

**AUGUSTO TAKAHIRO KIRAMOTO
GABRIEL RIBEIRO VARGAS DUARTE
STIVERSON STOPA ASSIS PALMA**

**SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA
POR PLATAFORMA HETEROGÊNEA
(J2EE E .NET)**

Projeto de Formatura apresentado à
disciplina PCS 2050 - Laboratório de
Projeto de Formatura II, da Escola
Politécnica da Universidade de São Paulo

Orientador:
Prof. Jorge Rady de Almeida Júnior

**São Paulo
2004**

**AUGUSTO TAKAHIRO KIRAMOTO
GABRIEL RIBEIRO VARGAS DUARTE
STIVERSON STOPA ASSIS PALMA**

**SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA
POR PLATAFORMA HETEROGÊNEA
(J2EE E .NET)**

Projeto de Formatura apresentado à
disciplina PCS 2050 - Laboratório de
Projeto de Formatura II, da Escola
Politécnica da Universidade de São Paulo

Área de Concentração:
Engenharia de Computação

Orientador:
Prof. Jorge Rady de Almeida Júnior

**São Paulo
2004**

FICHA CATALOGRÁFICA

Kiramoto, Augusto Takahiro
Duarte, Gabriel Ribeiro Vargas
Palma, Stiverson Stopa Assis

**Solução de Identificação Biométrica Gerenciada por
Plataforma Heterogênea (J2EE E .NET). São Paulo, 2004.**

**Projeto de Formatura apresentado à disciplina PCS 2050 -
Laboratório de Projeto de Formatura II, da Escola Politécnica da
Universidade de São Paulo. Departamento de Engenharia de
Computação.**

Aos meus pais, irmãos e à Cristina Ishii, grandes incentivadores da minha vida, sempre apoiando e vibrando a cada vitória.

Augusto Takahiro Kiramoto

Aos meus pais, irmãos, parentes e amigos que sempre foram grandes incentivadores.

Gabriel Ribeiro Vargas Duarte

Aos meus pais pelo grande incentivo a sempre enfrentar novas e grandes batalhas, à minha tia Idalina que me acolheu no momento em que mais precisei de ajuda, à minha namorada Danila que sempre me estimulou quando eu parecia desanimar e aos meus irmãos e amigos que se felicitam e se orgulham de meu feitos.

Stiverson Stopa Assis Palma

AGRADECIMENTOS

Agradecemos à todos os professores do Departamento de Engenharia de Computação da Escola Politécnica da Universidade de São Paulo pela paciência dispensada e confiança investida nos alunos do Curso Cooperativo. Em especial, agradecemos ao Professor Doutor Jorge Rady de Almeida Júnior pelo constante apoio e motivação como orientador do grupo, ao Professor Doutor Jorge Risco Becerra pelos conselhos técnicos e fornecimento de parte da infra-estrutura necessária para o projeto e ao amigo João Paulo Teixeira dos Santos pela sua dedicada participação.

RESUMO

O presente trabalho tem como principal objetivo eliminar a utilização de comandas (cartões de controle de consumo) e agilizar os processos operacionais de casas noturnas, restaurantes e bares através da utilização de um dispositivo handheld com leitor biométrico. O cliente é recepcionado por um funcionário que realiza seu cadastro no sistema através de dispositivos biométricos de reconhecimento de impressões digitais. A partir desse momento, o cliente já possui uma identidade única, um passaporte inviolável para consumir todos os serviços do estabelecimento por meio de um atendente portando o handheld acoplado ao leitor de digitais. O simples fato de se carregar uma comanda e ter que apresentá-la todas as vezes que o serviço da casa for utilizado, torna essa atividade repetitiva e incômoda. Além disso, há a constrangedora situação de perda do cartão, que implica o pagamento de multas exorbitantes. Dessa maneira, ao contrário dos estabelecimentos atuais, não há extravio de cartões. No fechamento de sua conta, o cliente novamente pressiona o identificador e o total a ser pago é apresentado. O trabalho também contempla a integração com o sistema de estoque, CRM e administração do estabelecimento.

ABSTRACT

The main goal of this solution is to eliminate a personal/consumption card and improve order processes in nightclubs, restaurants, bars and pubs. This goal is going to be reached using a handheld, which is assembled with a fingerprint device. At the entrance, clients will be registered in the system using their fingerprints. After that, they will have a personal unique identification, which will allow them to order any beverage or dish to a waiter using the wireless device. Using this solution, clients will not have to carry personal/consumption cards in the place, and consequently they will not be worried if they would lose the card and pay for it. When clients decide to go away, they will have to pass their fingers on the biometric device in order to calculate how much they have to pay. The solution also offers a supply system interface, CRM features and administrative tools.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ABREVIATURA E SIGLAS

1. INTRODUÇÃO	13
1.1 Objetivo.....	13
1.2 Motivação.....	14
2. ASPECTOS CONCEITUAIS	15
2.1 Web Services.....	16
2.2 J2EE	18
2.3 .NET	20
2.4 Biometria.....	21
2.5 Wi-Fi padrão IEEE 802.11b.....	22
3. ESPECIFICAÇÃO DO PROJETO DE FORMATURA.....	23
3.1 Especificações Funcionais.....	23
3.2 Modelagem de Projeto	26
3.2.1 Diagramas de Casos de Uso.....	26
3.2.2 Diagrama de Classes	27
3.2.3 Diagrama de Entidade-Relacionamento	30
3.3 Arquitetura de Software.....	32
3.3.1 IBM WebSphere Application Server 5.1	33
3.3.2 IBM DB2 Universal Database 8.1	33
3.3.3 IBM WebSphere Studio Application Developer 5.1.1	33
3.3.4 Microsoft Visual Studio .NET	33
3.3.5 Microsoft Internet Information Services	34
3.4 Arquitetura de Hardware	34
3.4.1 HP iPAQ h5550	35
3.4.2 Linksys BEFW11S4 Wireless BroadBand Router	35
3.4.3 Servidor de Aplicações / Servidor de Serviços	35
4. METODOLOGIA	36
5. PROJETO E IMPLEMENTAÇÃO	39
5.1 Banco de Dados	39
5.2 Biometria.....	39
5.2.1 A tecnologia biométrica	40
5.2.2 BIRs e Templates	42
5.2.3 O Modelo da API.....	43
5.2.4 FAR and FRR.....	44
5.2.5 Implementação	45
5.3 Servidor de Serviços.....	47
5.4 Servidor de Aplicações.....	56
5.5 Aplicação Móvel	59
6. TESTES E AVALIAÇÃO	63
7. CONSIDERAÇÕES FINAIS	65

LISTA DE REFERÊNCIAS

APÊNDICE I - CD

ANEXO A - DOCUMENTO INTEGRADO DE CASOS DE USO

LISTA DE FIGURAS

Figura 1 - Web Services	17
Figura 2 - Arquitetura J2EE.....	19
Figura 3 - Integração entre Soluções.....	20
Figura 4 - Dispositivos Biométricos Óticos	21
Figura 5 - Dispositivo Biométrico não-ótico	22
Figura 6 - Diagrama de Classes de Administração.....	27
Figura 7 - Diagrama de Classes de Negócios	29
Figura 8 - Diagrama de Entidade-Relacionamento	31
Figura 9 - Arquitetura de Software.....	32
Figura 10 - Arquitetura de Hardware.....	34
Figura 11 - Modelo Espiral	37
Figura 12 - Estratégias de Implementação	41
Figura 13 - BIR - Biometric Identification Record.....	42
Figura 14 - Níveis de Segurança da BioAPI.....	44
Figura 15 - JNI - Java Native Language.....	46
Figura 16 - WSAD Entity Beans	49
Figura 17 - Session Façade	51
Figura 18 - WSAD NoCardFacadeEJB	52
Figura 19 - Padrão MVC.....	58
Figura 20 - Login e Apresentação da Aplicação Móvel.....	60
Figura 21 - Entrada no Estabelecimento	61
Figura 22 - Cadastro de Pedidos.....	62

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface

BIR - Bimetric Identification Record

BLOB - Binary Large Object

BSP - Biometric Service Provider

CLR - Common Language Runtime

CMP - Container Managed Persistence

CRM - Customer Relationship Management

DHCP - Dynamic Host Configuration Protocol

DTO - Data Transfer Object

DLL - Dynamic Library Link

EJB - Enterprise Java Beans

FAR - False Accept Rate

FRR - False Reject Rate

GUI - Graphic User Interface

HTML - Hyper Text Markup Language

IBIA - International Biometric Industry Association

IEEE - Institute of Electrical and Electronics Engineers

J2EE - Java 2 Enterprise Edition

JNI - Java Native Interface

JSP - Java Server Pages

JVM - Java Virtual Machine

LAN - Local Area Network

MVC - Model View Controller

SOA - Service Oriented Architecture

SOAP - Simple Object Access Protocol

SQL - Structured Query Language

UDDI - Universal Description, Discovery and Integration

UML - Unified Modeling Language

WAS - WebSphere Application Server

WEP - Wired Equivalent Privacy

WI-FI - Wireless Fidelity

WLAN - Wireless Local Area Network

WSAD - WebSphere Studio Application Developer

XML - eXtended Markup Language

1. INTRODUÇÃO

1.1 Objetivo

O projeto propõe uma melhoria do conforto de clientes em bares, restaurantes e casas noturnas através da substituição do uso de comandas (cartões de controle de consumo) por um sistema de reconhecimento biométrico. Outro propósito dessa solução é agilizar os processos operacionais desses estabelecimentos.

Basicamente, o cliente é recepcionado por um funcionário que realiza seu cadastro no sistema através de dispositivos biométricos de reconhecimento de digitais. A partir desse momento, o cliente já possui uma identidade única, um passaporte inviolável para consumir todos os serviços durante sua permanência no estabelecimento. O simples fato de carregar uma comanda e ter que apresentá-la todas as vezes que o serviço da casa for utilizado, torna essa atividade repetitiva e incômoda. Além disso, há a constrangedora situação de perda do cartão, que implica o pagamento de multas exorbitantes. Dessa maneira, ao contrário dos estabelecimentos atuais, não há extravio de cartões.

Utilizando sua impressão digital, o cliente faz pedidos através dos dispositivos móveis (handhelds) com leitores biométricos, os quais são utilizados por garçons e estão conectados à rede local por um access point instalado no estabelecimento. Esse é o principal diferencial dessa solução, uma vez que irá aumentar significativamente o consumo durante o período de funcionamento do estabelecimento em função da agilidade oferecida por essa tecnologia.

Para realizar o fechamento de conta, a identidade do cliente é novamente reconhecida através dos leitores biométricos. Todos os detalhes de consumo e o total a ser pago são apresentados no computador de saída, através do qual é dado baixa do cliente no evento em questão.

Além dos processos para atender o evento realizado, a solução também contempla diferentes módulos para a administração do estabelecimento. Assim, será possível uma total integração com o sistema de estoque (entrada e saída de produtos) e a utilização de um eficiente módulo de CRM para armazenamento das informações dos clientes e envio de mala direta.

1.2 Motivação

A realização do projeto é motivada pela possibilidade de desenvolvimento de um sistema que tenha uma utilidade prática e contenha uma solução inovadora, sem precedências no mercado. Dessa maneira, foi decidido explorar o mercado de eventos, no qual enxergamos uma lacuna em que poderíamos aplicar tecnologias e conhecimentos acadêmicos para melhorias nos procedimentos e processos de negócio.

Em função dessa característica comercial, o projeto possui o “nome fantasia” NoCard System. Para as abordagens acadêmicas usa-se Solução de Identificação Biométrica Gerenciada por Plataforma Heterogênea (J2EE e .NET).

Além da abordagem prática e comercial, o projeto também tem alta motivação tecnológica, uma vez que exige uma vasta utilização de Engenharia de Software e abrange funcionalidades que utilizam biometria, dispositivos móveis e integração entre plataformas heterogêneas (J2EE e .NET).

2. ASPECTOS CONCEITUAIS

A área de concentração deste projeto é a Engenharia de Computação e compreende conceitos técnicos que estão em auge no mundo tecnológico, como identificação biométrica, comunicação wireless e troca de informações entre plataformas heterogêneas J2EE e .NET com utilização do padrão de comunicação Web Services.

A diversidade tecnológica que o escopo do projeto abrange permite-nos conhecer novos conceitos de extrema importância no mercado atual. Somando-se essa motivação técnica com a possível aceitação comercial do produto gerado, cria-se um ambiente de extrema expectativa em relação ao conhecimento adquirido e ao mercado tecnológico.

O tópico de identificação biométrica será explorado para a criação de uma comanda virtual do cliente baseada em sua impressão digital, a qual será introduzida no sistema no momento da entrada do cliente no estabelecimento e que será utilizada nos momentos de consumo de algum produto ou serviço e no momento de fechamento de contas.

A comunicação wireless é utilizada quando acontece a leitura da impressão digital do cliente através de um handheld iPAQ 5550 com leitor biométrico de digitais. Após a digitalização da leitura, as informações são recebidas pelo sistema através de um access point utilizando protocolo 802.11b.

A plataforma J2EE suporta o servidor de serviços do sistema, o qual contém todas as regras de negócio da solução. Já a plataforma .NET foi utilizada para o desenvolvimento do servidor de aplicações, que faz a camada intermediária entre as regras de negócio e o usuário. Além disso, a tecnologia .NET também foi utilizada para o desenvolvimento da aplicação móvel do handheld. Todos esses módulos são integrados pelas funções de comunicação do Web Services.

Em seguida, tem-se um aprofundamento teórico de cada um dos conceitos e tecnologias utilizados no desenvolvimento do projeto:

2.1 Web Services

Os Web Services são essencialmente um conjunto de funções chamáveis via Internet que usam o protocolo SOAP e têm sido usados por um número crescente de empresas à medida que elas divulgam seus produtos e serviços aos clientes e parceiros comerciais por meio da Internet e de extranets corporativas.

Para maior compreensão do funcionamento dos Web Services é preciso entender a linguagem XML. Essa linguagem tem como função criar documentos bem estruturados e padronizados para facilitar a troca de informações entre aplicações independentes, possibilitando uma melhor comunicação entre diferentes plataformas tecnológicas.

Os Web Services são componentes de negócios provedores de serviços localizados na Internet e que estão disponíveis para serem acessados por outras aplicações, facilitando a integração entre sistemas corporativos e evidenciando a principal característica dessa tecnologia que é a interoperabilidade.

Para que se possa utilizar os Web Services é necessário que se obtenha sua localização. A solução mais utilizada é o UDDI, um serviço de descoberta de Web Services através de pesquisa em um diretório, que é replicado e espelhado. IBM, Microsoft e outras empresas fornecem esse tipo de serviço.

A Figura 1 demonstra resumidamente uma arquitetura de Web Services:

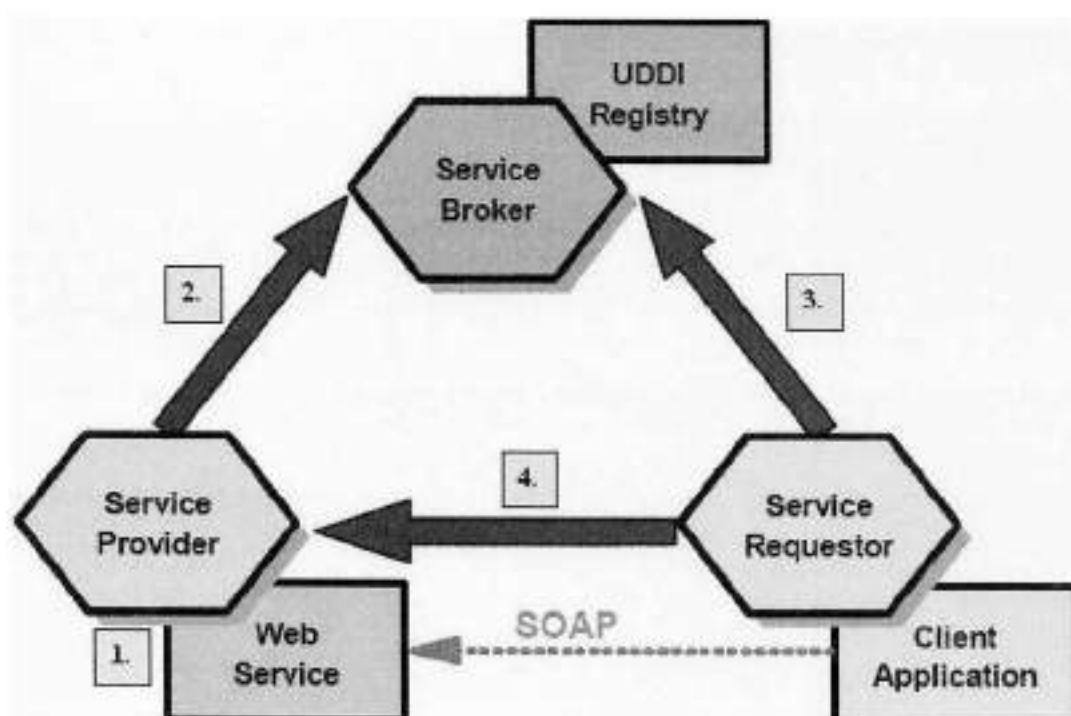


Figura 1 - Web Services

Fonte: IBM Software Group

Em (1) um Web Service é criado e suas interfaces devem ser definidas.

Em (2) o Web Service é publicado em um ou mais UDDI para a pesquisa dos usuários.

Em (3) o Web Service é localizado para ser invocado por usuários potenciais.

Em (4) o Web Service é invocado pelo usuário através do protocolo SOAP.

2.2 J2EE

A Plataforma J2EE é uma extensão da linguagem Java através de padrões para aplicações distribuídas utilizando um modelo baseado em componentes. É um ambiente para desenvolvimento de soluções corporativas que garantem segurança dos dados, troca de mensagens, alto desempenho e robustez.

Essa plataforma cresce a cada momento, ganhando novas APIs, tecnologias, frameworks específicos para camadas de aplicação e design patterns. Toda essa evolução tem o intuito de fornecer maior robustez, estabilidade, segurança e garantia de alto desempenho a sistemas web, desktops, soluções para dispositivos portáteis ou ambientes de integração.

A plataforma J2EE mostra várias camadas, com as seguintes funcionalidades:

- *Cliente*: a camada do cliente interage com o usuário e exibe as informações do sistema para o mesmo. A plataforma J2EE suporta diferentes tipos de clientes, incluindo clientes browser, applets Java embutidos em páginas Web, aplicações Java para desktops e clientes SOAP.
- *Aplicação*: a camada de aplicação pode ser subdividida nas camadas web e de negócios. A subcamada web gera uma lógica de apresentação e aceita as respostas do usuário a partir dos clientes, que são normalmente clientes páginas web, applets, e outros clientes da web. Baseada na solicitação recebida pelo cliente, a subcamada web gera a resposta apropriada para o cliente. É feito o uso do container web que implementa os componentes JSPs e servlets. A subcamada de negócios trata da principal lógica de negócio da aplicação. Ela fornece as interfaces necessárias aos componentes de serviços de negócio subjacentes. Os componentes de negócios são, normalmente, implementados como componentes EJB com suporte a partir de um container EJB, que facilita o ciclo de vida do componente e gerencia a persistência, as transações e a alocação de recursos.
- *Dados*: essa camada é responsável pelo sistema de informação da empresa, incluindo os sistemas de banco de dados, o de processamento de transação, os

sistemas herdados e os sistemas de planejamento de recurso da empresa. A camada de dados é o ponto onde as aplicações J2EE se integram com os sistemas que não são J2EE e os sistemas herdados.

