com que esta tecnologia seja favorita no cenário de desenvolvimento em que está inserida. Isto se deve a vários motivos:

- É uma especificação que faz parte da Enterprise Edition desde a versão 5;
- É um framework cuja API também foi pensada para os desenvolvedores de IDEs (Eclipse, NetBeans, JDeveloper, ...), não somente para os desenvolvedores de aplicação;
- Tem soluções para necessidades comuns do desenvolvimento web, propostas e consolidadas a partir da experiência de outros frameworks MVC de mercado, tais como Struts, Tapestry e Spring Web Flow;
- Faz parte da certificação SCWCD 6 junto com Servlets e JSP;
- Possui uma comunidade bastante ativa em fóruns e no IRC;
- Possui bons materiais de estudo no formato de livros, documentos digitais e sites;
- Exige pouco conhecimento inicial para construção de interfaces de usuário tradicionais, como consultas baseadas em lista e cadastros;
- Preparada para ser estendida, possui ótimas bibliotecas de componentes livres ou pagas desenvolvidas por terceiros. Estas bibliotecas oferecem recursos interativos similares aos encontrados na interface de usuário para desktop, evitando assim a necessidade da equipe de desenvolvimento dominar tecnologias cliente como HTML, Javascript e CSS;
- Oferece capacidades interativas integradas com AJAX. Isto com mínima necessidade de escrever Javascript. (Nunes, Magalhães 2009)

### 5.6 – PrimeFaces 2.2.1

Segundo a Optimus Prime (2011), o PrimeFaces é um conjunto de componentes para Java Server Faces de código aberto, com mais de 100 componentes integrados com Ajax. O PrimeFaces é mantido pela Prime-Technology, uma empresa de desenvolvimento de software Turco, especializada em consultoria Agile e Java EE. O projeto é liderado por Cagatay Civici(Optimus Prime) e um grupo de especialistas em JSF.

### **5.7 – Servidor de Aplicação Tomcat 6.0**

O servidor de aplicação escolhido foi o Apache Tomcat, trata-se de um projeto de código fonte aberto e que possui plugin de integração com o NetBeans, segundo a Fundação Apache(2011) o Apache Tomcat é desenvolvido em um ambiente aberto e participativo e liberado sob a Licença Apache 2.

### **5.8 – Fingerprint SDK e Leitor Biométrico.**

Segundo a Griaule Biometrics(2008) o Fingerprint é um kit de desenvolvimento de software, que lhe permite integrar uma ampla gama de aplicações por conter suporte a dezenas de linguagens, existem duas versões deste kit de desenvolvimento disponível, um para Windows que suporta vários tipos de linguagens, e outro para Java que possibilita a integração com sistemas multi-plataformas. O Fingerprint SDK é um software comercial pago, que possui uma versão trial gratuita para testes de possíveis implementações. O leitor utilizado será o Microsof Biometrics Reader USB.

Seu principal objetivo é permitir o desenvolvimento de aplicações personalizadas, tais como sistemas de autorização, gerenciamento de transações, tempo e atendimento, pontos de identificação de venda e controle de acesso físico autenticados através de impressão digital (Griaule Biometrics, 2008).

### **5.9.1 - Tipo de Pesquisa.**

A pesquisa foi experimental, ela será toda voltada para o desenvolvimento de um sistema de gerenciamento de controle de chamadas escolares, o sistema vai conter toda a parte cadastral dos alunos assim como dos professores, o diferencial do sistema será tanto a integração com a biometria digital, como o tratamento do tempo em sala de aula, o sistema vai trabalhar em cima de carga horária, como se fosse um sistema de ponto de uma empresa, porém no fim do semestre teremos as horas aulas de cada professor como as horas aulas dos alunos, será desenvolvida também toda a modelagem do banco de dados, assim como a implementação da biometria digital através de um leitor biométrico.

### 5.9.2 – Organograma.

O organograma abaixo ilustra como o sistema funciona. Primeiramente deverá ser realizada a parte cadastral, inserindo todos os dados necessários para o sistema funcionar. Ao chegar na sala de aula o aluno bate o ponto colocando o dedo no leitor biométrico, o sistema procura na base de dados um template (impressão digital) compatível com o aluno que colocou o dedo no leitor, ao encontrar o template o sistema lança o ponto para o aluno com a hora atual do servidor, esse mesmo processo deve ser feito na hora que o aluno sai para o intervalo, quando ele volta do intervalo e quando termina a aula. Quando o período de aula terminar o sistema vai conter na base de dados todos os dados necessário para computar o tempo em que o aluno ficou presente na sala de aula, e a quantidade de horas aulas que foi dada naquele período, dessa forma é possível gerar relatório informando se o aluno atingiu ou não o mínimo exigido pela entidade escolar para ser aprovado.