A Figura 2 retrata de forma simplificada a arquitetura da plataforma J2EE.

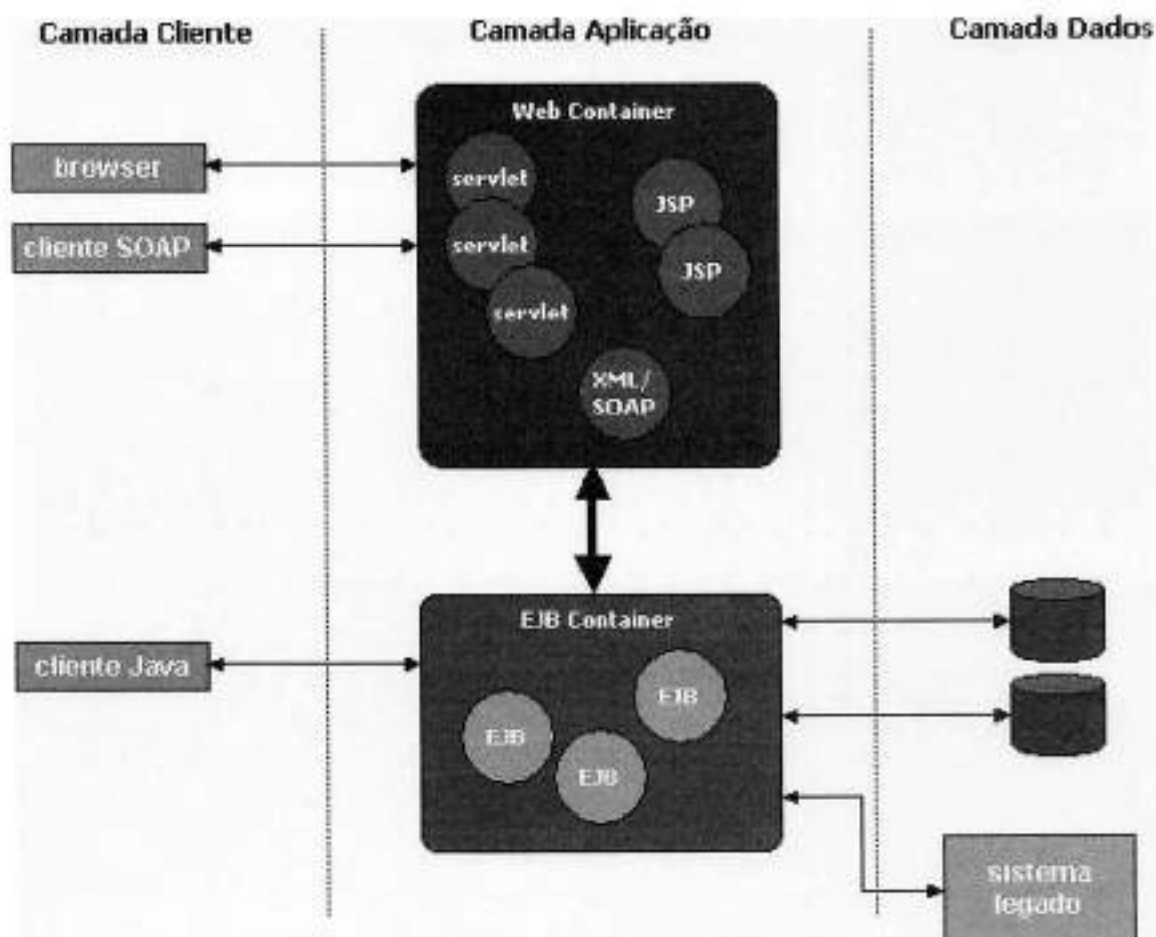


Figura 2 - Arquitetura J2EE.

Fonte: IBM Software Group

2.3 .NET

A plataforma .NET é um ambiente para desenvolvimento de software que traz um novo conceito de desenvolvimento voltado à Internet.

Uma de suas principais características é a independência do sistema operacional, já que o .NET possui sua própria biblioteca de classes, DLLs, funções e outros recursos, a qual é chamada de .NET Framework. Essas bibliotecas são executadas sobre uma linguagem intermediária, o CLR, ficando para o .NET Framework fazer a interpretação para a linguagem de máquina.

Essa independência de sistema operacional qualifica a plataforma .NET como uma arquitetura aberta, a qual permite a integração de outras empresas de software com a Microsoft, proprietária da plataforma. Em função dessa arquitetura, diversas linguagens tem sido portadas para interagir com o .NET como Java, Delphi, Perl, Cobol, Python, Smalltalk, Fortran, etc. Atualmente, mais de 30 tecnologias de desenvolvimento de software integram o ambiente .NET.

A diversidade dessa plataforma permite uma fácil integração entre diferentes soluções através do uso dos Web Services. A Figura 3 apresenta essa idéia.

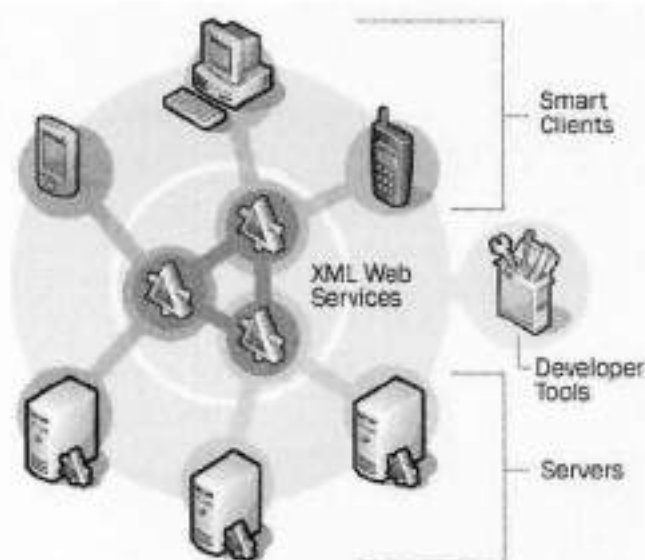


Figura 3 - Integração entre Soluções

Fonte: IBM Software Group

2.4 Biometria

A biometria utilizada no projeto é uma tecnologia de leitura e reconhecimento de impressão digital. No entanto, por trás dessa simples definição, existe um profundo esforço tecnológico e um vasto estudo sobre técnicas de reconhecimento de informações vindas do próprio corpo humano.

As impressões digitais são diferentes em cada pessoa e não mudam com o tempo e, por isso têm confiabilidade elevada. Essa particularidade é de grande interesse para o ramo de segurança e proteção de informações.

As características da impressão digital são extraídas através de um leitor de impressão e são armazenadas na memória do computador para comparações futuras. O sistema do reconhecimento de impressão digital pode ser utilizado para a comparação das impressões digitais como identificação de usuários de um sistema.

A eficiência de um sistema de reconhecimento de impressões digitais está estritamente ligada aos métodos de leitura que, basicamente, são de dois tipos: ótico e não-ótico.

O método ótico é composto de prismas e hologramas. O detector de prismas é um método para obter a impressão digital usando a reflexão total da luz incidindo em prismas. O dispositivo de hologramas obtém a imagem projetada refletida da impressão digital do dedo na placa lisa. A Figura 4 descreve as duas tecnologias:

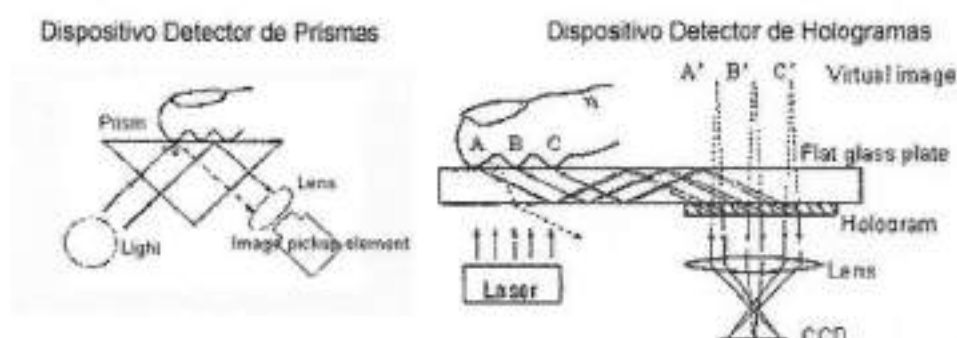


Figura 4 - Dispositivos Biométricos Óticos

Fonte: LINTEC Engenharia

O método não-ótico obtém a impressão digital através da conversão da imagem refletida em sinal elétrico. A Figura 5 expressa a idéia.

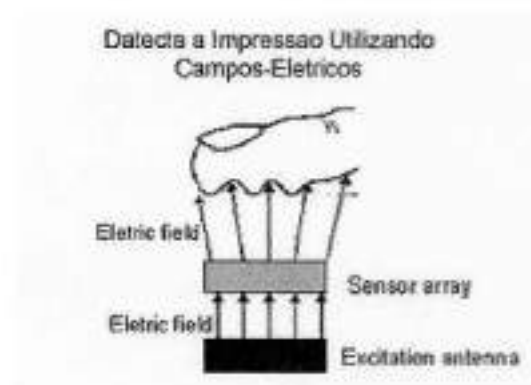


Figura 5 - Dispositivo Biométrico não-ótico

Fonte: LINTEC Engenharia

2.5 Wi-Fi padrão IEEE 802.11b

O Wi-Fi é a tecnologia de rede local sem fio padronizada pelo IEEE 802.11x e formalmente recebe o nome Wireless LAN (WLAN).

Dentre os padrões IEEE do Wi-Fi, aquele que mais se destaca é o 802.11b que opera em 2,4 GHz, possui taxa de transmissão de 11 Mbs e alcance de 100 a 300 metros. Esse é o padrão no qual operam os equipamentos utilizados no projeto.

A principal vantagem da tecnologia de rede sem fios é o dinamismo gerado pela eliminação de cabos de rede, característica que dá maior liberdade ao usuário de se deslocar e ao mesmo tempo acessar a Internet, e-mails e uma rede corporativa.

O Wi-Fi é principalmente aplicado em redes locais internas de escritórios e residências ou em redes públicas de acesso à Internet como em aeroportos, cafés, restaurantes, hotéis e universidades. Esses locais de acesso são conhecidos como hotspots e permitem que usuários com computadores portáteis ou dispositivos handhelds dotados de tecnologia wireless acessem a Internet através de um access point instalado na vizinhança.

3. ESPECIFICAÇÃO DO PROJETO DE FORMATURA

3.1 Especificações Funcionais

O grande foco do projeto é a eliminação do uso de comandas em bares, restaurantes e casas noturnas por um sistema de reconhecimento biométrico. Ele visa o oferecimento de conforto aos clientes, agilidade no atendimento e maior controle dos processos operacionais.

Em um primeiro momento, o cliente tem seu cadastro efetuado no sistema através de dispositivos biométricos de reconhecimento de impressões digitais, fato que o deixa apto a consumir os serviços do estabelecimento. A partir desse momento, o cliente faz pedidos através dos dispositivos móveis (handhelds) com leitores biométricos que são utilizados por garçons. Por fim, quando o cliente deixa o estabelecimento, sua impressão digital é novamente reconhecida e sua conta é fechada.

A solução também contempla diferentes módulos para a administração do estabelecimento, sendo possível uma total integração com o sistema de estoque (entrada e saída de produtos) e a utilização de um eficiente módulo de CRM.

Para atender a todas essas características da solução, as seguintes funcionalidades foram levantadas:

- *Cadastro de Novos Clientes:* o cadastro de novos clientes irá colher informações sobre o cliente do estabelecimento, tais como nome, endereço, data de nascimento.
- *Autenticação no Sistema via Impressão Digital:*
 - **No ato da chegada:** a autenticação no ato da chegada serve para que o cliente, depois de cadastrado no sistema, seja inserido no mesmo e esteja habilitado a consumir dentro do estabelecimento.
 - **No ato da compra:** a autenticação no ato da compra serve para que o consumo do cliente seja cadastrado no sistema e possa ser cobrado no ato de fechamento da conta. Além disso, serve para que o cliente não tenha que possuir uma comanda, onde seriam anotados seus gastos.

- **No ato da saída:** a autenticação no ato da saída serve para identificar a conta do cliente e ter à disposição os gastos do cliente no estabelecimento, e assim cobrá-lo do valor indicado.

- *Forma de Pagamento:* indica qual é a forma de pagamento do cliente, isto é, dinheiro, cheque ou cartão de crédito.
- *Cadastro de Pedidos:* o cadastro de pedidos ocorre quando o cliente pede um determinado item, e após ser reconhecido no sistema através da validação da sua digital, tem o pedido cadastrado na sua conta.
- *Visualização de Subtotal por Cliente:* a visualização do subtotal por cliente ocorre quando um cliente pergunta para um atendente sobre seus gastos no estabelecimento até o momento. Assim, o sistema fornece os gastos do cliente após o reconhecimento de sua impressão digital. A visualização ocorre também no ato do cadastro de um novo pedido para o cliente, fornecendo assim, um subtotal dos gastos até o momento do pedido.
- *Fechamento de Conta por Cliente:* o fechamento de conta ocorre quando o cliente deseja fechar a sua conta, o que é feito após o reconhecimento de sua impressão digital.
- *Cálculo do Total Consumido:* o sistema calcula o total consumido pelo cliente no estabelecimento.
- *Lista de Produtos Consumidos:* o sistema discrimina todos os produtos consumidos pelo cliente.
- *Gerenciamento do Estoque:* funcionalidade que gerencia o estoque do estabelecimento.
- *Baixa de Produto Automaticamente:* ao vender o produto ao consumidor, o sistema irá automaticamente dar baixa no estoque.
- *Consulta e Alerta de Nível de Estoque:* permite a consulta da quantidade de produtos que se encontram em estoque, e o sistema terá um alerta para dizer quando o estoque de determinado produto se encontrar abaixo de determinada quantidade.

- *Cadastro de Produtos de Estoque:* o cadastro de produtos no estoque ocorre quando chega um novo produto no estoque, e cadastra-se informações sobre ele, tais como nome, fabricante, preço, etc.
- *Cadastro de Fornecedores:* o cadastro de fornecedores ocorre quando um novo fornecedor é contratado para fornecer produtos para o estabelecimento.
- *Solicitação de Compra de Produtos de Estoque:* a solicitação de compra de produtos de estoque ocorre quando um funcionário do estabelecimento verifica que um produto está com nível de estoque baixo.
- *Cadastro de Compras de Produtos de Estoque:* o cadastro de compras de produtos de estoque ocorre na compra de produtos para reposição de estoque.
- *Cadastro de Produtos de Venda Final ao Consumidor:* o cadastro de produtos de venda final ao consumidor ocorre quando se deseja acrescentar um novo produto ao cardápio do estabelecimento, e tal produto vai ser diretamente consumido pelo cliente.
- *Fechamento Diário:* o fechamento diário ocorre no fim do horário comercial do estabelecimento.
- *Total de Arrecadação:* o sistema fornece o total da arrecadação feita com vendas de produtos e outras atividades no estabelecimento.
- *Quantidade e Discriminação de Produtos Consumidos:* o sistema fornece a discriminação de produtos consumidos, bem como a quantidade de cada produto que foi consumida.
- *Balanço Mensal:* o balanço mensal ocorre a cada mês, fornecendo:
 - **Total de arrecadação:** o sistema fornece o total da arrecadação feita no estabelecimento durante o mês corrente.
 - **Quantidade e discriminação de produtos consumidos:** o sistema fornece a discriminação de produtos e suas respectivas quantidades consumidas.
- *Contas a pagar:* o sistema possui um módulo que gerencia as contas a pagar do estabelecimento.

- *Relatórios Estatísticos:* o sistema gera relatórios estatísticos dos produtos mais consumidos, participação dos consumidores e faixa etária média dos clientes.
- *CRM:* o sistema implementa diversas funções de CRM, tais como mala direta, hábitos dos clientes, etc.
- *Cadastro de eventos.*
- *Cadastro de operadores.*
- *Administração do sistema.*
- *Perfil de acesso.*
- *Login.*
- *Segurança.*
- *Cadastro de Perfil de Consumo.*

É importante frisar que a abordagem do projeto para eventos não impede a realização de algumas personalizações para que ele atenda a qualquer estabelecimento que controle a entrada de clientes, os processos de atendimento e o fluxo de saída.

3.2 Modelagem de Projeto

Por se tratar de um projeto que envolve desenvolvimento de software, procurou-se dar uma grande ênfase na parte de modelagem através da criação dos principais diagramas UML, que refletiram no modo em que a implementação da solução foi feita.

Assim, baseado nos requisitos funcionais descritos acima, os seguintes diagramas foram produzidos:

3.2.1 Diagramas de Casos de Uso

O Diagrama de Casos de Uso, o qual descreve o relacionamento dos usuários (atores) com as funcionalidades do sistema, é apresentado no documento ANEXO A – Documento Integrado de Casos de Uso.

3.2.2 Diagrama de Classes

O Diagrama de Classes representa o relacionamento entre as futuras entidades do sistema. Ele é apresentado nas Figuras 6 e 7 pelos diagramas Administrativo e de Negócios e suas respectivas classes.

Classes do Diagrama Administrativo:

- *Usuário*: classe que representa o usuário no sistema.
- *Papel*: classe que representa os papéis existentes no sistema, a qual o usuário será associado.
- *Atividade*: classe que representa as atividades que um papel terá acesso no sistema.

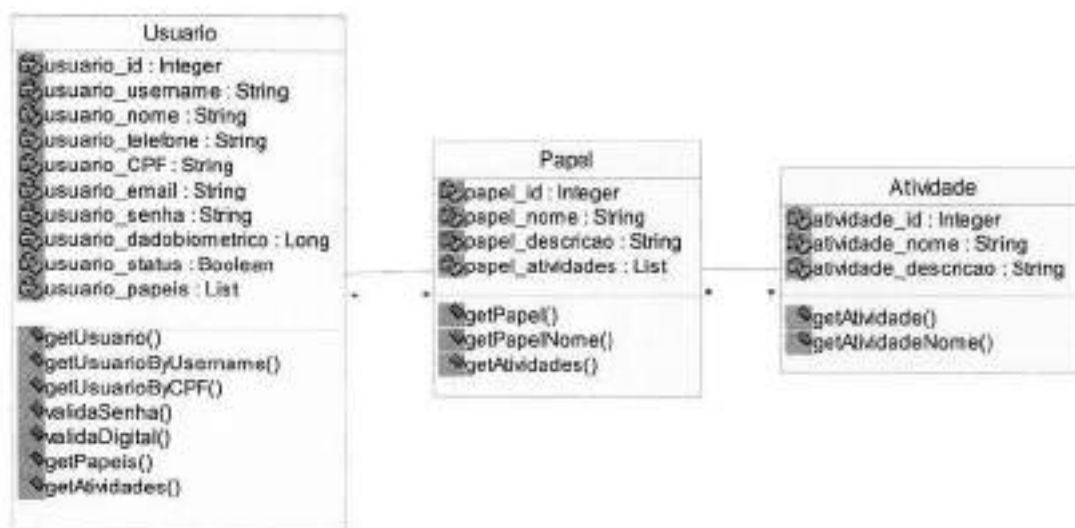


Figura 6 - Diagrama de Classes de Administração

Classes do Diagrama de Negócio:

- *Evento*: classe que representa o evento do dia no sistema.
- *Comanda*: classe que representa a comanda de um cliente, associando o cliente ao evento e aos produtos consumidos.
- *Cliente*: classe que representa o cliente no sistema.
- *ComandaProdutoConsumo*: classe que representa a relação entre produtos consumidos e a comanda de um cliente.

- *ProdutoConsumo*: classe que representa os produtos de consumo direto ao consumidor.
- *Categoria*: classe que representa uma determinada categoria de um produto de consumo ou de estoque (ex: categoria vinhos nacionais).
- *ProdutoConsumoComposicao*: classe que representa a composição de um produto de consumo, a partir de diversos produtos de estoque.
- *ProdutoEstoque*: classe que representa os produtos de estoque armazenados de alguma maneira.
- *Fornecedor*: classe que representa os fornecedores de produtos de estoque ao estabelecimento.
- *Pedido*: classe que representa o pedido de compra de um determinado produto de estoque.
- *ProdutoEstoquePedido*: classe que relaciona um determinado pedido a diversos produtos de estoque.
- *MovimentoEstoque*: classe que registra os movimentos de entrada e saída de produtos de estoque do estoque.
- *MovimentoEntrada*: classe que registra os movimentos de entrada de produtos de estoque do estoque.
- *MovimentoSaida*: classe que registra os movimentos de saída de produtos de estoque do estoque.
- *MensagemRetorno*: classe que contém o domínio de todas as mensagens de retorno do sistema.

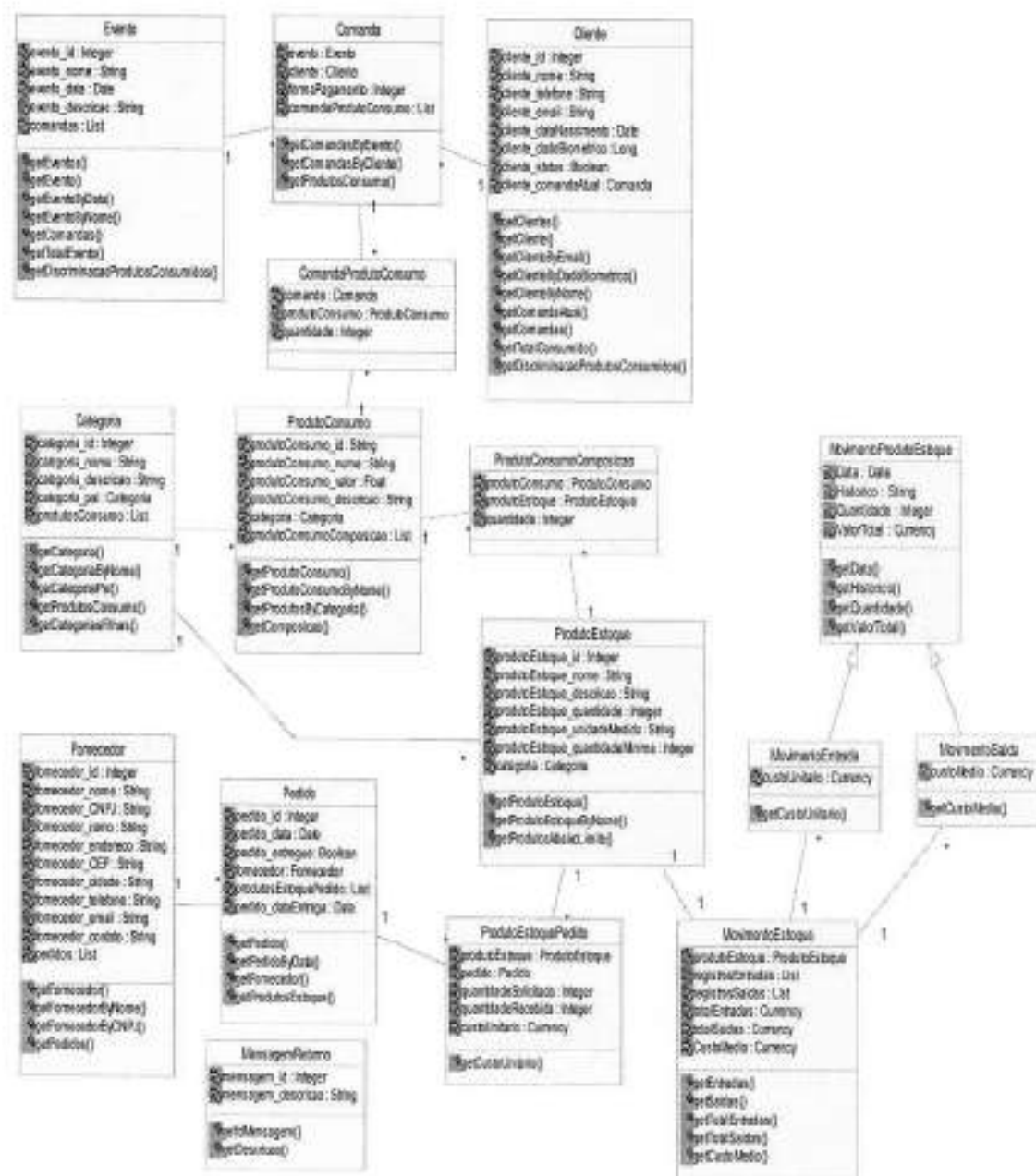


Figura 7 - Diagrama de Classes de Negócios

3.2.3 Diagrama de Entidade-Relacionamento

O Diagrama de Entidade-Relacionamento descreve as entidades de dados do sistema e como essas se relacionam. Para o sistema em questão, as seguintes entidades foram mapeadas:

- *Usuário*: Entidade que armazena o usuário no banco de dados.
- *Papel*: Entidade que armazena os papéis existentes no banco de dados.
- *Atividade*: Entidade que armazena as atividades que um papel terá acesso no banco de dados. Um papel possui diversas atividades.
- *Evento*: Entidade que armazena o evento do dia no banco de dados.
- *Comanda*: Entidade que armazena a comanda de um cliente. A comanda está associada a um evento, um cliente e diversos registros de Comanda_Produto_Consumo.
- *Cliente*: Entidade que armazena o cliente no banco de dados.
- *Comanda_Produto_Consumo*: Entidade que armazena a relação entre produtos consumidos e a comanda de um cliente.
- *Produto_Consumo*: Entidade que armazena os produtos de consumo vendidos diretamente ao consumidor.
- *Categoria*: Entidade que armazena uma determinada categoria de um produto de consumo ou de estoque (ex: categoria vinhos nacionais).
- *Produto_Consumo_Composicao*: Entidade que armazena a composição de um produto de consumo, a partir de diversos produtos de estoque.
- *Produto_Estoque*: Entidade que armazena os produtos de estoque armazenados de alguma maneira.
- *Fornecedor*: Entidade que armazena os fornecedores de produtos de estoque do estabelecimento.
- *Pedido*: Entidade que armazena o pedido de compra de um determinado produto de estoque.
- *Produto_Estoque_Pedido*: Entidade que relaciona um determinado pedido a diversos produtos de estoque.

- *Movimento_Estoque*: Entidade que registra os movimentos de entrada e saída de produtos de estoque do estoque.
- *Tipo_Movimento*: Entidade que registra os tipos de movimento de estoque.
- *Mensagem_Retorno*: Entidade que contém o domínio de todas as mensagens de retorno do banco de dados.

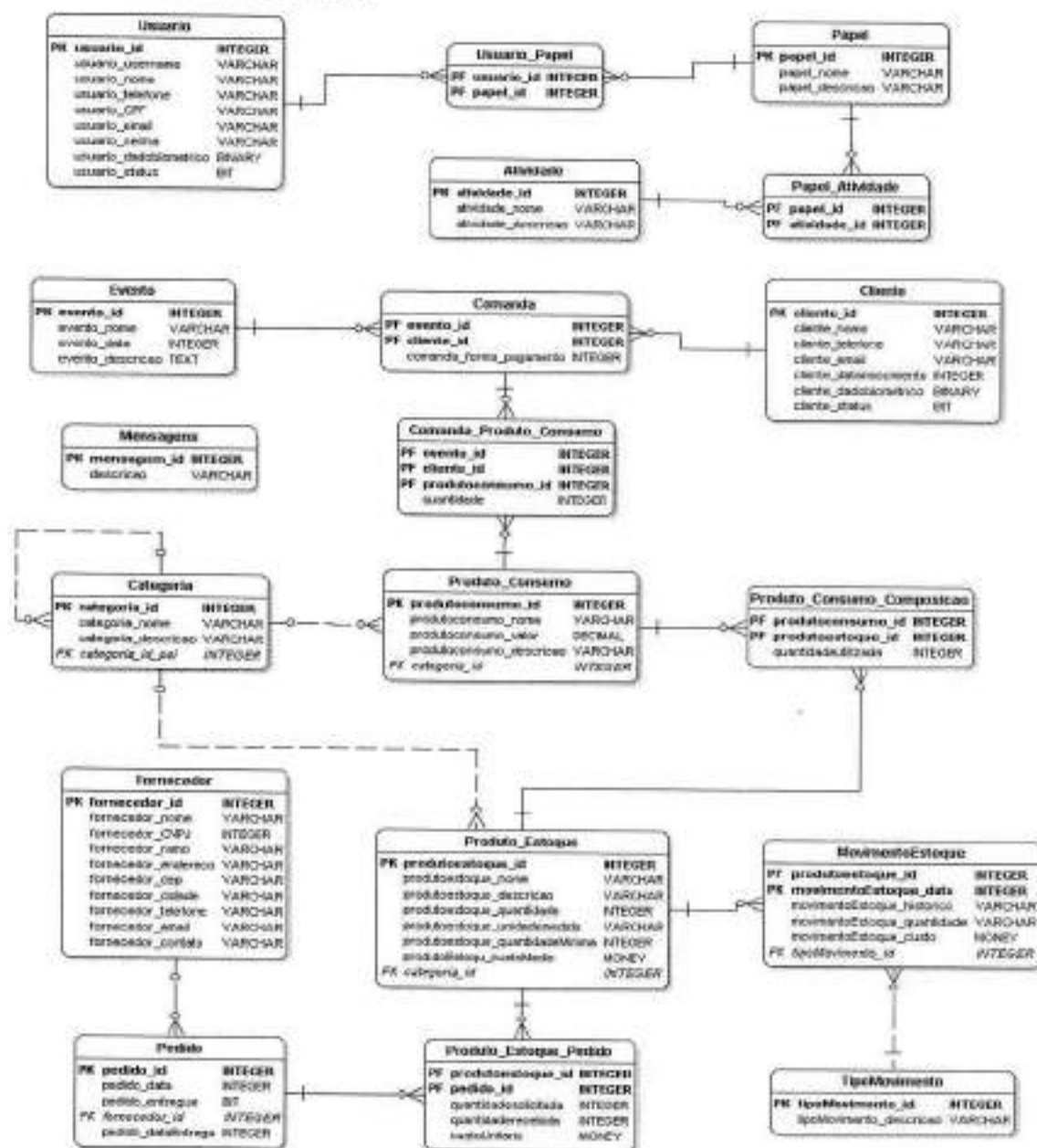


Figura 8 - Diagrama de Entidade-Relacionamento

3.3 Arquitetura de Software

A Arquitetura de Software apresenta o relacionamento entre os módulos de software da solução. Com esse propósito, são apresentados o Servidor de Serviços desenvolvido em plataforma J2EE, o Servidor de Aplicação e a aplicação para o handheld desenvolvidos na plataforma .NET, e a comunicação entre todos os módulos que foi implementada em Web Services.

A Figura 9 apresenta a Arquitetura de Software:

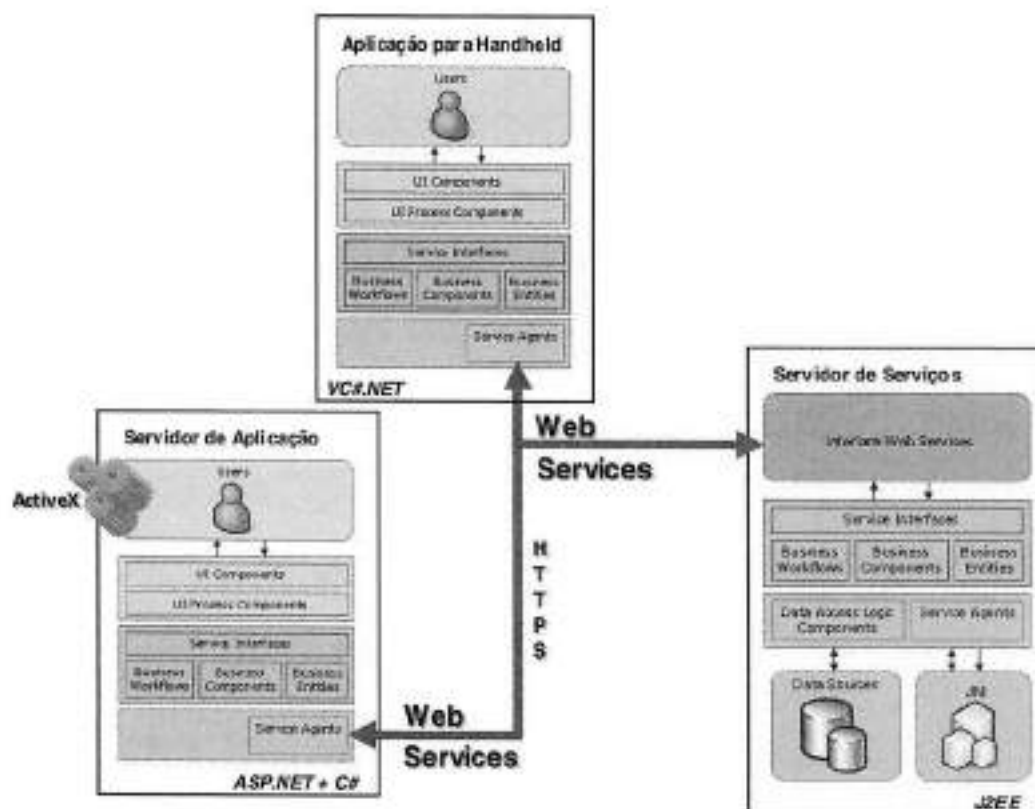


Figura 9 - Arquitetura de Software

Fonte: Microsoft

A Arquitetura de Software foi implementada utilizando as seguintes ferramentas:

3.3.1 IBM WebSphere Application Server 5.1

O produto IBM WAS 5.1 é um Application Server que suporta toda a especificação J2EE 1.3 e possui suporte a Web Services, sendo um middleware orientado à objetos em que as aplicações J2EE podem ser instaladas e disponibilizadas aos usuários.

O produto possui um container utilizado para suportar e disponibilizar aplicações Web dinâmicas utilizando a especificação JSP 2.0. Possui também um container EJB que suporta a especificação EJB 2.0. Assim, suporta aplicações que utilizem a tecnologia JSP, servlets, EJB e Web Services.

Tal ferramenta possui uma grande aceitação pelo mercado, sendo líder no segmento que atua, provendo segurança, performance, disponibilidade, conectividade e escalabilidade para aplicações.

3.3.2 IBM DB2 Universal Database 8.1

O produto IBM DB2 8.1 é o gerenciador de banco de dados relacional da IBM, o qual suporta milhares de transações e terabytes de dados, provendo estabilidade, disponibilidade e segurança para a base de dados.

3.3.3 IBM WebSphere Studio Application Developer 5.1.1

O produto IBM WSAD 5.1.1 é um ambiente integrado de desenvolvimento para visualmente desenhar, construir, testar e implantar Web Services, portais e aplicações J2EE. O WSAD acelera o desenvolvimento de aplicações com um conjunto de ferramentas de alta produtividade, templates e wizards, e utiliza o Java como linguagem de programação.

3.3.4 Microsoft Visual Studio .NET

O produto Visual Studio .NET é um ambiente de desenvolvimento que permite a integração entre aplicações através de XML Web Services. Ele provê uma interface de todas as linguagens de programação, dentre elas Visual Basic .NET, Visual C++ .NET,

Visual C# .NET e Visual J# .NET, sendo usado para desenvolver aplicações Web, aplicações Web Services e aplicações Windows.

Toda as aplicações desenvolvidas pelo Visual Studio .NET para desktops utilizam o .NET Framework 1.1 no ambiente de execução e as aplicações desenvolvidas para handhelds utilizam o .NET Compact Framework 1.1.

3.3.5 Microsoft Internet Information Services

O produto Microsoft IIS é um Web Application Server que provê confiabilidade e disponibilidade, gerenciamento, escalabilidade e segurança, voltados à aplicações que utilizem a plataforma Microsoft .NET.

3.4 Arquitetura de Hardware

A Arquitetura de Hardware apresenta toda a infra-estrutura física necessária para o funcionamento da solução.