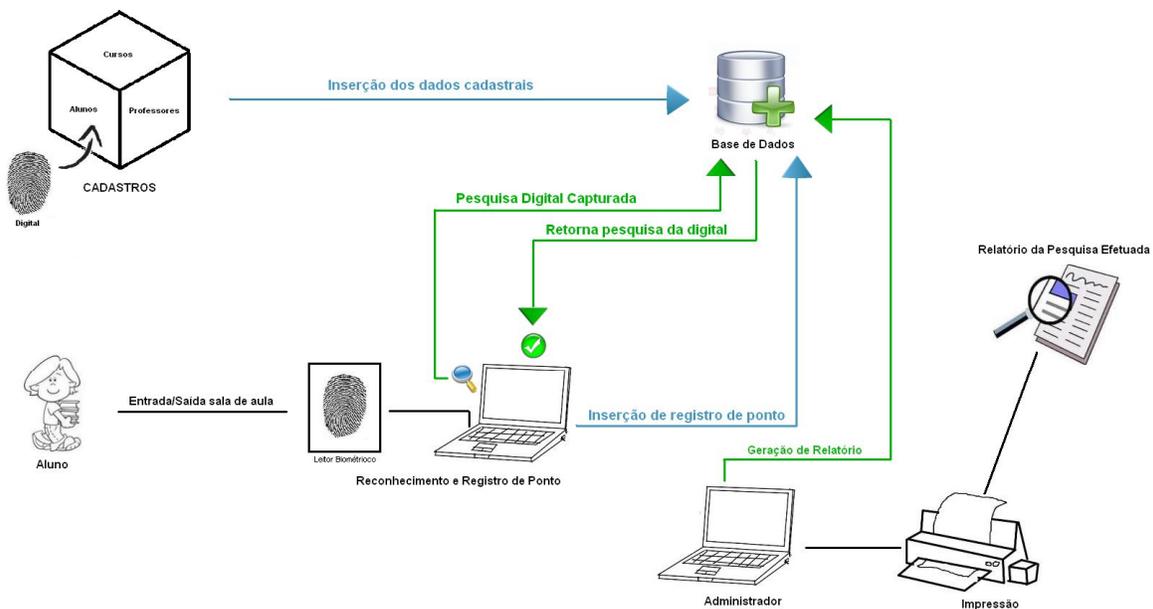


Figura 11– Organograma do sistema de controle de chamadas escolares

## 6 – Resultados.

A biometria é uma ferramenta atualmente muito estudada tanto para reconhecimento de face, fala, digital entre outros. Neste trabalho está sendo apresentada uma ferramenta que utiliza a biometria para facilitar o trabalho do professor na sala de aula, onde, a chamada será realizada através da biometria.

Para o desenvolvimento do software foi utilizada tecnologias e metodologias atuais tais como: Hibernate, JSF 2.0, Primefaces e MVC.

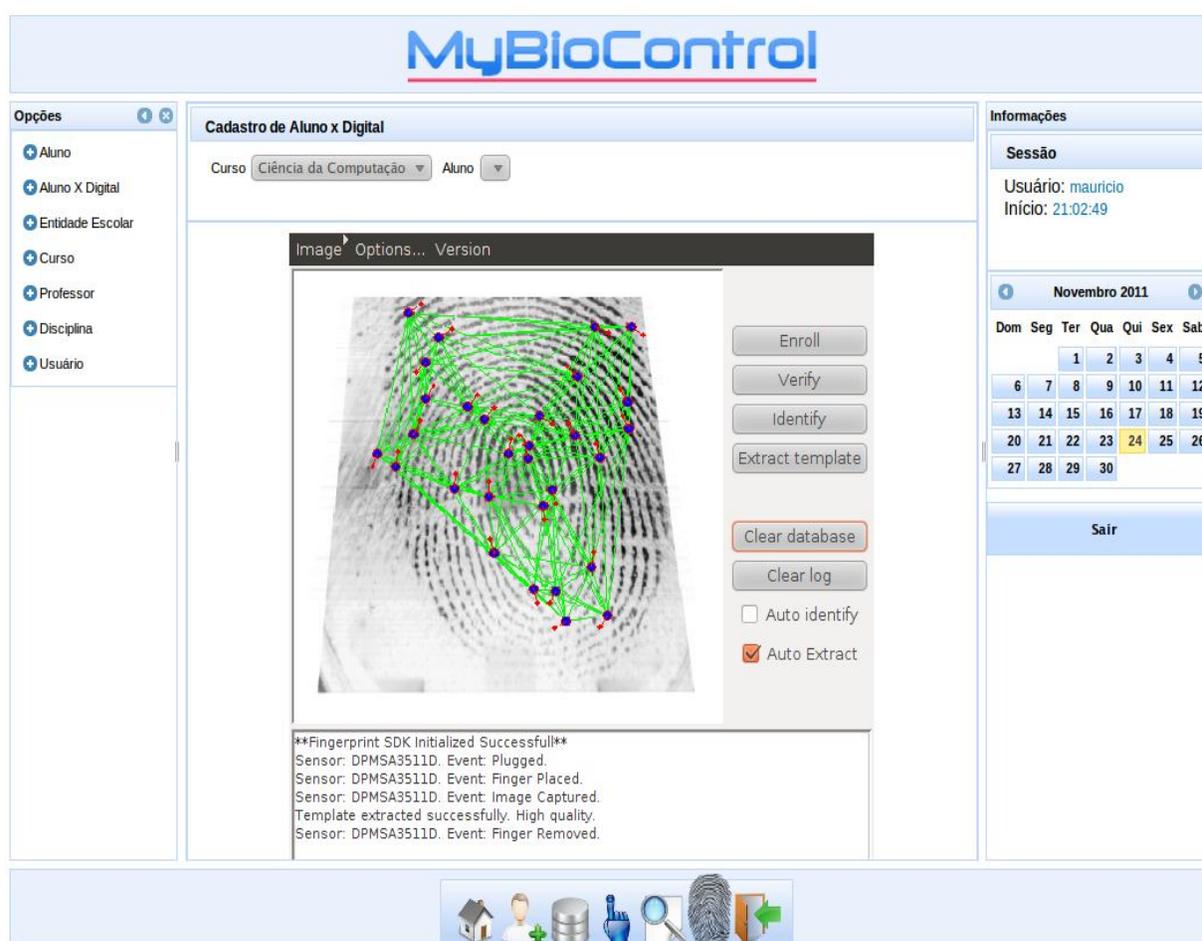
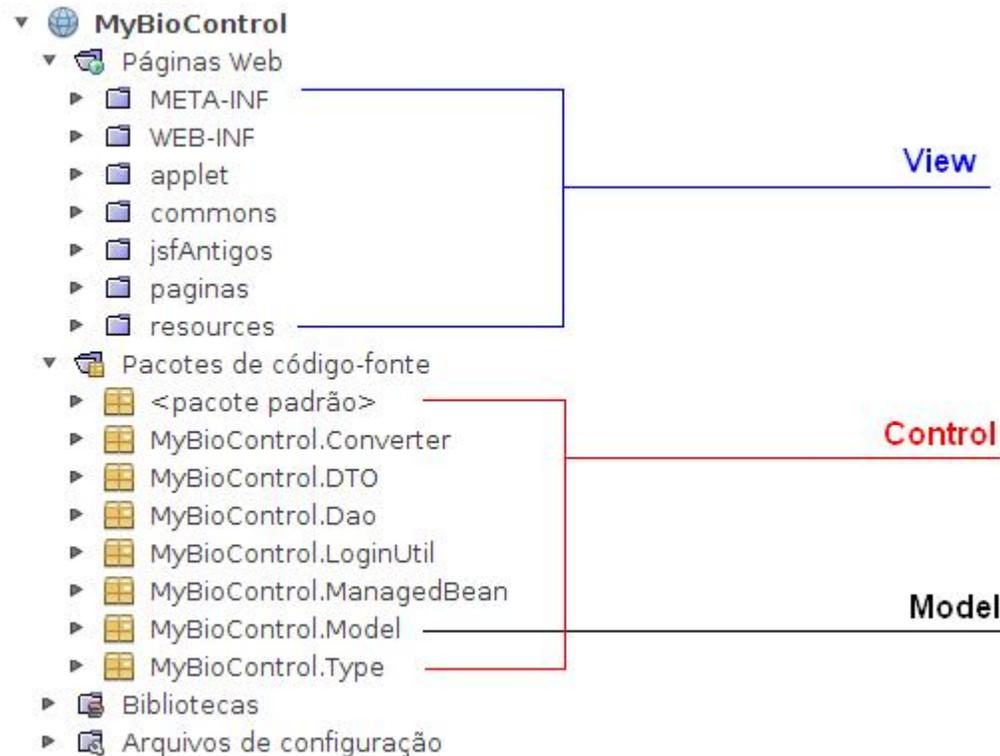