A Figura 10 esclarece a Arquitetura de Hardware do sistema:

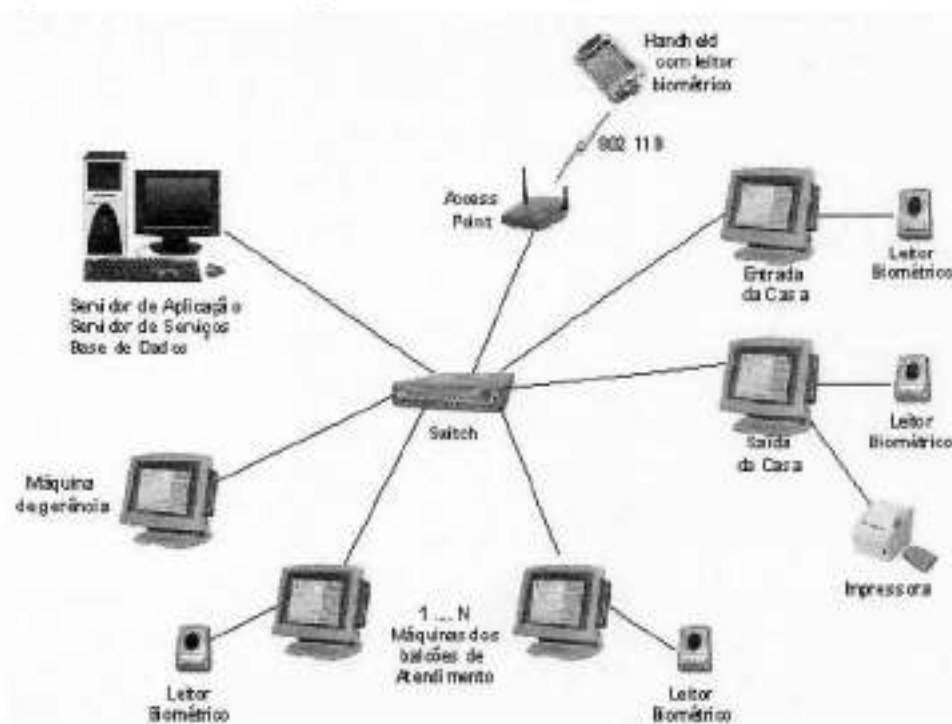


Figura 10 - Arquitetura de Hardware

3.4.1 HP iPAQ h5550

O iPAQ h5550 é um handheld da HP cujo sistema operacional é o Windows CE 2003 Premium, com um processador Intel XScale de 400MHz e 128MB SDRAM e 48 MB Flash ROM.

Ele possui embutida uma placa WiFi padrão IEEE 802.11b para acesso a rede wireless e conta com um mecanismo de encriptação de dados wireless WEP de 64 e 128 bits. Conta também com um leitor biométrico de digitais integrado com o equipamento.

3.4.2 Linksys BEFW11S4 Wireless BroadBand Router

O produto BEFW11S4 da Linksys é um access point que suporta o padrão IEEE 802.11b, com encriptação de dados wireless WEP de 64 e 128 bits. Ele possui 4 portas Ethernet 10/100 Mbps e funciona também como roteador, possuindo internamente um servidor DHCP.

3.4.3 Servidor de Aplicações / Servidor de Serviços

O Servidor de Aplicações foi colocado em um computador cujas características são: processador Athlon XP 2600+, 1512 MB de RAM e 80 GB de HD.

O Servidor de Serviços foi colocado em outro computador cujas características são: processador Athlon XP 1700+, 256 MB de RAM e 40GB de HD.

4. METODOLOGIA

Basicamente pode-se dividir o desenvolvimento do projeto em duas fases diferentes: especificação e implementação.

A fase de especificação, a qual descreve todos os módulos do sistema, foi desenvolvida com grande utilização dos conceitos de Engenharia de Software, que segundo [Pressman, 2001] “é a aplicação de uma abordagem sistemática, disciplinada, e quantificável, para o desenvolvimento, operação e manutenção de software”.

Seguindo essa filosofia, adotou-se a linguagem UML como metodologia de desenvolvimento de um sistema orientado a objetos. Dessa maneira, os principais diagramas sugeridos pelo UML foram criados como forma de apoio para a segunda fase de implementação.

As funcionalidades mapeadas como requisitos da solução foram formalizadas através dos Diagramas de Casos de Uso. Desses requisitos, foram extraídos os objetos da aplicação, que foram reunidos em classes conforme suas características e modelados através do Diagrama de Classes. A partir do Diagrama de Classe, criou-se o Diagrama de Entidade-Relacionamento que retrata as entidades geradoras do banco de dados da solução.

Essa fase de especificação foi dividida em diferentes etapas, que tiveram um representante designado para ser seu responsável. Assim, atuando como um gerente, o responsável controlava os produtos gerados, os prazos de entrega, o gerenciamento de recursos, a atribuição de atividades e a comunicação interna com o coordenador.

Para se relatar todas as tarefas necessárias do projeto, os prazos de entrega e os responsáveis por cada atividade adotou-se o Microsoft Project como ferramenta de gerenciamento. Assim, foi criado um cronograma que relatava todas as responsabilidades dos integrantes da equipe e fazia o controle das atividades.

Na fase de implementação, foi adotada a metodologia em espiral, que segundo [Pressman, 2001] “trata-se de um modelo evolucionário que combina a natureza iterativa da prototipagem com os aspectos controlados e sistemáticos do modelo sequencial linear. Fornece potencial para o desenvolvimento rápido de versões incrementais do

software. Usando o modelo espiral, o software é desenvolvido numa série de versões incrementais. A versão incremental, durante as primeiras iterações, pode ser um modelo de papel ou protótipo. Durante as últimas iterações, são produzidas versões cada vez mais completas do sistema”.

A Figura 11 retrata a idéia do modelo espiral de desenvolvimento de sistemas:

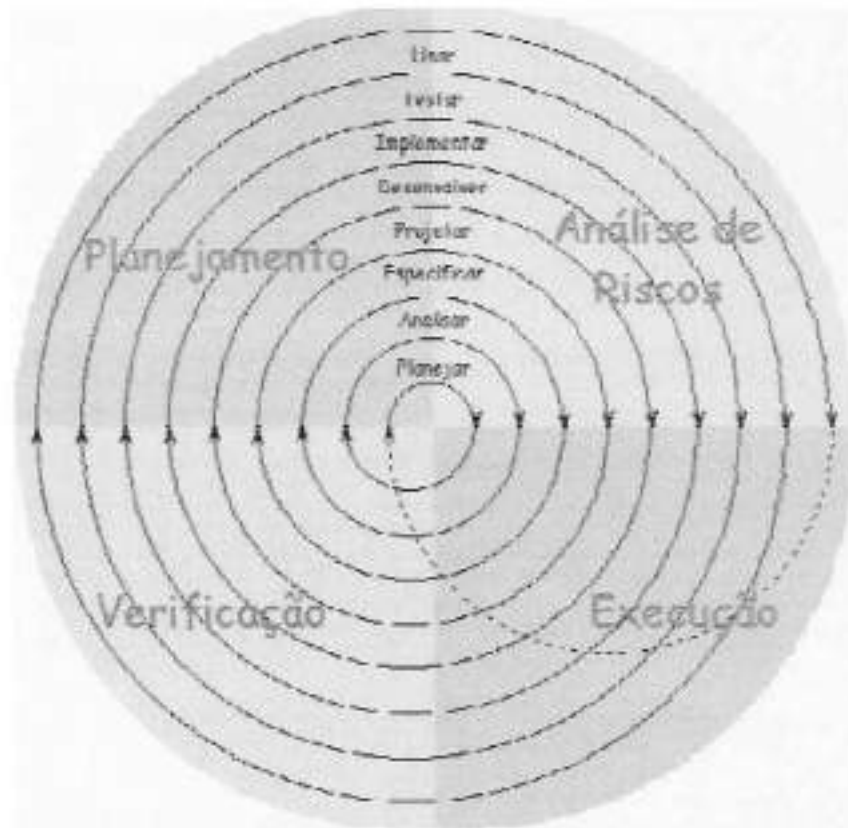


Figura 11 - Modelo Espiral

Fonte: <http://www.engenharia-software.com>

Esse conceito teórico foi parcialmente utilizado durante a fase de implementação, já que as etapas de planejamento, análise, especificação e projeto já haviam sido desenvolvidas na primeira etapa. Considerando o modelo espiral, praticaram-se as etapas de desenvolvimento, implementação, testes e uso.

Basicamente, as regras de negócio foram mapeadas a partir do documento de Casos de Uso da primeira etapa do projeto. Dessas regras, criava-se um serviço e sua

correspondente aplicação no Servidor de Aplicações. Fazendo uso da aplicação, os serviços eram testados e validados, iniciando, assim, uma nova iteração desse processo contínuo.

Com o desenrolar dessa metodologia para a fase de desenvolvimento, a solução ganhava mais maturidade a cada iteração realizada, e tanto o Servidor de Serviços quanto o de Aplicações tinham suas funcionalidades incrementadas gradualmente.

5. PROJETO E IMPLEMENTAÇÃO

A etapa de implementação do projeto é descrita nos itens abaixo e está apresentada conforme os módulos que constituem a solução.

5.1 Banco de Dados

Utilizou-se o software Design Databases v3.1 para efetuar a modelagem do Diagrama de Entidade-Relacionamento, criando as entidades e relacionamentos entre as mesmas. Depois disso, foi gerado um script SQL contendo todos os comandos de geração de tabelas a partir das entidades modeladas, bem como os comandos para geração dos relacionamentos entre as tabelas.

Utilizou-se a ferramenta Command Center, do gerenciador de banco de dados DB2, para carregar esse script e executá-lo, criando o banco de dados do sistema.

Não houve a necessidade de criação de índices, views ou triggers para a utilização do banco de dados, visto que a camada de persistência da aplicação foi modelada para que componentes mapeiem diretamente as tabelas e relacionamentos.

5.2 Biometria

A biometria do sistema foi desenvolvida utilizando a implementação do framework do consórcio BioAPI.org, que foi fundado para especificar uma API que traz independência de plataforma e de dispositivo para os desenvolvedores de aplicações e para os provedores de serviços de biometria. O consórcio é um grupo formado por mais de 90 companhias e organizações que possuem interesse comum em promover o crescimento do mercado de biometria. Dentre essas companhias se encontra a HP, fornecedora do dispositivo de leitura biométrica utilizado no projeto.

O consórcio desenvolveu uma especificação e uma implementação para uma API padronizada que é compatível com uma gama grande de aplicações que utilizem biometria e um grande número de tecnologias biométricas como impressão digital, íris, reconhecimento facial, geometria das mãos, dentre outras.

5.2.1 A tecnologia biométrica

O nível de Aplicação é o nível mais alto onde as funções biométricas básicas são implementadas – aquelas onde uma aplicação geralmente utilizaria para incorporar características biométricas com o intuito de identificação humana.

A intenção é de que o nível de Aplicação da API contenha todas as funções necessárias por uma aplicação que utilize autenticação biométrica. Além disso, o conjunto de funcionalidades opcionais deve ser minimizado. A principal função opcional é a Identification, em que somente BSPs especializados irão implementar essa função para a identificação de uma informação biométrica em uma grande população.

A abordagem tomada é a de se encapsular os aspectos únicos de tecnologias individuais de biometria e de implementações particulares de empresas, produtos e dispositivos, ao mesmo tempo provendo uma abstração de alto nível, que podem ser usadas com número de potenciais aplicações de software. O acesso à mecanismos de biometria deve ser através de interfaces padronizadas definidas pela especificação. Teoricamente, os BSPs providos pelas empresas em conformidade com essa especificação de interface podem ser usadas em qualquer aplicação desenvolvida com essa definição da BioAPI.

O fluxo básico de funcionamento é o mesmo para todos os tipos de tecnologia biométrica. Primeiro, o template inicial da informação biométrica do usuário deve ser construído. Isso é feito coletando um número de amostras através de um sensor. As características salientes são extraídas das amostras, e os resultados são combinados no template. A construção inicial desse template é chamada de Enrollment e os algoritmos usados para construção desse template geralmente são proprietários.

Feito isso, o template inicial é armazenado na aplicação, e usado no lugar de uma senha. Sempre que o usuário precisar ser autenticado, as amostras são capturadas do dispositivo, processadas de maneira a dar um grau de usabilidade e então verificadas contra o template armazenado anteriormente. Essa forma de autenticação biométrica é chamada Verification, uma vez que verifica que o usuário é quem ele diz ser.

Além disso, a tecnologia biométrica permite uma nova forma de autenticação chamada Identification. Nessa forma, o usuário não precisa afirmar uma identidade; o BSP compara as amostras do usuário contra um universo de templates, e decide quais delas se ajustam melhor a informação coletada do usuário. Dependendo das probabilidades de ajuste, conclui-se que a identidade do usuário corresponde ao template com maior índice de ajuste.

A figura 12 mostra algumas possíveis estratégias de implementação.

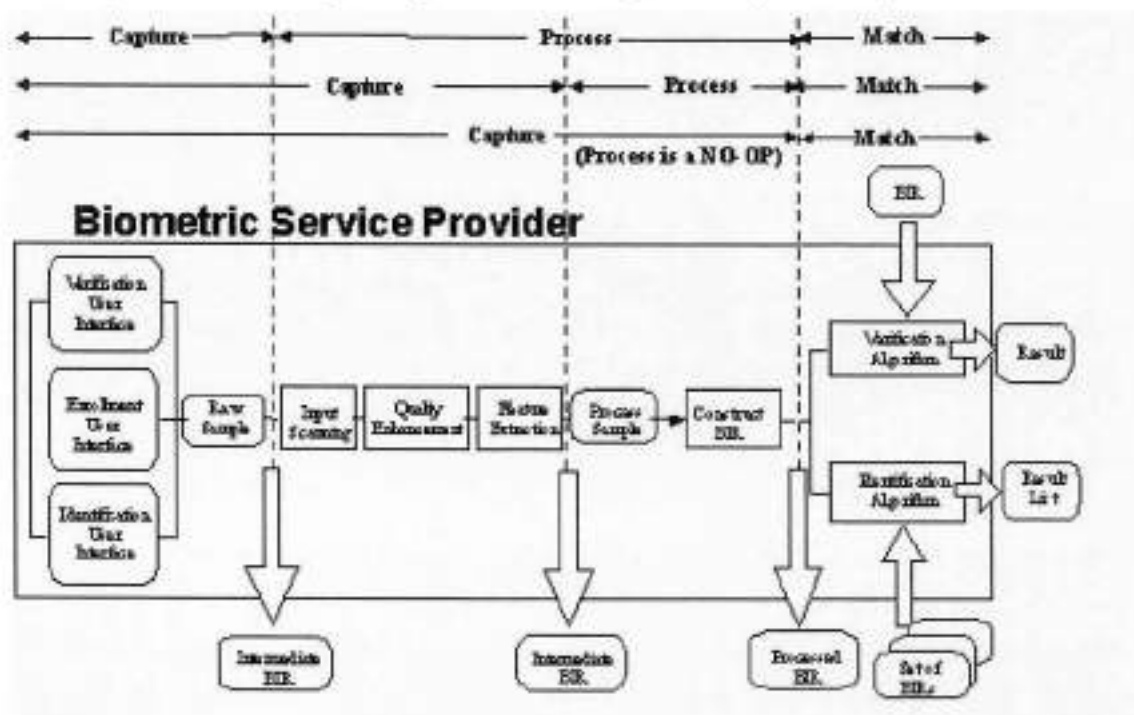


Figura 12 - Estratégias de Implementação

Fonte: BioAPI Specification / Version 1.01

Os vários passos nas operações de verificação e na identificação são mostrados na caixa denominada Biometric Service Provider. Os estágios identificados acima da caixa referem-se a funções primitivas da interface de alto nível: Capture, Process e Match, e isto mostra que um BSP tem níveis de liberdade no desenvolvimento das funções dessas primitivas. Esses níveis de liberdade não são mostrados na figura; o fabricante é livre para colocar a maioria, senão toda a função do BSP no dispositivo de leitura. Portanto, se o dispositivo contém uma base BIR, todas as funções podem ser executadas no próprio dispositivo.

5.2.2 BIRs e Templates

Esse padrão usa o termo *template* para se referir ao dado biométrico registrado do usuário. Esse *template* deve ser confrontado (com uma tolerância específica) contra amostra(s) lidas do usuário, para que possa ser autenticado.

O termo BIR se refere a qualquer dado biométrico que seja retornado pela aplicação; incluindo dados de baixo nível, dados intermediários, amostras processadas prontas para verificação ou identificação, além de dados registrados. Tipicamente o único dado armazenado persistentemente pela aplicação é o BIR a ser registrado, que é próprio *template*.

A estrutura é mostrada na figura 13.

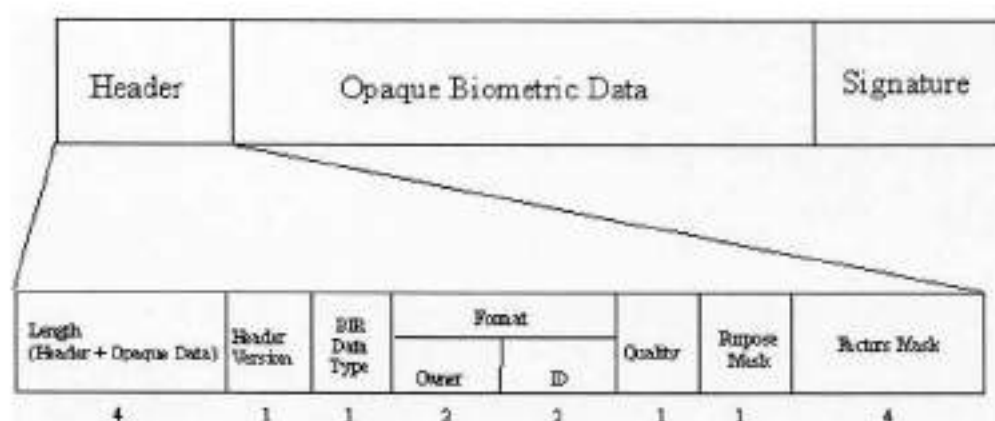


Figura 13 – BIR – Biometric Identification Record

Fonte: *BioAPI Specification / Version 1.01*

O formato do Opaque Biometric Data é indicado pelo campo Format do Header. Isso pode ser um formato proprietário ou um formato padronizado. O dado Opaque pode ser encriptado.

Os valores do campo Format Owner são atribuídos e registrados pela IBIA, que garante a unicidade desses valores. Os proprietários desses formatos registrados podem então criar um ou mais FormatIDs (tanto padronizados, quanto proprietários), correspondendo a um formato definido para os dados biométricos opacos subsequentes, que podem ser opcionalmente registrados pela IBIA.

O campo Signature é opcional. Para formatos BIR padronizados, a assinatura irá ter uma forma padronizada (a ser determinada quando o formato é registrado). Para formatos BIR proprietários (todos os existentes no presente momento), a assinatura pode ter qualquer forma que seja adequada ao BSP. Por isso, não há nenhuma estrutura em linguagem de programação C que defina a assinatura.

O campo BIR Data Type indica se a BIR é assinada e/ou encriptada.

Quando o provedor de serviços cria uma nova BIR, ele retorna um Handle para a o mesmo. Com isso, a maioria das operações pode ser realizada sem precisar retirar o controle sobre o BIR do provedor de serviços. Portanto, quando a aplicação necessita realizar alguma operação com a BIR a mesma é referenciada através do handle gerado.

Além disso, quando uma aplicação necessita de uma BIR como entrada, ela pode ser adquirida de três maneiras:

- *Por referência ao handle.*
- *Por referência a um índice de uma BIR em uma base de dados.*
- *Através da própria BIR.*

5.2.3 O Modelo da API

Existem três funcionalidades no nível mais alto de abstração da API:

- *Enroll: Quando as amostras são capturadas de um dispositivo, processados e retornados para a aplicação.*
- *Verify: Quando uma ou mais amostras são capturadas, processadas e confrontadas com um template de referência, retornando os resultados da comparação.*
- *Identify: Quando uma ou mais amostras são capturadas, processadas e confrontadas com um conjunto de templates. Uma lista é retornada identificando quais são as candidatas do conjunto para as amostras verificadas.*

Com essas funcionalidades de alto nível presentes, o desenvolvedor ganha o máximo de liberdade no desenvolvimento de suas aplicações permitindo com que o processamento dos dados possa ser implementado em uma arquitetura cliente / servidor.