Figura 12 – Reconhecimento da impressão Digital.

## 6.1 – Padrão MVC.

O projeto foi desenvolvido utilizando padrão de desenvolvimento MVC(Model View Control). A figura 14 ilustra a estrutura dos arquivos(classe, xhtml, css,xml)



**Figura 13 – Estrutura do Padrão MVC (Model View Control)**

O projeto foi desenvolvido utilizando o padrão MVC, ele é dividido em 3 camadas, onde o Model são as classes que deverão persistir na base de dados, o View é a interface gráfica por onde o usuário interage com o sistema, e o Control é o intermediário entre a View e o Model, ele controla as requisições e respostas devidamente realizadas pelo usuário.

## 6.2 – Cadastros.

O sistema contém toda a parte cadastral, a figura abaixo ilustra as classes que foram devidamente mapeadas para persistirem na base de dados.

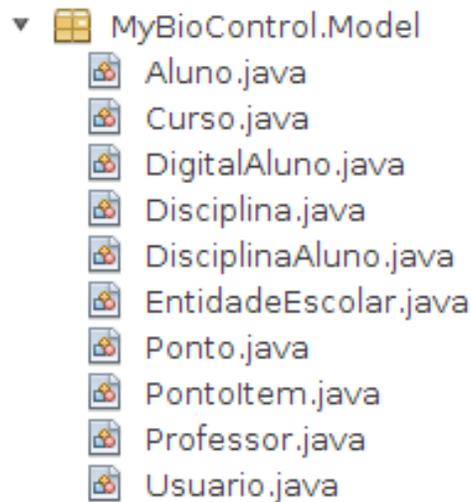


Figura 14 – Classes Mapeadas Para Persistirem na Base de Dados

- **Classe Aluno:** A classe Aluno possui os dados pessoais e acadêmicos do aluno, sua chave primária aluno\_id e um relacionamento com a classe curso, para que no cadastro de aluno o usuário digite o curso cujo o aluno esta matriculado.
- **Classe Curso :** A classe Curso Possui apenas o nome do Curso, sua chave primária curso\_id e um relacionamento com a Classe EntidadeEscolar., para que no cadastro de curso seja informado a Entidade Escolar para o curso cadastrado.
- **Classe Digital Aluno:** A classe digital Aluno possui apenas 3 campos, a sua chave primária digital\_aluno\_id, sua chave estrangeira aluno\_id e o campo aluno\_template onde será salvo o template extraído do leitor biométrico. Essa

classe é de extrema importância ela relaciona aluno com digital do aluno, é ela quem diz que aquele aluno possui aquela ou aquelas digitais.