Assim, a BioAPI permite que uma aplicação faça a captura da informação em uma máquina cliente e faça a verificação em uma máquina servidor.

5.2.4 FAR and FRR

As amostras de dados biométricos são compostas por streams de dados complexos produzidos por dispositivos sensíveis, e desta forma, nunca duas amostras provenientes de um mesmo usuário são iguais. Os templates são resultado da digitalização, processamento e compreensão destas amostras, as quais não são representações precisas de cada usuário. Desta forma, os resultados do confronto entre amostras biométricas e o template armazenado são apresentados em termos de probabilidade.

Existem duas formas possíveis de erro para o resultado do confronto: FAR ou falso positivo, e FRR ou falso negativo. O FAR é a probabilidade de uma amostra, erroneamente, ser autenticada contra um template armazenado, enquanto o FRR é a probabilidade de uma amostra ser, erroneamente, rejeitada por um template armazenado.

Para a versão da BioAPI implementada pela HP, as taxas de FRR e FAR para os níveis de segurança Regular, High e Extra High são fornecidos pela Figura 14.

<u>Security Level</u>	<u>FRR</u>	<u>FAR</u>	<u>Average of attempts</u>
<u>Regular</u>	<u>0.2331%</u> <u>(1 out of 429)</u>	<u>0.0010%</u> <u>(41 out of 4,054,737)</u>	<u>1.081967</u>
<u>High</u>	<u>0.4662%</u> <u>(2 out of 429)</u>	<u>0.0001%</u> <u>(4 out of 4,054,737)</u>	<u>1.107573</u>
<u>Extra High</u>	<u>0.6993%</u> <u>(3 out of 429)</u>	<u>0.0000%</u> <u>(0 out of 4,054,737)</u>	<u>1.153226</u>

Figura 14 - Níveis de Segurança da BioAPI

Fonte: <http://www.hp.com.br>

Este estudo demonstra a grande viabilidade da aplicação para uma grande população, como é o caso do projeto em questão, pois segundo o nível mais baixo de segurança, uma taxa de falso negativo de 41 ocorrências em 4.054.737 tentativas é um número muito baixo para o propósito da aplicação.

5.2.5 Implementação

Os clientes do estabelecimento possuem os dados persistidos no banco de dados, incluindo o dado biométrico, armazenado na forma de um BLOB, pois é necessário o cadastro de tais dados para que seja permitida a entrada do cliente no estabelecimento.

Assim, é possível identificar o cliente a partir do seu dado biométrico através do uso da BioAPI, comparando o dado biométrico a ser identificado com os dados armazenados no banco de dados. Para que isso seja possível, é necessária a utilização de um componente Java que possa chamar as funções disponibilizadas pela BioAPI. Como são duas plataformas totalmente distintas, uma na linguagem Java e a outra totalmente nativa, escrita em C++ e encapsulada na forma de uma DLL, é necessário a utilização de uma arquitetura da linguagem Java que é o JNI.

A arquitetura JNI permite que código Java que roda em uma JVM utilize aplicações e bibliotecas escritas em outras linguagens, como C, C++ e Assembly. Os desenvolvedores utilizam o JNI para escrever métodos nativos que lidam com situações onde a aplicação não pode ser inteiramente escrita na linguagem Java, como é o caso da necessidade da identificação do cliente utilizando a BioAPI.

A figura 15 descreve o funcionamento e a utilidade do JNI:

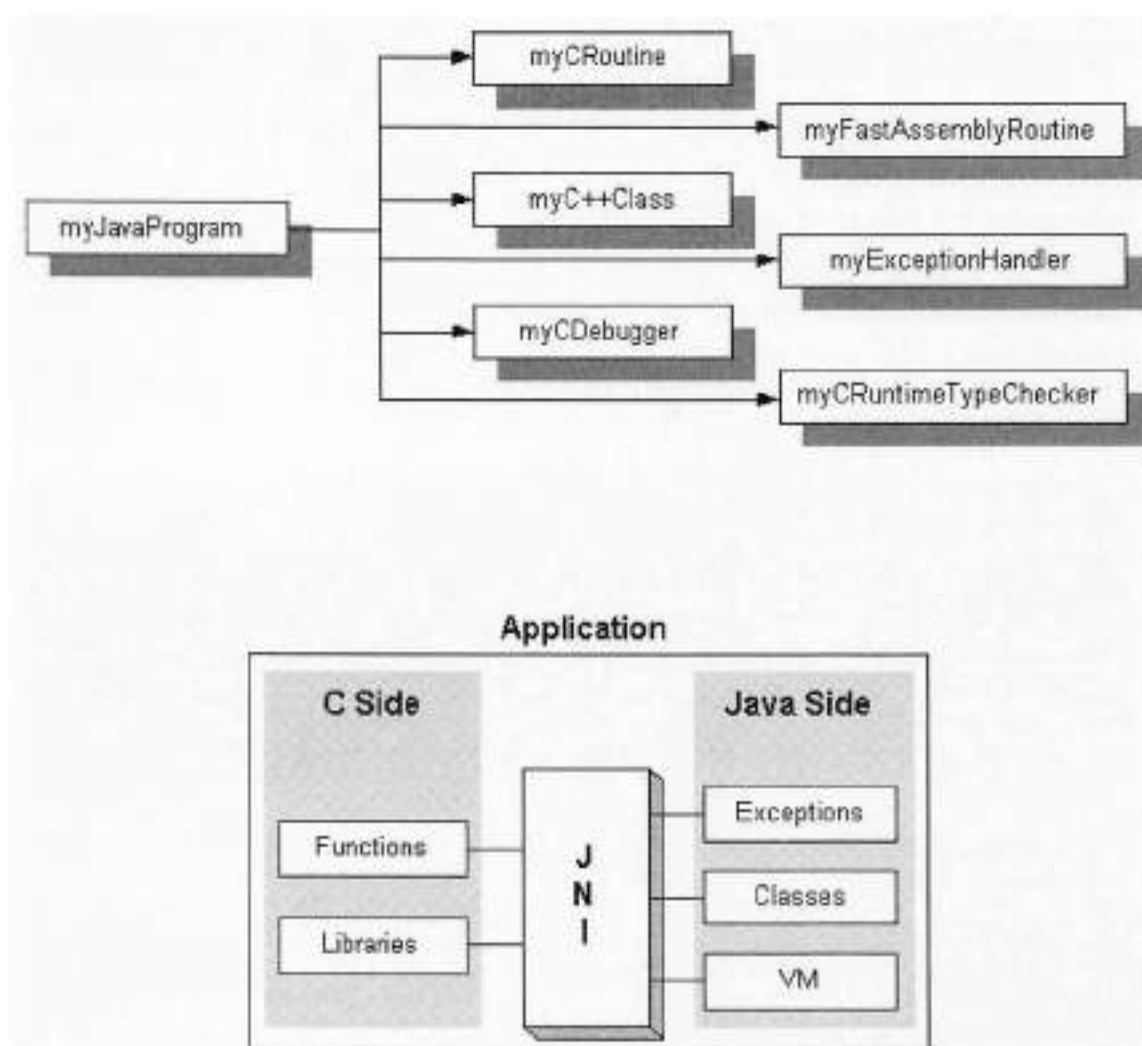


Figura 15 – JNI – Java Native Language

Fonte: <http://www.portaljava.com.br>

Desenvolveu-se um componente em C++ para chamar as funções da BioAPI desejadas para verificação do dado biométrico, utilizando como exemplo uma implementação de referência disponibilizada no site do consórcio BioAPI.org. Encapsulou-se esse componente em uma DLL para ser carregado na JVM e então poder ser utilizado pela classe Java responsável pela identificação do cliente.

Porém, a implementação de referência consistia em uma DLL versão “dummy” da BioAPI, que não implementava todas as funcionalidades do padrão utilizado pela BioAPI. Assim, isso impossibilitou o uso dessa DLL no sistema, uma vez que o método necessário para a verificação de entre duas BIR não estava implementado.

Não havendo outras alternativas de versões de BioAPI que pudessem ser utilizadas na arquitetura x86 e sistema operacional Windows 2000, optamos por mudar o planejamento inicial e realizar a verificação entre BIRs no dispositivo móvel, uma vez que, possuindo o dispositivo biométrico, tínhamos a implementação da BioAPI para essa arquitetura (processador Intel XScale com Windows CE) com todas as funcionalidades necessárias.

5.3 Servidor de Serviços

O Servidor de Serviços do sistema foi desenvolvido utilizando a plataforma J2EE, cuja principal função é disponibilizar serviços a serem consumidos pelo Servidor de Aplicações.

Com esse propósito, a arquitetura do Servidor de Serviços foi elaborada usando o conceito de SOA, que é um modelo de componentes que inter-relacionam diferentes unidades funcionais de uma aplicação através de uma interface e um contrato bem-definido entre esses serviços. Essa interface é definida de maneira que deva ser independente da plataforma de hardware, do sistema operacional e da linguagem de programação em que o serviço é implementado. Isso permite que os serviços interajam de maneira uniforme e universal.

Para desenvolver as aplicações do Servidor de Serviços, utilizou-se a ferramenta de desenvolvimento IBM WSAD, voltada para aplicações J2EE. O principal atrativo da ferramenta é a facilidade com que aplicações J2EE são criadas, por oferecer facilidades de criação de códigos, além de possuir embutido um WebSphere Application Server, possibilitando o deploy e o debug das aplicações no momento de desenvolvimento.

Já a camada de persistência, contendo o banco de dados da aplicação, foi implementada utilizando componentes CMP Entity EJB. Cada componente mapeia uma tabela do banco de dados, e a persistência dos dados, por se tratar de um CMP, é controlada pelo Container EJB.

Assim, para cada tabela do banco de dados, foi criado um componente Entity EJB, localizados no pacote `nocard.ejb.model`. Os seguintes componentes foram criados:

- *Atividade*.

- *Categoria.*
- *Cliente.*
- *Comanda.*
- *Comandaprodutoconsumo.*
- *Evento.*
- *Fornecedor.*
- *Mensagens.*
- *Movimentoestoque.*
- *Papel.*
- *Papelatividade.*
- *Pedido.*
- *Produtoconsumo.*
- *Produtoconsumocomposicao.*
- *Produtoestoque.*
- *Produtoestoquepedido.*
- *Tipomovimento.*
- *Usuário.*
- *Usuariopapel.*

A figura 16 mostra a implementação dos componentes utilizando a ferramenta WSAD.

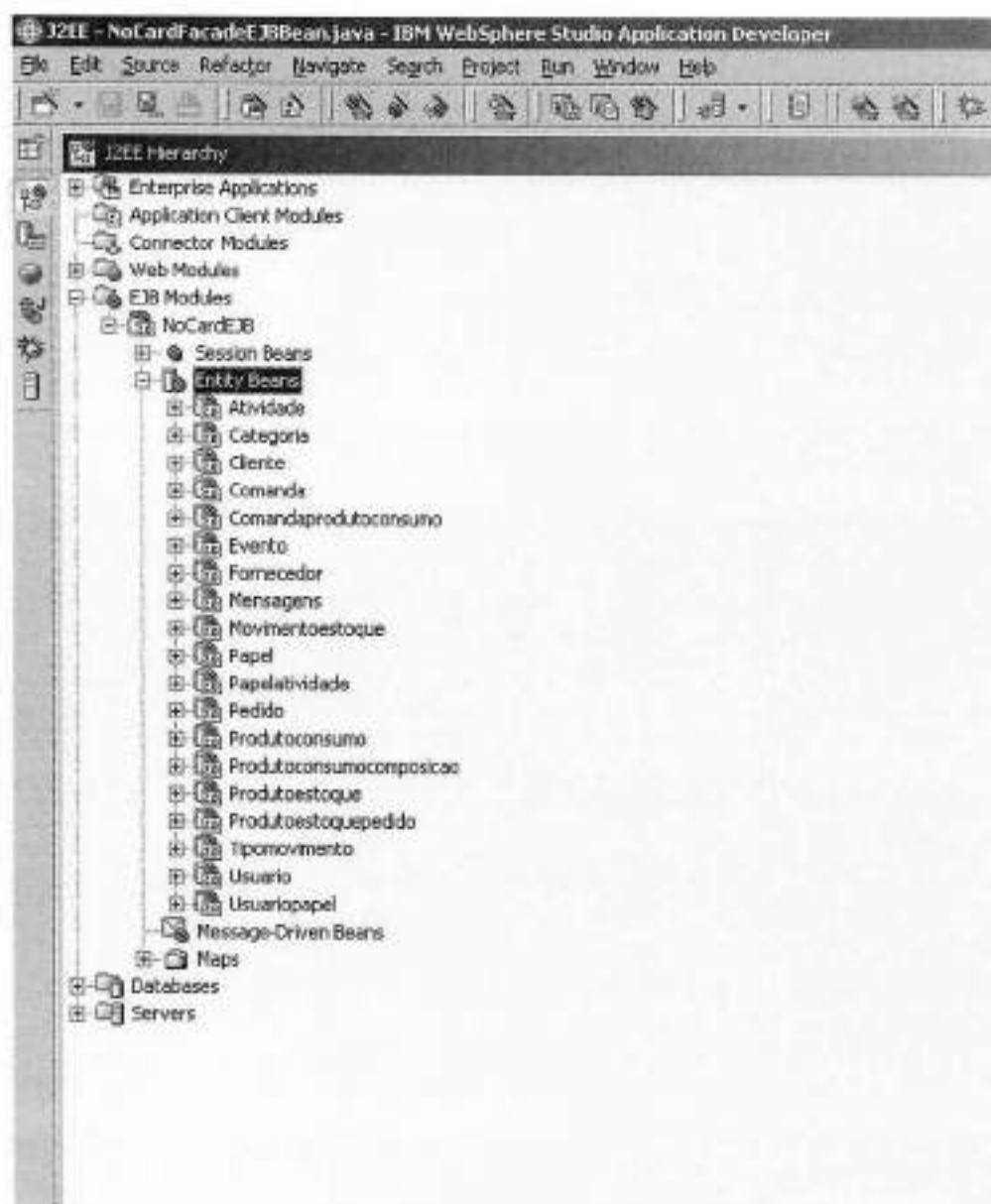


Figura 16 - WSAD Entity Beans

Após a implementação da camada de persistência, utilizou-se um Java Pattern chamado DTO, através do qual são criados objetos simples, compostos de atributos e métodos de `get()` e `set()`, utilizados para carregar dados de um lugar para outro.

Foram criados os seguintes DTOs, localizados no pacote `nocard.dto`:

- *Atividade.*
- *Categoria.*

- *Cliente.*
- *Comanda.*
- *Comandaprodutoconsumo.*
- *Evento.*
- *Fechamento.*
- *Fornecedor.*
- *Mensagens.*
- *Movimentoestoque.*
- *Papel.*
- *Papelatividade.*
- *Pedido.*
- *Produtoconsumo.*
- *Produtoconsumocomposicao.*
- *Produtoestoque.*
- *Produtoestoquepedido.*
- *Tipomovimento.*
- *Usuário.*
- *Usuariopapel.*

Esses DTOs foram implementados tendo atributos simples, como int, double, long e String, e atributos complexos como outros objetos DTO. Arrays simples ou complexos também foram adotados como atributos dos DTOs. O cuidado por uma restrita definição de atributos foi tomado para que esses objetos pudessem ser trafegados via Web Services, seguindo um padrão para se evitar conflitos.

No passo seguinte, foi utilizado um design pattern chamado Façade, cuja implementação se dá através de um Stateless Session EJB, para implementação das regras de negócio e acesso à camada de persistência, sendo assim responsável por acessar os Entity EJBs.

Esse design pattern define um componente de negócio de alto nível que contém e centraliza interações complexas entre componentes de negócio de nível mais baixo.

Assim, provê aos clientes uma interface única para a funcionalidade de uma aplicação ou um conjunto de aplicações. Também desacopla componentes de nível mais baixo de outros, fazendo que o design seja mais flexível e compreensível.

A figura 17 mostra como o Façade atua.

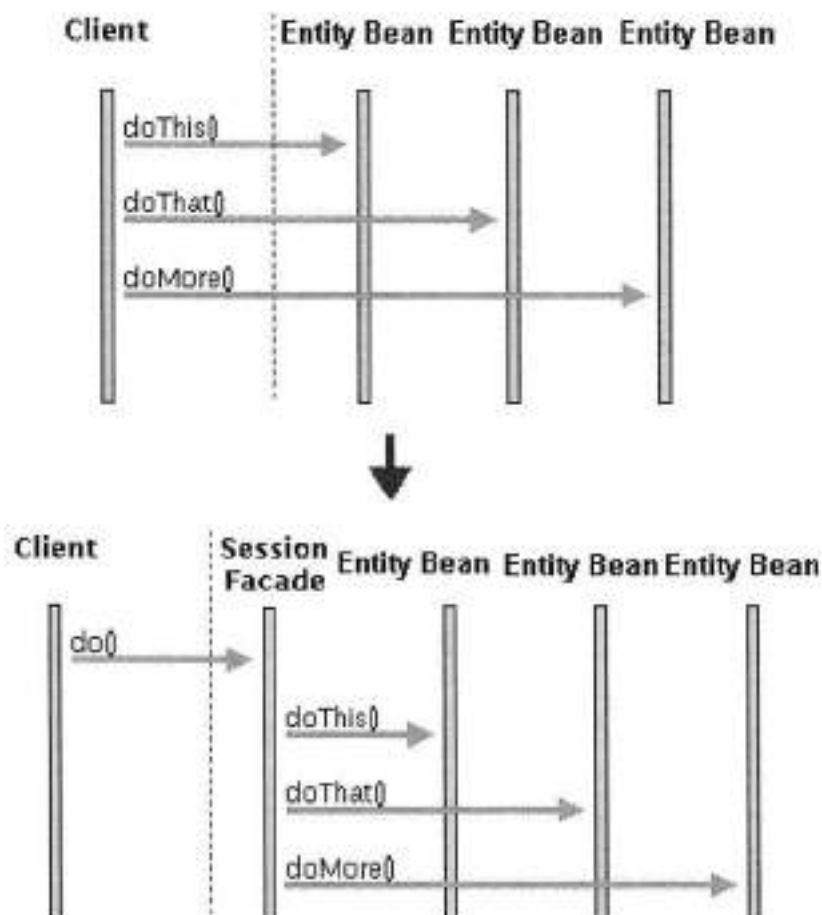


Figura 17 - Session Façade

Com isso, toda lógica de negócio foi colocada nesse componente, chamado NoCardFacadeEJB, sendo uma interface única para acesso ao Servidor de Serviços.