- **Classe Disciplina:** A classe Disciplina possui relacionamento com professor e curso além de conter sua chave primária e o campo no\_disciplina para ser descrito o nome da disciplina.
- **Classe DisciplinaAluno :** A classe disciplina aluno possui relacionamento com a classe Aluno e Disciplina, além de possuir sua chave primária.
- **Classe EntidadeEscolar :** Essa classe possui apenas os campos de nome e telefone na Entidade Escolar além de sua chave primária.
- **Classe Ponto :** A classe ponto possui relacionamento com a classe Aluno e Disciplina, também possui campos com os valores de data e hora para ser calculado os pontos do aluno.
- **Classe Pontoltem:** Essa classe possui apenas um campo de hora para salvar a hora do ponto do aluno, possui sua chave primária e um relacionamento com a classe Ponto.
- **Classe Professor:** A classe Professor possui dados do professor como telefone e cpf, além de ter sua chave primária e relacionamento com a classe curso.
- **Classe Usuário :** Essa classe possui apenas login e senha para ser cadastrado usuários para o sistema, além de possuir sua chave primária.

Para demonstrar um mapeamento foi escolhido a classe de Professor como podemos visualizar na figura 15.

```
/**
 *
 * @author mauricio
 */
@Entity
@Table(name="professor")
@SequenceGenerator(name="professor_seq",sequenceName="professor_seq")
public class Professor implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Column(name="professor_id")
    @GeneratedValue(generator="professor_seq")
    private long professorId;

    @Column(name="no_professor")
    private String noProfessor;

    @Column(name="fone_professor")
    private String foneProfessor;

    @Column(name="cpf_professor")
    private String cpfProfessor;

    @ManyToOne
    @JoinColumn(name="curso_id", referencedColumnName="curso_id")
    private Curso curso;
```

**Figura 15 – Classe Professor Mapeada.**

A figura 15 ilustra a classe de professor mapeada, para realizar o mapeamento foi utilizado notações que são referenciadas pelo símbolo @, as notações utilizadas foram:

- **@Entity**: Significa que está classe é um entidade e que seus dados (atributos) deverão persistirem na base de dados.

- **@Table:** Significa que a classe faz referência à uma tabela do banco de dados, no exemplo a referência é para tabela professor.
- **@SequencGenerator:** Esta notação faz referência a sequence criada na base dados para incrementar o registro, essa notação é utilizada em campos do tipo @Id.
- **@Id:** Esta notação significa que o atributo é chave primária (identificador) na tabela.
- **@Column:** Faz a ligação do atributo mapeado com o campo na tabela, no exemplo o campo é professor\_id.
- **@GeneratedValue:** Mapea o atributo para receber o valor da sequence, tornando o campo um auto incremento.
- **@ManyToOne:** Faz referencia a outra classe, essa notação faz com que o atributo possua um relacionamento de muitos para um (N - 1) com outra classe (tabela), no exemplo a classe utilizada é a de Curso o que significa que um curso tem muitos professores.
- **@JoinColumn:** Descreve através de qual campo na tabela será feito o relacionamento, no exemplo o campo utilizado é o curso\_id.

Os cadastros no sistema seguem um padrão, ao selecionar a opção de qualquer cadastro a tela apresentada será a listagem de todos os registros existentes para aquele cadastro conforme figura 16.

**MyBioControl**

**Opções**

- Aluno
- Aluno X Digital
- Entidade Escolar
- Curso
- Professor
- Disciplina
- Usuário

**Lista de Professores**

1

Professor	Curso	Editar	Excluir
Patricia Belin Ribeiro	Ciência da Computação		
Patrick Pedreira	Ciência da Computação		

1

**Incluir**

**Informações**

**Sessão**

Usuário: mauricio  
Início: 20:17:31

Novembro 2011

Dom	Seg	Ter	Qua	Qui	Sex	Sab
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

**Sair**

Icons: Home, User, Database, Hand, Magnifying Glass, Fingerprint, Arrow

**Figura 16 – Cadastro de Professor (Listagem)**

A figura 16 ilustra a listagem do cadastro de professor, todos os cadastros possuem esse padrão, na imagem é possível observarmos o botão de editar e excluir referente ao registro em que o botão se encontra. Abaixo da listagem existe o botão incluir que nos leva para a tela de inclusão de professor conforme mostra a figura 16.

**MyBioControl**

**Opções**

- Aluno
- Aluno X Digital
- Entidade Escolar
- Curso
- Professor
- Disciplina
- Usuário

**Cadastro Professor**

Nome: Anderson Talon

Telefone: (014)3232-3232

Cpf: 321.546.513-21

Curso: Ciência da Computação

Salvar Cancelar

**Informações**

Sessão

Usuário: mauricio  
Início: 20:17:31

Novembro 2011

Dom	Seg	Ter	Qua	Qui	Sex	Sab
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Sair

**Figura 17 - Cadastro de professor (inclusão)**

A figura 17 ilustra a tela de inclusão, com o botão de Salvar e Cancelar, o botão salvar grava os registros digitados na tela, o botão Cancelar anula a digitação e volta para a tela de listagem.