A figura 18 mostra o desenvolvimento do NoCardFacadeEJB utilizando o WSAD, bem como as três classes que o implementam.

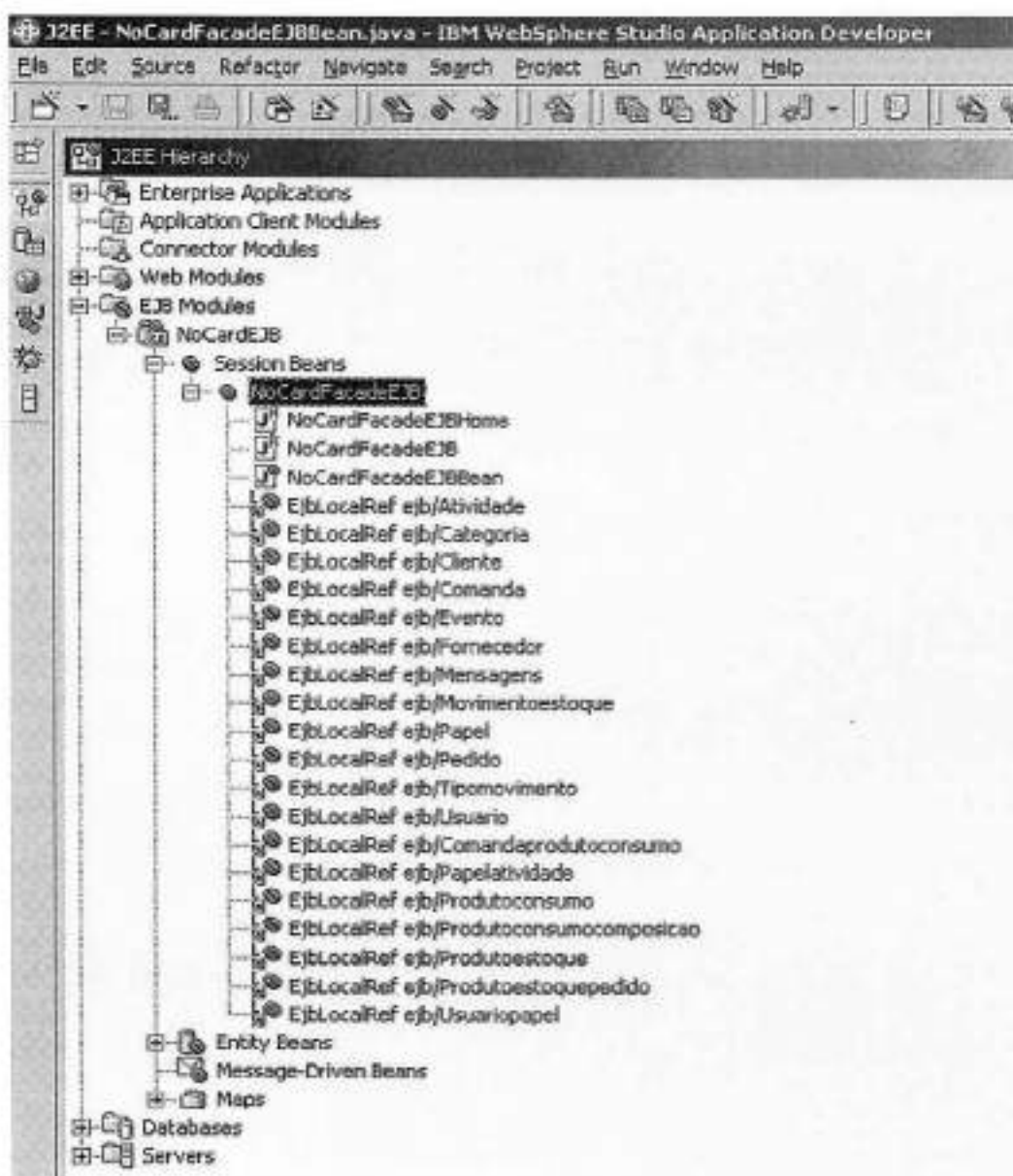


Figura 18 - WSAD NoCardFacadeEJB

Dessas três classes, existe a interface chamada NoCardFacadeEJBBean, que é a classe que implementa todos os métodos de negócio dentro do componente. Essa classe possui os seguintes métodos, separados por caso de uso implementado, que posteriormente foram transformados em serviços:

- *Cadastro de Usuários:*
 - `getUsuarios()`.

- createUsuario(usuarioDTO, UsuarioPapel[]).
- inativaUsuario(usuarioID).
- updateUsuario(usuarioDTO, usuarioPapel[]).
- deleteUsuario(usuarioID).
- getUsuarioByUsername(username).
- getAtividades().
- createAtividade(atividadeDTO).
- updateAtividade(atividadeDTO).
- deleteAtividade(atividadeID).
- getPapeis().
- createPapel(papelDTO, papelAtividade[]).
- updatePapel(papelDTO, papelAtividade[]).
- deletePapel(papelID).
- getAtividadesPapel(papelID).
- getPapeisUsuario(username).
- *Login no Sistema:*
 - verificaSenha(username, senha).
- *Cadastro de Eventos:*
 - getEventos().
 - filtroEventosPeriodoData(dataInicial, dataFinal).
 - filtroEventosNome(nome).
 - createEvento(eventoDTO).
 - updateEvento(eventoDTO).
 - deleteEvento(eventoID).
- *Cadastro de Clientes:*
 - getClientes().
 - filtroClientesNome(filtro).
 - createCliente(clienteDTO).
 - createComanda(clienteDTO, eventoDTO).
 - updateCliente(clienteDTO).

- deleteCliente(clienteID).
- inativaCliente(clienteID).
- *Solicitação de Identificação no Sistema:*
 - buscaClienteAutenticado(dadoBiometrico, eventoDTO).
 - getComanda(clienteDTO, eventoDTO).
- *Entrada no Estabelecimento:*
 - buscaCliente(dadoBiometrico, eventoDTO).
- *Solicitação de Saída do Estabelecimento:*
 - updateComanda(comandaDTO).
- *Cadastro de Categoria:*
 - getCategorias().
 - filtroCategorias(filtro).
 - getCategoriasFilhas(categoriaID).
 - createCategoria(categoriaDTO).
 - updateCategoria(categoriaDTO).
 - deleteCategoria(categoriaID).
- *Cadastro de Produto de Consumo:*
 - getProdutosconsumo().
 - filtroProdutoconsumo(filtro).
 - getProdutosconsumoCategoria(categoriaID).
 - createProdutoconsumo(produtoconsumoDTO, produtoconsumocomposicao[]).
 - updateProdutoconsumo(produtoconsumoDTO, produtoconsumocomposicao[]).
 - deleteProdutoconsumo(produtoconsumoID).
- *Cadastro de Pedidos:*
 - adicionaComandaProdutoConsumo(ComandaproductconsumDTO).
 - retiraComandaProdutoConsumo(ComandaprodutoconsumoDTO).
- *Fechamento Diário:*
 - fechamentoDiario(eventoDTO).

- *Cadastro de Produtos de Estoque:*
 - **getProdutosestoque().**
 - **filtroProdutosestoque(filtro).**
 - **getProdutoestoqueCategoria(categoriaID).**
 - **createProdutoestoque(produtoestoqueDTO).**
 - **deleteProdutoestoque(produtoestoqueID).**
 - **updateProdutoestoque(produtoestoqueDTO).**
 - **getProdutoestoqueLimite().**
- *Solicitação de Pedido de Produto de Estoque:*
 - **getProdutoEstoqueLimite().**
 - **getPedidosFornecedor(fornecedorID).**
 - **updatePedido(pedidoDTO, Produtoestoquepedido[]).**
 - **deletePedido(pedidoID).**
 - **getMovimentoestoque(produtoestoqueID).**
 - **createPedido(pedidoDTO, Produtoestoquepedido[]).**
 - **getPedidos().**
 - **getProdutosEstoquePedido(pedidoID).**
- *Cadastro de Compra de Produto de Estoque:*
 - **updatePedido(PedidoDTO).**
- *Devolução de Produtos de Estoque:*
 - **updatePedido(PedidoDTO).**
- *Relatório de inventário de estoque:*
 - **getProdutosestoque().**
 - **getMovimentosestoque(produtoestoqueID).**
- *Cadastro de Fornecedores:*
 - **getFornecedores().**
 - **filtroFornecedores(filtro).**
 - **createFornecedor(fornecedorDTO).**
 - **deleteFornecedor(fornecedorID).**
 - **updateFornecedor(fornecedorDTO).**

- *Solicitação de Balanço:*
 - **fechamentoPeriodo(dataInicial, dataFinal).**
 - **getMovimentosestoque(produtoestoqueID).**
- *Geração de Mala Direta:*
 - **getClientesAniversario(mesInicial, mesFinal).**
 - **enviaEmail(emailHTML).**
 - **enviaEmailClientes(clienteDTO[], emailHTML).**

Cada método possui uma implementação de uma regra de negócio definida na descrição dos casos de uso.

Após o desenvolvimento do NoCardFacadeEJB, foi criado um projeto Web para implementação dos Web Services, chamado NoCardWSRouter. A ferramenta WSAD facilita a criação dos Web Services a partir de um Session EJB, possuindo um wizard que mapeia os métodos desse componente para disponibilizá-los como Web Services, criando uma interface WSDL, NoCardFacadeEJB.wsdl, para ser disponibilizada ao Servidor de Aplicações.

Com isso, todos os componentes necessários para disponibilização dos serviços do Servidor de Serviços foram desenvolvidos, sendo então implementados no WAS da ferramenta de desenvolvimento.

5.4 Servidor de Aplicações

O Servidor de Aplicações do sistema foi desenvolvido utilizando a plataforma Microsoft .NET, e sua principal função é disponibilizar aplicações ao usuário final, utilizando como regras de negócio os serviços disponibilizados pelo Servidor de Serviços.

Utilizando a ferramenta Visual Studio .NET, mapeou-se, automaticamente, a interface WSDL disponibilizada pelo Servidor de Serviços para um componente Proxy Web Services que centraliza o acesso aos serviços e contém uma descrição dos tipos complexos envolvidos na comunicação entre as duas plataformas.

Em função de algumas incompatibilidades entre as plataformas J2EE e .NET, principalmente em relação à maneira de como os dados são transmitidos nas mensagens SOAP trocadas nos serviços e a algumas necessidades particulares da linguagem .NET, foi necessário criar Wrappers que encapsulavam os dados provenientes dos serviços. Em alguns casos um tratamento especial foi realizado, como por exemplo, conversão de tipos de dados.

Um outro componente do tipo Proxy foi criado para centralizar todas as interações com os Proxy Web Services e com os Wrappers da aplicação, sendo um ponto de acesso único às lógicas de negócio por parte da aplicação do Servidor de Aplicações.

Foi utilizado o padrão MVC para desenvolvimento da parte Web da aplicação, cuja origem se deu na linguagem SmallTalk e atualmente se tornou um padrão largamente utilizado no desenvolvimento de aplicações empresariais de multicamadas.

As camadas do padrão MVC são descritas abaixo:

- *Model*: o Model representa os dados empresariais e as regras de negócio que gerenciam e atualizam esses dados.
- *View*: a View apresenta o conteúdo do Model. Acessa os dados empresariais através do Model e especifica como os dados devem ser apresentados, tendo como responsabilidade manter a consistência na apresentação quando o Model se altera. Isso pode ser feito de duas maneiras: usando o modelo push, onde a View se registra no Model para receber notificações de mudança, ou usando o modelo pull, onde a View é responsável por chamar o Model quando precisa dos dados mais atuais.
- *Controller*: o Controller traduz interações com a View em ações para serem executadas pelo Model. Em um cliente GUI stand-alone, as interações do usuário podem ser clicks em botões ou seleção em um menu, enquanto em uma aplicação Web essas ações aparecem como requisições HTTP post e get. As ações exercidas pelo Model incluem ativar processos de negócio ou mudar o estado do Model. Baseado nas interações e no resultado das ações do Model, o Controller responde selecionando a View apropriada.

A Figura 19 mostra como funcionam as interações entre as camadas do padrão MVC:

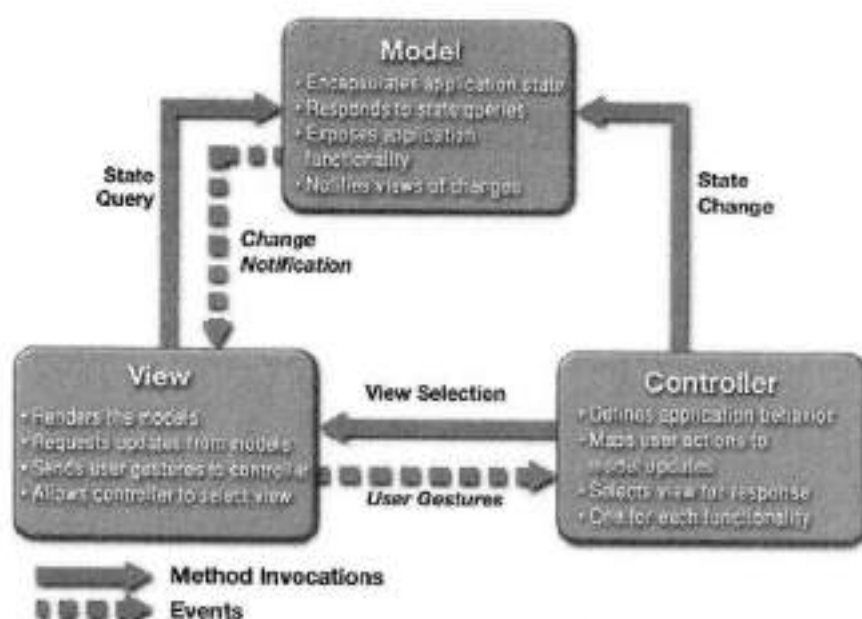


Figura 19 - Padrão MVC

Fonte: IBM Software Group

Esse tipo de arquitetura foi utilizado porque a linguagem .NET permite que se associe um Controller a cada página, podendo assim mapear toda a interação que o usuário introduzir na interface à alguma regra de negócio contida no Proxy criado.

Cada página desenvolvida é um WebForm, cujo desenvolvimento gráfico é facilitado pelo Visual Studio .NET, onde cada componente da interface está disponível em uma barra de ferramentas, bastando selecionar e organizar o componente no espaço reservado. É possível atribuir uma ou mais ações a cada componente, controlando-as através da classe de Controller de cada página.

O desenvolvimento de páginas é facilitado com o uso da ferramenta, uma vez que muitos componentes Web estão disponíveis ao desenvolvedor, bastando acrescentar alguma lógica de negócio para o devido funcionamento.

Do planejamento inicial, as seguintes funcionalidades foram implementadas:

- *Login.*
- *Administração do sistema.*

- *Cadastro de Usuários.*
- *Cadastro de Eventos.*
- *Atualização do Cadastro de Clientes.*
- *Cadastro de Categorias.*
- *Cadastro de Produtos de Estoque.*
- *Cadastro de Produtos de Consumo.*
- *Cadastro de Fornecedores.*
- *Gerenciamento do Estoque.*
- *Cadastro de Compras de Produtos de Estoque.*
- *Fechamento Diário.*
- *CRM.*

As outras funcionalidades planejadas inicialmente não foram desenvolvidas devido a quatro fatores:

- *Redução do grupo durante o andamento do projeto.*
- *Indisponibilidade do sensor biométrico.*
- *Falta de experiência de alguns membros na plataforma adotada.*
- *Falta de tempo hábil para treinamentos e para o desenvolvimento em si.*

5.5 Aplicação Móvel

A aplicação móvel foi desenvolvida utilizando WinForms da ferramenta de desenvolvimento Visual Studio .NET.

O ambiente de execução em um handheld para a aplicação móvel é limitado pelo Compact Framework .NET, que é uma versão do Framework .NET voltada para dispositivos móveis com sistema operacional Windows CE e que possuem recursos reduzidos. O Compact Framework .NET gerencia essas limitações do sistema.

A maior parte da estrutura montada para o uso dos serviços pela aplicação do Servidor de Aplicações foi reutilizada, tendo que ser adequada para as limitações do Compact Framework .NET. Porém, toda a idéia da utilização de tal estrutura foi mantida.

Cada interface criada é um WinForm, e a facilidade de desenvolvimento gráfico oferecida pelo Visual Studio .NET é a mesma daquela utilizada na criação do Servidor de Aplicações, ou seja, cada componente da interface está disponível em uma barra de ferramentas, bastando o desenvolvedor selecionar e organizar o componente no espaço reservado.

O dado biométrico é obtido com a leitura da digital do usuário pelo leitor do handheld, e é interpretada e transformada em um array de bytes por uma API que faz a interface entre o hardware e a aplicação. Essa API baseia-se na especificação da BioAPI, conforme explicado anteriormente, e limita-se a tratar somente do leitor biométrico do dispositivo.

Com isso, cada interface é responsável por instanciar e colocar os dados nos objetos de modo a utilizar os serviços do Proxy corretamente, que por sua vez utiliza os serviços disponíveis que efetuam a regra de negócio adequada para cada ação do usuário.