### **6.3 – Ponto (Lançamento Manual, Abertura, Fechamento, Consulta).**

O projeto possui o lançamento de ponto manual, esse item foi desenvolvido como um plano de contingência, a figura 18 ilustra como a tela funciona. Essa tela possui 3 combos, Professor, Disciplina e Aluno, ao selecionar o Professor o combo de disciplina é carregado com todas as disciplinas que o professor possui, ao selecionar a disciplina o combo de aluno é carregado com todos os alunos matriculados naquela disciplina, após selecionar o aluno o usuário lança a data e a hora do ponto, o sistema só lança o ponto se realmente existir ponto aberto para aquele aluno e disciplina na data informada, caso contrário o sistema lança uma

mensagem de aviso informando que não existe ponto aberto para o aluno seleciona na data informada.

The screenshot shows the 'MyBioControl' web application interface. The main content area is titled 'Lançamento de Horário' (Time Release) and contains the following fields:

- Professor: Patricia Belin Ribeiro
- Disciplina: PIS - Processamento de Imagens e Sinais
- Aluno: Mauricio Gallis Sahão
- Data do Ponto: 15/11/2011
- Hora Inicial: 19:20

At the bottom of the form are 'Salvar' (Save) and 'Cancelar' (Cancel) buttons. On the left side, there is a sidebar with 'Opções' (Options) including: Gerar Ponto, Lançar Horário, Consulta de Ponto, and Fechar Ponto. On the right side, there is an 'Informações' (Information) section showing the user 'mauricio' and session start time '21:02:49'. Below this is a calendar for November 2011, with the 24th highlighted. At the bottom of the page, there is a navigation bar with icons for home, user, database, search, fingerprint, and exit.

**Figura 18 – Lançamento de Ponto Manual**

O protótipo possui uma tela de abertura de ponto conforme figura 19, deve ser o primeiro procedimento do professor ao chegar na sala de aula, a tela possui um combo de professor e de disciplina, ao selecionar o professor o combo de disciplina carrega as disciplinas cadastradas para o professor selecionado, e no momento em que ele selecionar a disciplina o sistema preenche o horário e a data atual do servidor em que o sistema esta rodando, após os dados informados basta salvar a abertura do ponto.

**MyBioControl**

**Opções**

- Abertura de Ponto
- Lançar Horário
- Consulta de Ponto
- Fechar Ponto

**Abertura de Ponto**

Professor: Patricia Belin Ribeiro

Disciplina: PIS - Processamento de Imagens e Sinais

Data do Ponto: 24/11/2011

Hora Inicial: 21:50

Salvar Cancelar

**Informações**

**Sessão**

Usuário: mauricio  
Início: 21:02:49

Novembro 2011

Dom	Seg	Ter	Qua	Qui	Sex	Sab
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Sair

Home, Add, Database, Hand, Search, Fingerprint, Arrow

**Figura 19 – Abertura de Ponto**

Para realizar o fechamento de ponto o sistema possui outra tela, a tela Fechamento de Ponto conforme figura 20, ela possui um combo de professor e de disciplina, ao selecionar o professor o combo de disciplina carrega as disciplinas cadastradas para o professor selecionado, e no momento em que ele selecionar a disciplina o sistema preenche o horário e a data inicial do ponto que está em aberto para o professor na disciplina selecionada, a tela ainda possui outros 2 campos, tempo de intervalo e hora final, após serem informados o usuário clica em salvar e o ponto é fechado.

**MyBioControl**

**Opções**

- Gerar Ponto
- Lançar Horário
- Consulta de Ponto
- Fechar Ponto

**Fechamento de Ponto**

Professor: Patricia Belin Ribeiro

Disciplina: PIS - Processamento de Imagens e Sinais

Data do Ponto: 17/11/2011

Hora Inicial: 23:21

Tempo de Intervalo:

Hora Final:

Salvar Cancelar

**Informações**

Sessão

Usuário: mauricio  
Início: 21:02:49

Novembro 2011

Dom	Seg	Ter	Qua	Qui	Sex	Sab
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Sair

Home, User, Database, Search, Fingerprint, Exit

**Figura 20 – Fechamento de Ponto**

O projeto também possui uma tela de consulta conforme figura 21, para o professor e o aluno consultarem os pontos registrados, nessa tela o usuário informa a data do ponto, após informar a data do ponto o combo de professor é carregado com todos os professores que possui ponto para aquela data informada, após selecionar o professor o combo de disciplina é carregado com todas as disciplinas cadastradas para o professor informado, após preenchido os campos de parâmetros da consulta basta clicar no botão consultar.

**MyBioControl**

**Opções**

- Gerar Período
- Lançar Horário
- Consulta de Ponto
- Fechar Ponto

**Consulta de Ponto**

Data do Ponto: 17/11/2011 Professor: Patricia Belin Ribeiro Disciplina: PIS - Processamento de Imagens e Sinais

Consultar Cancelar

Nome	Entrada	Saída Intervalo	Entrada Intervalo	Saída
Mauricio Gallis Sáhão	19:00	19:15	19:30	

**Informações**

**Sessão**

Usuário: mauricio  
Início: 23:04:56

**Novembro 2011**

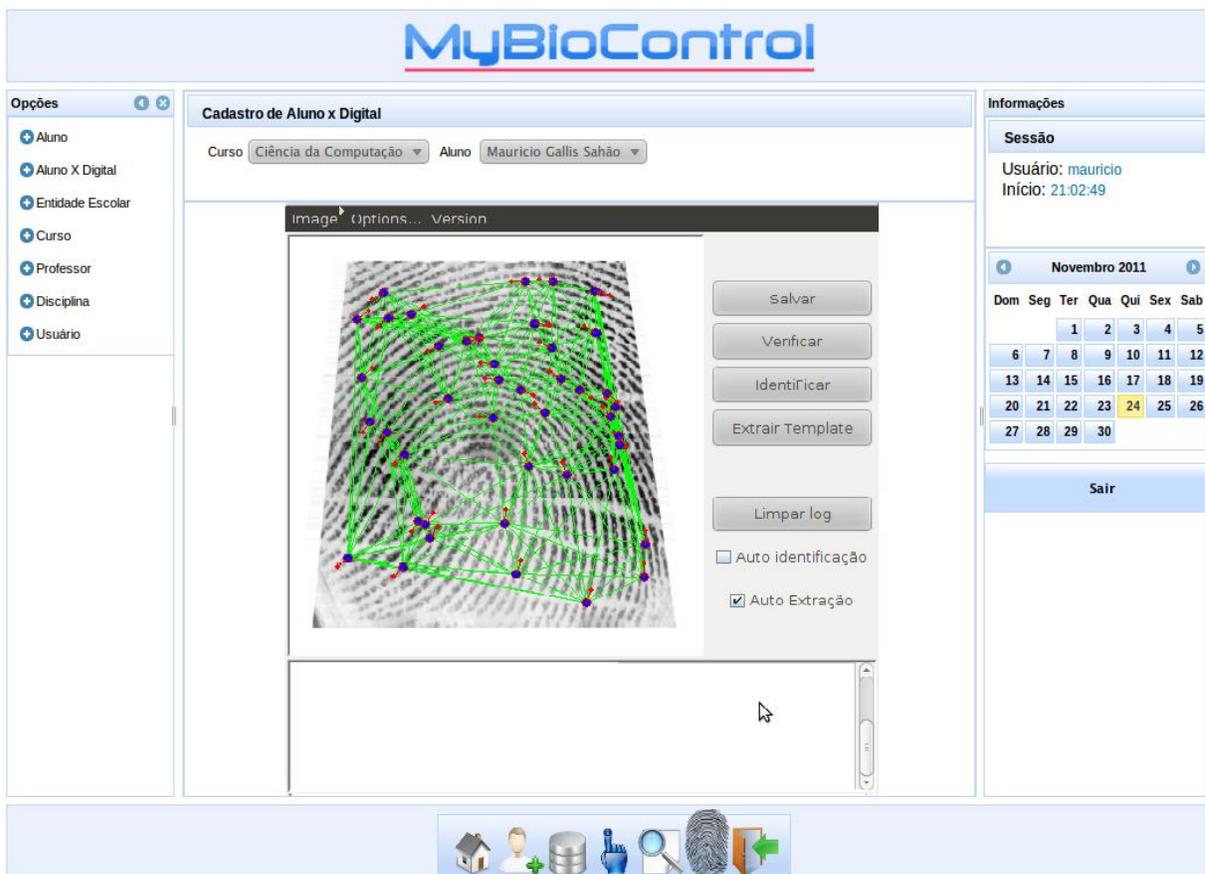
Dom	Seg	Ter	Qua	Qui	Sex	Sab
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Sair

Figura 21 – Consulta de Ponto

#### 6.4 – Cadastro de digital e lançamento de ponto através do reconhecimento.

O projeto possui um cadastro da impressão digital que é capturada através de um leitor biométrico, o nome do cadastro é Aluno Digital, ele relaciona um aluno à uma ou mais digitais, foi utilizado uma applet para interagir com o protótipo, essa applet utiliza o Fingerprint SDK, a applet foi completamente modificada para se comunicar com o projeto, nesta tela o usuário seleciona o curso em um combo, ao selecionar o curso o combo de aluno é carregado com todos os alunos daquele curso, após selecionar o aluno basta colocar o dedo do aluno no leitor e clicar no botão salvar da applet, que a própria applet salvará na base de dados o identificador do aluno e seu template. Conforme figura 22.



**Figura 22 – Cadastro Aluno Digital**

Para o lançamento de ponto através da digital o sistema possui a tela de Lançamento de ponto (figura 23), ela possui um campo com a data do ponto, neste campo deverá ser informada a data atual, o sistema compara com a data do servidor e permite o lançamento do ponto se as datas forem iguais, após informar a data o combo de professores é carregado com todos os professores que possuem pontos abertos para a data informada, após informar o professor o combo de disciplina é carregado com as disciplinas cadastradas para o professor, após selecionar a disciplina o aluno deverá colocar o dedo no leitor, a applet vai capturar a imagem, então basta clicar no botão verificar para que a applet reconheça o aluno e salve o ponto para ele na hora atual do servidor, para a data e disciplina selecionada.



Figura 23 – Lançamento de Ponto

## 6.5 – Banco de Dados.

O sistema contém um total de 10 tabelas, a figura 24 ilustra o diagrama da base de dados, seus respectivos relacionamentos, suas chaves primárias (PK) e suas chaves estrangeiras (FK).

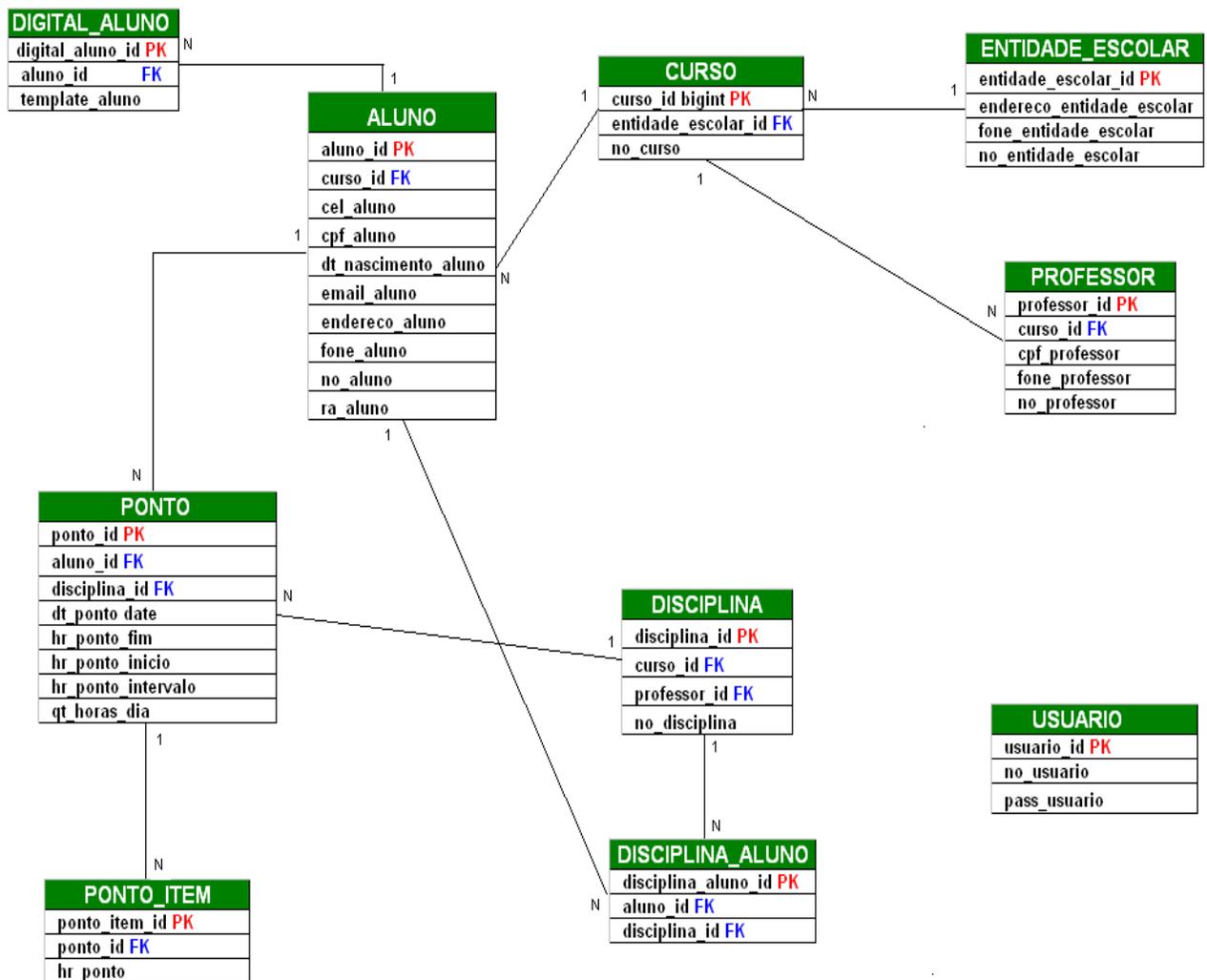


Figura 24 – Diagrama Banco de Dados.

Para os processos relacionados à banco de dados como, consultas (select), inserções (insert), atualizações (update), exclusões (delete), foi utilizado o Hibernate, para a configuração do Hibernate é necessário mapear as entidades em um arquivo xml denominado hibernate.cfg.xml, a figura 25 ilustra esse arquivo de configuração, nele é inserido todas as classe do Model, e as configurações para a conexão com o banco, como senha, usuário, dialeto, driver, url de conexão, e um parâmetro para que seja mostrado o sql que o framework realiza, no log do servidor de aplicação (Tomcat).

```

<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
    <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
    <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/mbc</property>
    <property name="hibernate.connection.username">postgres</property>
    <property name="hibernate.connection.password">postgres</property>
    <property name="hibernate.show_sql">>true</property>

    <mapping class="MyBioControl.Model.Aluno" />
    <mapping class="MyBioControl.ModelCurso" />
    <mapping class="MyBioControl.Model.EntidadeEscolar" />
    <mapping class="MyBioControl.Model.Disciplina" />
    <mapping class="MyBioControl.Model.Professor" />
    <mapping class="MyBioControl.Model.Disciplinaluno" />
    <mapping class="MyBioControl.Model.Usuario" />
    <mapping class="MyBioControl.Model.Ponto" />
    <mapping class="MyBioControl.Model.PontoItem" />
    <mapping class="MyBioControl.Model.DigitalAluno" />

  </session-factory>
</hibernate-configuration>

```

Figura 25 – Arquivo de configuração Hibernate.

## 6.6 – Geração de Relatório.

O projeto possui a opção para gerar relatório, esse relatório é pesquisado por período e disciplina, onde o sistema calcula todas as horas aulas que foram realizadas no período informado e calcula o tempo em que o aluno esteve presente em sala de aula, o relatório informa em porcentagem qual foi o tempo em que o aluno esteve presente. A figura 26 ilustra a impressão do relatório.

MyBioControl		Resumo de Ponto			
Professor:	Patricia Belin Ribeiro	Curso: Ciência da Computação			
Disciplina:	Redes Neurais	Período Pesquisado: 19/12/2011 à 19/12/2011			
Aluno	Qtd. Hr. Presente	Tot. Hr. Aula	%	Situação	
Mauricio Polegar Direito	01:50	01:51	98%	APROVADO	
Mauricio Polegar Esquerdo	01:31	01:51	81%	APROVADO	
Mauricio Indicador Esquerdo	00:49	01:51	43%	REPROVADO	
Mauricio Indicador Direito	00:00	01:51	0%	REPROVADO	

Figura 26 – Impressão de Relatório

## 7 – Conclusão

Podemos nos referir a Biometria como uma senha única, uma senha que não pode ser compartilhada com outra pessoa, uma senha que nunca será perdida, concluímos assim que a Biometria é uma das tecnologias de reconhecimento mais eficaz disponível na Ciência. E não seria diferente com a impressão digital, trata-se do método mais utilizado no mercado atualmente, oferece eficiência no reconhecimento de indivíduos de forma rápida e precisa.

As soluções encontradas para implementar a biometria tem o seu custo, a grande parte das tecnologias disponíveis no mercado são pagas, a implementação da biometria digital também é bastante custosa no quesito técnico, trata-se de um processo longo e requer um grande tempo de desenvolvimento. Porém o custo para implementação da biometria digital é recompensado pela segurança e eficiência que a tecnologia oferece.

O protótipo desenvolvido poderá ser muito útil para facilitar o controle de presença dos alunos na sala de aula, agilizando o processo que normalmente é feito de forma manuscrita. Ele também contorna várias ineficiências que podem ocorrer com o lançamento manuscrito, como o atraso do aluno no início da aula, o atraso do retorno do aluno na volta do intervalo de aula, a saída do aluno antes do término da aula.

Em projetos futuros, será instalado o software em um computador móvel para ser realizados testes do sistema em uma sala de aula, onde será verificado a viabilidade do software, a confiabilidade e seu desempenho.

**REFERÊNCIAS**

ARAUJO, Clemil José. AFIS : Sistemas Automáticos de Impressões Digitais, Brasília – DF, 2000. Disponível em : <<http://www.papiloscopistas.org/afis.html>>. Acessado em 20 de maio de 2011.

CARRIÇO PROENÇA, Hugo Pedro Martins. Towards Non-Cooperative Biometric Iris Recognition, University of Beira Interior, 2006.

CESTA, André Augusto. A linguagem de Programa JAVA. Campinas: Unicamp, 1996.

DA SILVA TEIXEIRA, Raoni Florentino. Análise da Qualidade e melhoramento de Imagens: um estudo de caso com Impressões Digitais. Campinas: Universidade Estadual de Campinas, 2011.

ELMASRI, Ramez. Sistemas de Banco de Dados: 4<sup>o</sup> Edição, São Paulo: Pearson, 2010.

GRIAULE BIOMETRICS, Fingerprint SDK – 2008. Disponível em: <[http://www.griaulebiometrics.com/page/pt-br/fingerprint\\_sdk/overview](http://www.griaulebiometrics.com/page/pt-br/fingerprint_sdk/overview)> Acessado em 10/11/2011

HEUSER, Carlos Alberto. Projeto de Banco de Dados. – 6. Ed. – Porto Alegre: Bookman, 2010

LINHARES, Mauricio. Introdução ao Hibernate 3. GUJ, 2008. Disponível em: <[http://www.guj.com.br/content/articles/hibernate/intruducacao\\_hibernate3\\_guj.pdf](http://www.guj.com.br/content/articles/hibernate/intruducacao_hibernate3_guj.pdf)>. Acessado em 30 de abril de 2011.

MAGALHÃES Eder, NUNES Vinny, Java Magazine 78 – São Paulo, Disponível em : <[http://www.devmedia.com.br/websys.3/webreader.asp?cat=6&revista=javamagazine\\_78#a-2419](http://www.devmedia.com.br/websys.3/webreader.asp?cat=6&revista=javamagazine_78#a-2419)>. Acessado em 02/11/2011.

MEDEIROS, Ernani Sales de. Desenvolvendo Software com UML 2.0: definitivo. São Paulo: Pearson Makron Books, 2004.

PRIME TECHNOLOGY, Framework Primefaces – 2010. Disponível em: <<http://www.primefaces.org/whyprimefaces.html>> Acessado em 28/10/2011.

POSTGRESQLBR, Centro de informações da comunidade brasileira, PSQlBR – 2007. Disponível em: <<http://www.timbira.com/tmp/pgsql-a4-flyer-pt+en.pdf>>. Acessado em 15 de maio de 2011.

QUITES REIS, Rodrigo. Usando UML para Especificações de Sistemas Orientados a Objetos, Universidade Federal do Rio Grande do Sul, 2003.

R. COSTA, Luciano, R. OBELHEIRO, Rafael, S. FRAGA, Joni. Capítulo 3: Introdução à Biometria, Universidade Federal de Santa Catarina: 2006.

THE APACHE SOFTWARE FOUNDATION – Apache Tomcat – 2011. Disponível em: <<http://tomcat.apache.org/>> Acessado em 25/10/2011