A figura 20 mostra a tela de login e a tela inicial de apresentação do sistema:

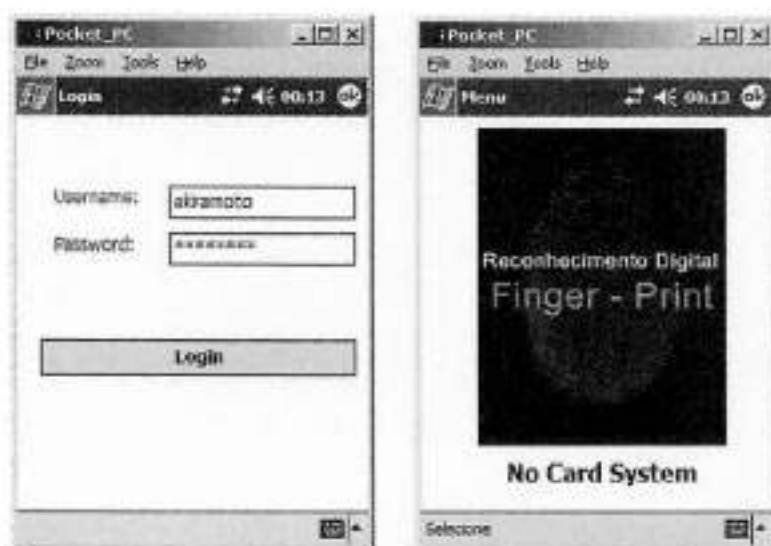


Figura 20 - Login e Apresentação da Aplicação Móvel

A figura 21 mostra as interfaces que controlam a entrada do cliente no estabelecimento:

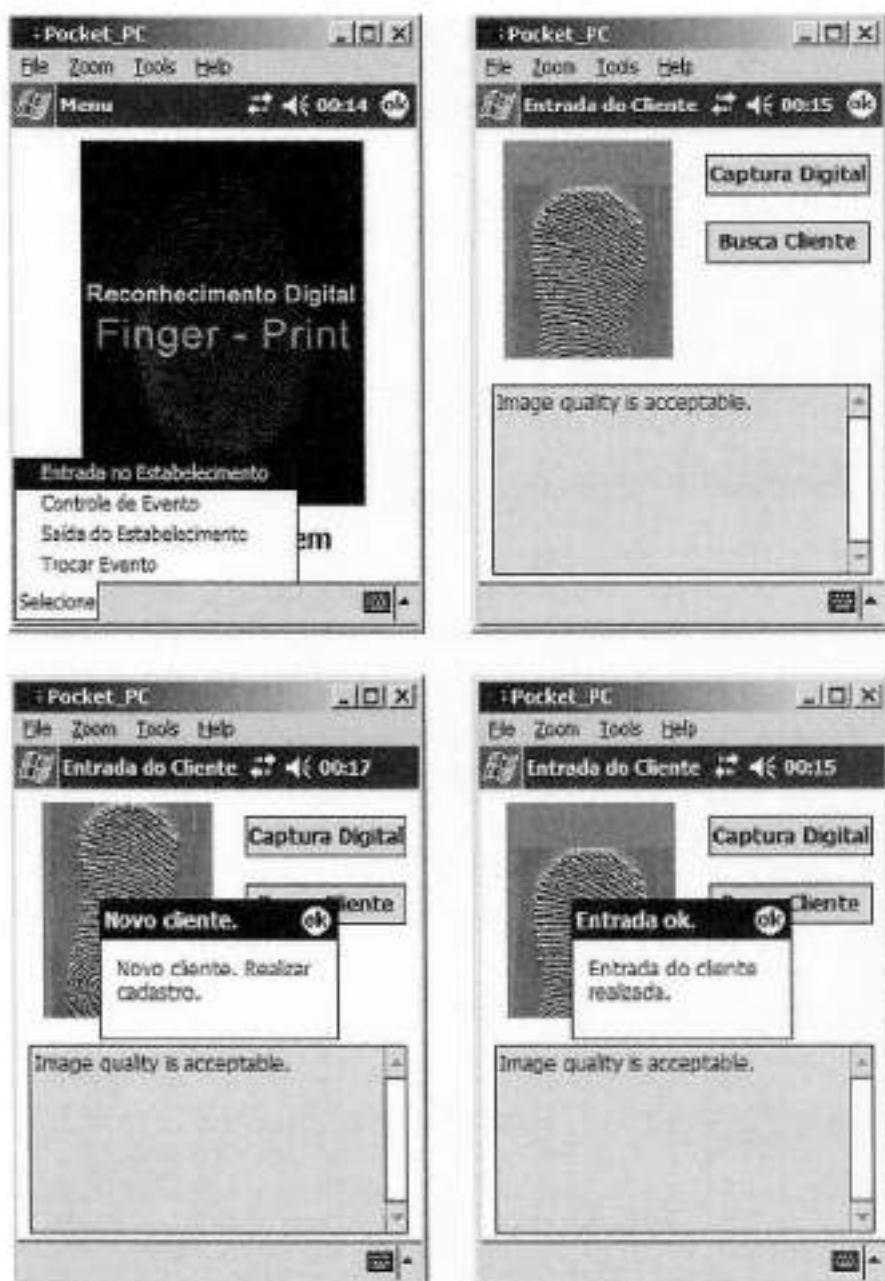


Figura 21 - Entrada no Estabelecimento

A Figura 22 mostra as interfaces que são utilizadas para cadastrar novos pedidos para um cliente autenticado:



Figura 22 - Cadastro de Pedidos

6. TESTES E AVALIAÇÃO

A fase de testes e avaliações é uma etapa crítica de garantia de qualidade de software e representa a revisão final da especificação, projeto e geração de código. Em função da importância dessa fase, ela foi bastante abordada durante toda a fase de implementação do projeto e após a sua finalização.

O grande objetivo dessa fase é descobrir e corrigir tantos erros quantos possíveis antes da entrega da versão final da solução. Em linhas gerais, esses erros foram mapeados através de duas diretrizes sistemáticas: o exercício da lógica interna dos componentes de software e o exercício dos domínios de entrada e saída do programa para descobrir erros nas funções, comportamento e desempenho.

Essas duas diretrizes foram abordadas através da metodologia de se testar os módulos do sistema após cada iteração do processo de implementação e após a concepção final da solução, sempre usando o princípio de que todos os testes são relacionados aos requisitos do cliente.

Buscando respeitar esse princípio, foi usado o método de teste comportamental, ou teste caixa-preta, o qual focaliza a busca pelo cumprimento das especificações funcionais de software e por isso aconteceram sobre cada caso de uso implementado.

O teste caixa-preta permite a derivação de conjuntos de condições de entrada que vão exercitar plenamente todos os requisitos funcionais do programa e tenta encontrar erros das seguintes categorias:

- *funções incorretas ou omitidas.*
- *erros de interface.*
- *erros de estrutura de dados ou de acesso a base de dados.*
- *erros de comportamento ou desempenho.*
- *erros de iniciação e término.*

Em termos gerais, a cada Caso de Uso implementado, o teste de caixa-preta era realizado na tentativa de se encontrar erros funcionais do Servidor de Serviços, nas interações com o usuário da aplicação do Servidor de Aplicações e na comunicação entre os módulos através de Web Services.

Depois que cada iteração foi testada e todas as partes do sistema acopladas, foi feito então um teste de caixa-preta final do sistema, visando a validação do poder de comunicação entre todos os módulos. Nesse momento, o sistema é testado “no varejo”, já que o objetivo é avaliar o sistema como um todo.

7. CONSIDERAÇÕES FINAIS

Em termos gerais, o projeto foi bem sucedido, visto que o principal objetivo foi alcançado: o desenvolvimento de uma solução que elimina a utilização de comandas em bares, restaurantes e casas noturnas.

Além disso, a oportunidade de familiarização com diferenciadas tecnologias como biometria, comunicação wireless e integração de plataformas heterogêneas também permitiu um maior amadurecimento dos integrantes do grupo. Outros pontos importantes a serem considerados foram a intensa utilização dos conceitos de Engenharia de Software e de práticas de gerenciamento de projetos.

O uso intenso dessas tecnologias possibilitou uma grande troca de experiências entre a equipe, além de proporcionar um aprendizado conjunto no uso das mesmas. Ainda, o projeto possibilitou e exigiu muito trabalho em equipe, além de exercitar todo o conhecimento de Engenharia de Computação aprendido durante o curso. Isso trouxe uma bagagem muito boa para os integrantes, pois foi vivenciado um ambiente de trabalho que certamente os mesmos irão encontrar no breve futuro profissional.

A divisão do sistema em Servidor de Serviços, Servidor de Aplicações e Dispositivo Móvel, cada qual com suas tecnologias e produtos, foi benéfica pois possibilitou a divisão de tarefas de maneira estruturada entre os componentes do grupo e também um maior controle sobre atividades e produtos gerados.

Entretanto, nem todas as funcionalidades especificadas foram implementadas, por diferentes motivos: redução do número de integrantes do grupo durante o andamento do projeto, falta de tempo hábil para treinamentos em algumas tecnologias e para o desenvolvimento do projeto, indisponibilidade de alguns recursos de infra-estrutura física, como o sensor biométrico e certas incompatibilidades de padrões, hardware e software.

Devido ao forte apelo comercial do projeto, as perspectivas de continuidade são muito boas, uma vez que o projeto de formatura foi somente um primeiro passo para verificar a viabilidade tecnológica da solução e para que os integrantes tivessem um

maior contato com as tecnologias abordadas. O grupo pretende dar continuidade, melhorando as regras de negócio, a robustez e as interfaces com o usuário final. Além disso, pretende-se abordar as funcionalidades que foram retiradas, buscando principalmente a integração de um leitor biométrico para o Servidor de Aplicações.

Vale a pena mencionar que o grupo recebeu uma consulta por parte de uma empresa espanhola a respeito da maturidade do projeto e da viabilidade de comercialização do produto no mercado brasileiro e no mercado europeu. Isso abriu novas perspectivas e motivou o grupo ainda mais a continuar o projeto.

“O trabalho de formatura agregou muito valor para minha formação como Engenheiro da Computação, uma vez que pude exercitar todo o conhecimento adquirido durante os cinco anos da POLI, direta e indiretamente. Apesar de todas as dificuldades enfrentadas, aprendi muito, tanto no aspecto técnico quanto no aspecto emocional. Além disso, foi extremamente gratificante trabalhar em um grupo motivado e interessado em sempre evoluir conjuntamente.” – Augusto Takahiro Kiramoto.

“A oportunidade de se trabalhar em uma equipe de elevada e constante motivação pela busca do sucesso é extremamente enriquecedora e gratificante. Além disso, acrescenta-se à essa experiência pessoal o crescimento técnico adquirido durante o desenvolvimento do projeto, fato que certamente contribuirá para futuras empreitadas profissionais.” – Gabriel Ribeiro Vargas Duarte.

“A oportunidade de realizar um projeto de tamanha complexidade e trabalhar com uma equipe de elevado nível profissional, em que cada integrante possuía experiências em diferentes áreas da computação, tornou o projeto uma experiência extremamente enriquecedora. Além disso, a qualidade do trabalho realizado, proporcionou-me uma gratificante sensação de que a jornada foi cumprida após os cinco anos no curso de Engenharia da Computação da Escola Politécnica.” – Stiverson Stopa Assis Palma.

LISTA DE REFERÊNCIAS

PRESSMAN, R. S. **Engenharia de Software**. 5 ed. Rio de Janeiro:McGraw-Hill, 2002.

GAMMA, E.; HELM, R.; JOHNSON, R.; **Padrões de Projeto**. São Paulo: Bookman, 2002.

MELO, A. C. **Desenvolvendo Aplicações com UML**. 1 ed. Rio de Janeiro: Brasport, 2002.

THE BIOAPI CONSORTIUM. **BioAPI Specification**. 1.01 v. Disponível em: <<http://www.bioapi.org>>. Acesso em: 19 de out. 2004.

IBM SOFTWARE GROUP. **IBM WebSphere Software**. Disponível em: <<http://www-306.ibm.com/software/websphere/>>. Acesso em: 03 de set. 2004.

IBM SOFTWARE GROUP. **IBM DB2 Software**. Disponível em: <<http://www-306.ibm.com/software/data/db2/>>. Acesso em: 03 de set. 2004.

JAVA SUN MICROSYSTEMS. Apresenta recursos e atividades desenvolvidas. Disponível em: <java.sun.com/j2ee/>. Acesso em: 03 de set. 2004.

MICROSOFT .NET. Apresenta recursos e atividades desenvolvidas. Disponível em: <www.microsoft.com/net/>. Acesso em: 12 de set. 2004.

APÊNDICE I

PCS2050-10-CD-2004: CD contendo a documentação, código e manual do usuário referentes ao projeto.



Solução de Identificação Biométrica
Gerenciada por Plataforma Heterogênea (J2EE E .NET)
Documento Integrado de Casos de Uso

SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA
GERENCIADA POR PLATAFORMA HETEROGÊNEA
(J2EE E .NET)

ANEXO A

Documento Integrado de Casos de Uso



Índice

1. CADASTRO DE USUÁRIOS	9
1.1. ATORES	9
1.2. DIAGRAMA DE CASOS DE USO	9
1.3. INSERÇÃO DE USUÁRIO	10
1.3.1. FLUXO DE EVENTOS	10
1.3.2. PRÉ CONDIÇÕES	10
1.3.3. PÓS CONDIÇÕES	11
1.4. INATIVAÇÃO DE USUÁRIO	11
1.4.1. FLUXO DE EVENTOS	11
1.4.2. PRÉ CONDIÇÕES	11
1.4.3. PÓS CONDIÇÕES	11
1.5. ALTERAÇÃO DE DADOS DE USUÁRIO	12
1.5.1. FLUXO DE EVENTOS	12
1.5.2. PRÉ CONDIÇÕES	12
1.5.3. PÓS CONDIÇÕES	12
1.6. VISUALIZAÇÃO DE DADOS DE USUÁRIO	13
1.6.1. FLUXO DE EVENTOS	13
1.6.2. PRÉ CONDIÇÕES	13
1.6.3. PÓS CONDIÇÕES	13
1.7. EXCLUSÃO DE USUÁRIOS	14
1.7.1. FLUXO DE EVENTOS	14
1.7.2. PRÉ CONDIÇÕES	14
1.7.3. PÓS CONDIÇÕES	14
2. LOGIN NO SISTEMA	15
2.1. ATORES	15
2.2. DIAGRAMAS DE CASOS DE USO	15
2.3. FLUXO DE EVENTOS	15
2.4. PRÉ CONDIÇÕES	16
2.5. PÓS CONDIÇÕES	16
3. CADASTRO DE EVENTOS	17
3.1. ATORES	17
3.2. DIAGRAMA DE CASOS DE USO	17
3.3. INSERÇÃO DE EVENTOS	18
3.3.1. FLUXO DE EVENTOS	18
3.3.2. PRÉ CONDIÇÕES	18

3.3.3.	PÓS CONDIÇÕES.....	18
3.4.	ALTERAÇÃO DE DADOS DE EVENTO	19
3.4.1.	FLUXO DE EVENTOS	19
3.4.2.	PRÉ CONDIÇÕES	19
3.4.3.	PÓS CONDIÇÕES	19
3.5.	VISUALIZAÇÃO DE DADOS DE EVENTO	20
3.5.1.	FLUXO DE EVENTOS	20
3.5.2.	PRÉ CONDIÇÕES	20
3.5.3.	PÓS CONDIÇÕES	20
3.6.	EXCLUSÃO DE EVENTO.....	21
3.6.1.	FLUXO DE EVENTOS	21
3.6.2.	PRÉ CONDIÇÕES	21
3.6.3.	PÓS CONDIÇÕES	21
4.	CADASTRO DE CLIENTES	22
4.1.	ATORES	22
4.2.	DIAGRAMA DE CASO DE USO	22
4.3.	INSERÇÃO DE CLIENTES	23
4.3.1.	FLUXO DE EVENTOS	23
4.3.2.	PRÉ CONDIÇÕES	23
4.3.3.	PÓS CONDIÇÕES	23
4.4.	ALTERAÇÃO DE DADOS DE CLIENTES	24
4.4.1.	FLUXO DE EVENTOS	24
4.4.2.	PRÉ CONDIÇÕES	24
4.4.3.	PÓS CONDIÇÕES	24
4.5.	VISUALIZAÇÃO DE DADOS DE CLIENTE	25
4.5.1.	FLUXO DE EVENTOS	25
4.5.2.	PRÉ CONDIÇÕES	25
4.5.3.	PÓS CONDIÇÕES	25
4.6.	INATIVAÇÃO DE CLIENTE	26
4.6.1.	FLUXO DE EVENTOS	26
4.6.2.	PRÉ CONDIÇÕES	26
4.6.3.	PÓS CONDIÇÕES	26
5.	SOLICITAÇÃO DE IDENTIFICAÇÃO NO SISTEMA	27
5.1.	ATORES	27
5.2.	DIAGRAMAS DE CASOS DE USO	27
5.3.	FLUXO DE EVENTOS	27
6.1.	PRÉ CONDIÇÕES	28

	<p>SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA POR PLATAFORMA HETEROGÊNEA (J2EE E .NET)</p> <p>Documento Integrado de Casos de Uso</p>	
--	---	--

6.2. PÓS CONDIÇÕES	28
7. ENTRADA NO ESTABELECIMENTO	29
7.1. ATORES	29
7.2. DIAGRAMA DE CASO DE USO	29
7.3. FLUXO DE EVENTOS	29
7.4. PRÉ CONDIÇÕES	30
7.5. PÓS CONDIÇÕES	30
8. VISUALIZAÇÃO DO TOTAL DO CLIENTE	31
8.1. ATORES	31
8.2. DIAGRAMA DE CASO DE USO	31
8.3. FLUXO DE EVENTOS	31
8.4. PRÉ CONDIÇÕES	32
8.5. PÓS CONDIÇÕES	32
9. SOLICITAÇÃO DE SAÍDA DO ESTABELECIMENTO	33
9.1. ATORES	33
9.2. DIAGRAMA DE CASO DE USO	33
9.3. FLUXO DE EVENTOS	33
9.4. PRÉ CONDIÇÕES	33
9.5. PÓS CONDIÇÕES	34
10. CADASTRO DE CATEGORIA	34
10.1. ATORES	34
10.2. DIAGRAMA DE CASOS DE USO	34
10.3. INSERÇÃO DE CATEGORIA	35
10.3.1. FLUXO DE EVENTOS	35
10.3.2. PRÉ CONDIÇÕES	35
10.3.3. PÓS CONDIÇÕES	36
10.4. ALTERAÇÃO DE DADOS DE CATEGORIA	36
10.4.1. FLUXO DE EVENTOS	36
10.4.2. PRÉ CONDIÇÕES	37
10.4.3. PÓS CONDIÇÕES	37
10.5. VISUALIZAÇÃO DE DADOS DE CATEGORIA	37
10.5.1. FLUXO DE EVENTOS	37
10.5.2. PRÉ CONDIÇÕES	37
10.5.3. PÓS CONDIÇÕES	38
11. CADASTRO DE PRODUTO DE CONSUMO	38
11.1. ATORES	38
11.2. DIAGRAMA DE CASOS DE USO	38

11.3.	INSERÇÃO DE PRODUTO DE CONSUMO	39
11.3.1.	FLUXO DE EVENTOS	39
11.3.2.	PRÉ CONDIÇÕES	39
11.3.3.	PÓS CONDIÇÕES	39
11.4.	ALTERAÇÃO DE DADOS DE PRODUTO DE CONSUMO	40
11.4.1.	FLUXO DE EVENTOS	40
11.4.2.	PRÉ CONDIÇÕES	40
11.4.3.	PÓS CONDIÇÕES	41
11.5.	VISUALIZAÇÃO DE DADOS DE PRODUTO DE CONSUMO	41
11.5.1.	FLUXO DE EVENTOS	41
11.5.2.	PRÉ CONDIÇÕES	41
11.5.3.	PÓS CONDIÇÕES	42
11.6.	EXCLUSÃO DE PRODUTO DE CONSUMO	42
11.6.1.	FLUXO DE EVENTOS	42
11.6.2.	PRÉ CONDIÇÕES	42
11.6.3.	PÓS CONDIÇÕES	42
12.	CADASTRO DE PEDIDOS	43
12.1.	ATORES	43
12.2.	DIAGRAMA DE CASO DE USO	43
12.3.	INCLUSÃO DE PEDIDOS	44
12.3.1.	FLUXO DE EVENTOS	44
12.3.2.	PRÉ CONDIÇÕES	44
12.3.3.	PÓS CONDIÇÕES	44
12.4.	CANCELAMENTO DE PEDIDOS	45
12.4.1.	FLUXO DE EVENTOS	45
12.4.2.	PRÉ CONDIÇÕES	45
12.4.3.	PÓS CONDIÇÕES	45
13.	FECHAMENTO DIÁRIO	46
13.1.	ATORES	46
13.2.	DIAGRAMAS DE CASOS DE USO	46
13.3.	FLUXO DE EVENTOS	46
13.4.	PRÉ CONDIÇÕES	47
13.5.	PÓS CONDIÇÕES	47
14.	CADASTRO DE PRODUTOS DE ESTOQUE	48
14.1.	ATORES	48
14.2.	DIAGRAMA DE CASO DE USO	48
14.3.	INSERÇÃO DE PRODUTO DE ESTOQUE	49

	SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA POR PLATAFORMA HETEROGÊNEA (J2EE E .NET) Documento Integrado de Casos de Uso	
--	---	--

14.3.1.	FLUXO DE EVENTOS	49
14.3.2.	PRÉ CONDIÇÕES	49
14.3.3.	PÓS CONDIÇÕES	49
14.4.	EXCLUSÃO DE PRODUTOS DE ESTOQUES	50
14.4.1.	FLUXO DE EVENTOS	50
14.4.2.	PRÉ CONDIÇÕES	50
14.4.3.	PÓS CONDIÇÕES	50
14.5.	ALTERAÇÃO DE DADOS DE PRODUTOS DE ESTOQUE	51
14.5.1.	FLUXO DE EVENTOS	51
14.5.2.	PRÉ CONDIÇÕES	51
14.5.3.	PÓS CONDIÇÕES	51
14.6.	VISUALIZAÇÃO DE PRODUTOS DE ESTOQUE	52
14.6.1.	FLUXO DE EVENTOS	52
14.6.2.	PRÉ CONDIÇÕES	52
14.6.3.	PÓS CONDIÇÕES	52
15.	SOLICITAÇÃO DE PEDIDO DE PRODUTO DE ESTOQUE	53
15.1.	ATORES	53
15.2.	DIAGRAMA DE CASOS DE USO	53
15.3.	SOLICITAÇÃO DE PEDIDO DE PRODUTO DE ESTOQUE MANUAL	54
15.3.1.	FLUXO DE EVENTOS	54
15.3.2.	PRÉ CONDIÇÕES	54
15.3.3.	PÓS CONDIÇÕES	54
16.	CADASTRO DE COMPRA DE PRODUTO DE ESTOQUE	55
16.1.	ATORES	55
16.2.	DIAGRAMA DE CASO DE USO	55
16.3.	INSERÇÃO DE COMPRAS DE PRODUTO DE ESTOQUE	56
16.3.1.	FLUXO DE EVENTOS	56
16.3.2.	PRÉ CONDIÇÕES	56
16.3.3.	PÓS CONDIÇÕES	56
16.4.	ALTERAÇÃO DE DADOS DE COMPRA DE PRODUTOS DE ESTOQUE	57
16.4.1.	FLUXO DE EVENTOS	57
16.4.2.	PRÉ CONDIÇÕES	58
16.4.3.	PÓS CONDIÇÕES	58
16.5.	VISUALIZAÇÃO DE DADOS DE COMPRA DE PRODUTOS DE ESTOQUE	59
16.5.1.	FLUXO DE EVENTOS	59
16.5.2.	PRÉ CONDIÇÕES	59
16.5.3.	PÓS CONDIÇÕES	59

17.	DEVOLUÇÃO DE PRODUTOS DE ESTOQUE	60
17.1.	ATORES	60
17.2.	DIAGRAMA DE CASO DE USO	60
17.3.	FLUXO DE EVENTOS	60
17.4.	PRÉ CONDIÇÕES	61
17.5.	PÓS CONDIÇÕES	61
18.	RELATÓRIO DE INVENTÁRIO DE ESTOQUE	62
18.1.	ATORES	62
18.2.	DIAGRAMA DE CASO DE USO	62
18.3.	FLUXO DE EVENTOS	62
18.4.	PRÉ CONDIÇÕES	63
18.5.	PÓS CONDIÇÕES	63
19.	CADASTRO DE FORNECEDORES	64
19.1.	ATORES	64
19.2.	DIAGRAMA DE CASO DE USO	64
19.3.	INSERÇÃO DE FORNECEDORES	65
19.3.1.	FLUXO DE EVENTOS	65
19.3.2.	PRÉ CONDIÇÕES	65
19.3.3.	PÓS CONDIÇÕES	66
19.4.	EXCLUSÃO DE FORNECEDORES	66
19.4.1.	FLUXO DE EVENTOS	66
19.4.2.	PRÉ CONDIÇÕES	66
19.4.3.	PÓS CONDIÇÕES	66
19.5.	ALTERAÇÃO DE DADOS DE FORNECEDORES	67
19.5.1.	FLUXO DE EVENTOS	67
19.5.2.	PRÉ CONDIÇÕES	67
19.5.3.	PÓS CONDIÇÕES	67
19.6.	VISUALIZAÇÃO DE DADOS DE FORNECEDORES	68
19.6.1.	FLUXO DE EVENTOS	68
19.6.2.	PRÉ CONDIÇÕES	68
19.6.3.	PÓS CONDIÇÕES	68
20.	SOLICITAÇÃO DE BALANÇO	69
20.1.	ATORES	69
20.2.	DIAGRAMA DE CASO DE USO	69
20.3.	FLUXO DE EVENTOS	69
21.	GERAÇÃO DE RELATÓRIOS ESTATÍSTICOS	70
21.1.	ATORES	70

	SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA POR PLATAFORMA HETEROGÊNEA (J2EE E .NET) Documento Integrado de Casos de Uso	
--	---	--

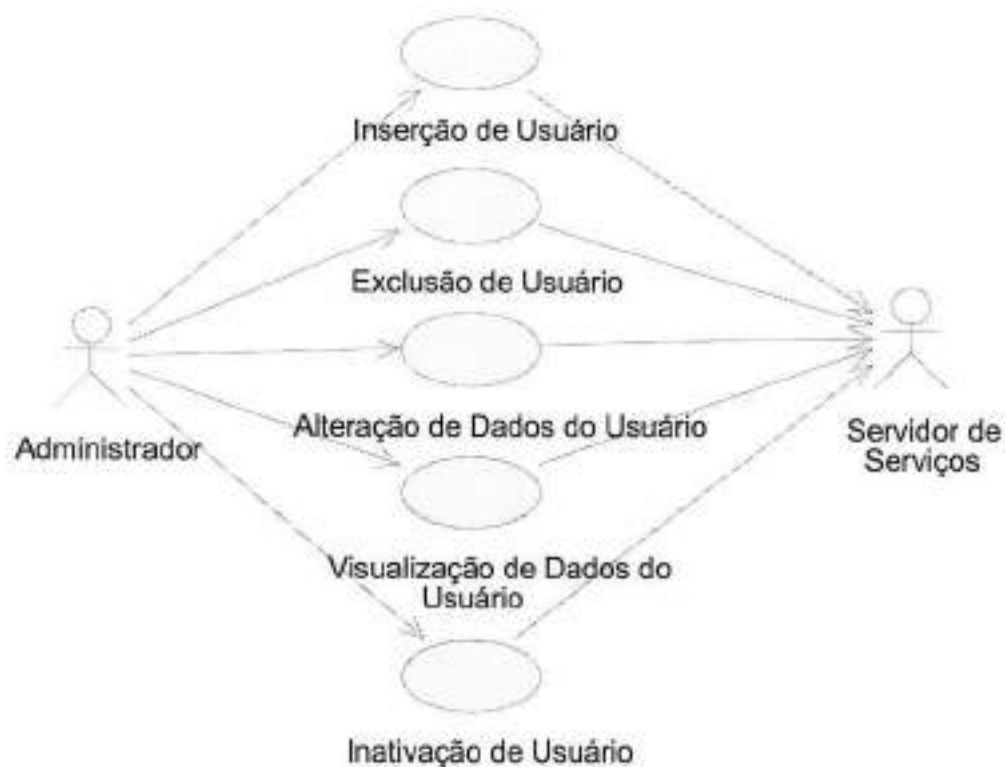
21.2.	DIAGRAMA DE CASO DE USO	70
21.3.	FLUXO DE EVENTOS	70
21.4.	PRÉ CONDIÇÕES	70
21.5.	PÓS CONDIÇÕES	71
22.	GERAÇÃO DE MALA DIRETA	71
22.1.	ATORES	71
22.2.	DIAGRAMA DE CASO DE USO	71
22.3.	FLUXO DE EVENTOS	71
22.4.	PRÉ CONDIÇÕES	72
22.5.	PÓS CONDIÇÕES	72

1. Cadastro de Usuários

1.1. Atores

Administrador.

1.2. Diagrama de Casos de Uso



1.3. Inserção de Usuário

Descreve o processo de inserção de um novo usuário no sistema.

1.3.1. Fluxo de Eventos

1.3.1.1. Fluxo Básico

1. O Usuário solicita a inserção de um novo usuário.
2. O servidor de aplicações solicita os dados relativos ao Usuário.
3. O Usuário insere o nome, username, telefone, CPF e e-mail e dado biométrico do usuário.
4. O servidor de aplicações solicita ao Usuário escolher o perfil do usuário.
5. O Usuário escolhe o perfil do usuário.
6. O servidor de aplicações valida os dados inseridos.
7. O servidor de aplicações envia os dados para o servidor de serviços.
8. O servidor de serviços verifica a existência dos dados recebidos no banco de dados.
9. O servidor de serviços gera um userid e uma senha provisória para o novo usuário e retorna o userid e a senha para o email do usuário.
10. O servidor de serviços armazena os dados no banco de dados e retorna uma mensagem ao servidor de aplicações.
11. O Usuário finaliza a operação de inserção de novo usuário.
12. Fim do caso de uso.

1.3.1.2. Fluxos Alternativos

1.3.1.2.1. Dados inseridos inválidos

Início: Após o item 6 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos".
2. Fim do caso de uso.

1.3.1.2.2. Dados recebidos já existentes

Início: Após o item 8 do fluxo básico.

Condição: O servidor de serviços verifica que os dados recebidos já existem no banco de dados.

1. O servidor de serviços envia a seguinte mensagem ao servidor de aplicações: "Dados inseridos já existentes no sistema. Por favor, tente novamente".
2. O servidor de aplicações exibe a mensagem.
3. Fim de caso de uso.

1.3.2. Pré Condições

Usuário autenticado no sistema.

1.3.3. Pós Condições

O usuário cadastrado e apto a utilizar o sistema.

1.4. Inativação de Usuário

Descreve o processo de inativação de um usuário no sistema.

1.4.1. Fluxo de Eventos

1.4.1.1. Fluxo Básico

1. O Usuário solicita ao servidor de aplicações inativar um usuário.
2. O servidor de aplicações solicita ao servidor de serviços a lista de usuários do sistema.
3. O servidor de serviços busca os usuários no banco de dados e retorna uma lista de usuários ao servidor de aplicações.
4. O servidor de aplicações exibe a lista de usuários.
5. O Usuário escolhe um usuário da lista fornecida pelo servidor de aplicações.
6. O servidor de aplicações pergunta se o Usuário deseja realmente inativar o usuário.
7. O servidor de aplicações manda uma mensagem ao servidor de serviços passando o id do usuário, solicitando que a inativação do usuário.
8. O servidor de serviços inativa o usuário e retorna uma mensagem ao servidor de aplicações.
9. O Usuário finaliza a operação de inativação de usuários.
10. Fim do caso de uso.

1.4.1.2. Fluxos Alternativos

1.4.1.2.1. Usuário selecionado ativo autenticado no sistema

Início: Após o item 8 do fluxo básico.

Condição: O servidor de serviços verifica que o usuário selecionado está autenticado no sistema.

1. O servidor de serviços manda a seguinte mensagem ao servidor de aplicações "Não foi possível a inativação do usuário, pois ele está utilizando o sistema no momento."
2. O servidor de aplicações exibe a seguinte mensagem.
3. Fim do caso de uso.

1.4.2. Pré Condições

Usuário autenticado no sistema.

1.4.3. Pós Condições

O usuário inativo e não- apto a utilizar o sistema.

1.5. Alteração de dados de usuário

Descreve o processo de alteração de dados de um usuário no sistema.

1.5.1. Fluxo de Eventos

1.5.1.1. Fluxo Básico

1. O Usuário solicita ao servidor de aplicações alterar dados de um usuário.
2. O servidor de aplicações solicita ao servidor de serviços a lista de usuários do sistema.
3. O servidor de serviços busca os usuários no banco de dados e retorna uma lista de usuários ao servidor de aplicações.
4. O servidor de aplicações exibe a lista de usuários.
5. O Usuário escolhe um usuário da lista fornecida pelo servidor de aplicações.
6. O Usuário altera algum(ns) atributo(s) do usuário (o nome, username, telefone, CPF ou e-mail) do usuário.
7. O servidor de aplicações valida os dados inseridos.
8. O servidor de aplicações envia os dados para o servidor de serviços, solicitando a alteração dos dados do usuário.
11. O servidor de serviços altera os dados do usuário e retorna uma mensagem ao servidor de aplicações.
12. O Usuário finaliza a operação de alteração de dados de usuário.
13. Fim do caso de uso.

1.5.1.2. Fluxos Alternativos

1.5.1.2.1. Dados inseridos inválidos

Início: Após o item 9 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
2. Fim do caso de uso.

1.5.2. Pré Condições

Usuário autenticado no sistema.

1.5.3. Pós Condições

O usuário escolhido com os dados atualizados.

1.6. Visualização de dados de usuário

Descreve o processo de visualização de dados de um usuário no sistema.

1.6.1. Fluxo de Eventos

1.6.1.1. Fluxo Básico

1. O Usuário solicita ao servidor de aplicações visualizar dados de um usuário.
2. O servidor de aplicações solicita ao servidor de serviços a lista os usuários do sistema.
3. O servidor de serviços busca os usuários no banco de dados e retorna uma lista de usuários ao servidor de aplicações.
4. O servidor de aplicações exibe a lista de usuários.
5. O Usuário escolhe um usuário da lista fornecida pelo servidor de aplicações.
6. O Usuário finaliza a operação de visualização de dados de usuário.
7. Fim do caso de uso.

1.6.2. Pré Condições

Usuário autenticado no sistema.

1.6.3. Pós Condições

O usuário escolhido com os dados visualizados.

1.7. Exclusão de Usuários

Descreve o processo de exclusão de um usuário no sistema.

1.7.1. Fluxo de Eventos

1.7.1.1. Fluxo Básico

1. O Usuário solicita ao servidor de aplicações excluir um usuário.
2. O servidor de aplicações solicita ao servidor de serviços a lista os usuários do sistema.
3. O servidor de serviços busca os usuários no banco de dados e retorna uma lista de usuários ao servidor de aplicações.
4. O servidor de aplicações exibe a lista de usuários.
5. O Usuário escolhe um usuário da lista fornecida pelo servidor de aplicações.
6. O servidor de aplicações pergunta se o Usuário deseja realmente excluir o usuário.
7. O servidor de aplicações manda uma mensagem ao servidor de serviços passando o id do usuário, solicitando que a exclusão do usuário.
8. O servidor de serviços exclui o usuário e retorna uma mensagem ao servidor de aplicações.
9. O Usuário finaliza a operação de exclusão de usuários.
10. Fim do caso de uso.

1.7.1.2. Fluxos Alternativos

1.7.1.2.1. Usuário selecionado ativo autenticado no sistema

Início: Após o item 8 do fluxo básico.

Condição: O servidor de serviços verifica que o usuário selecionado está autenticado no sistema.

1. O servidor de serviços manda a seguinte mensagem ao servidor de aplicações "Não foi possível a exclusão do usuário, pois ele está utilizando o sistema no momento."
2. O servidor de aplicações exibe a seguinte mensagem.
3. Fim do caso de uso.

1.7.2. Pré Condições

Usuário autenticado no sistema.

1.7.3. Pós Condições

O usuário excluído e não-apto a utilizar o sistema.

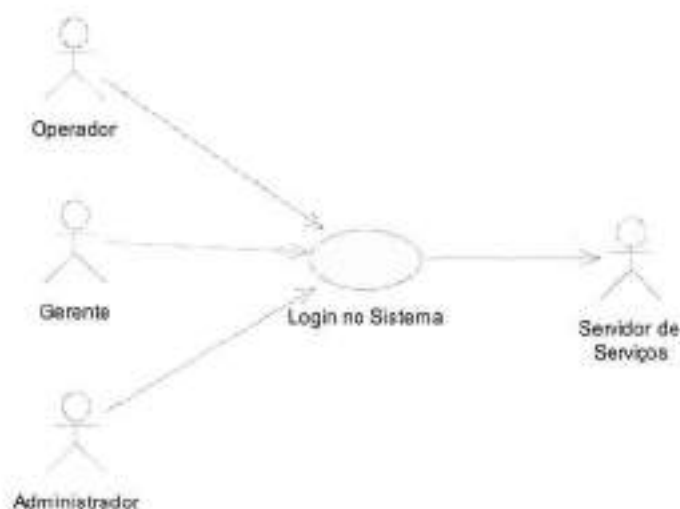
2. Login no Sistema

Este caso de uso descreve o processo de login no sistema pelos usuários.

2.1. Atores

Operador.
Administrador.
Gerente.

2.2. Diagramas de Casos de Uso



2.3. Fluxo de Eventos

2.3.1. Fluxo Básico

1. O usuário solicita o login no sistema.
2. O Servidor de Aplicações solicita o username do usuário.
3. O usuário insere o username.
4. O Servidor de Aplicações envia ao Servidor de Serviços o pedido de login no sistema, passando o username como parâmetro.
5. O Servidor de Serviços verifica se o operador já cadastrou as impressões digitais para acesso ao sistema e retorna uma mensagem.
6. O Servidor de Serviços solicita ao Servidor de Aplicações a digital do usuário.
7. O Servidor de Aplicações solicita que o usuário insira a digital no leitor biométrico de digitais.
8. O Servidor de Aplicações envia a digital para o Servidor de Serviços.
9. O Servidor de Serviços compara a digital recebida com a digital armazenada no banco de dados e retorna uma mensagem ao Servidor de Aplicações.
10. O Servidor de Aplicações retorna uma mensagem ao usuário.
11. Fim do caso de uso.

2.3.2. Fluxos Alternativos

2.3.2.1. O usuário ainda não cadastrou as digitais.

Início: Após o item 5 do fluxo básico.

Condição: O servidor de serviços verifica que o usuário ainda não cadastrou as digitais para acessar o sistema.

1. O servidor de serviços manda a seguinte mensagem para o Servidor de Aplicações "O usuário ainda não cadastrou as digitais para acessar o sistema. Por favor, faça o cadastro."
2. O Servidor de Aplicações exibe a mensagem.
3. É invocado o caso de uso Cadastro de Digitais de Usuários.
4. Fim do caso de uso.

2.3.2.2. Erro na verificação da digital

Início: Após o item 9 do fluxo básico.

Condição: O servidor de serviços verifica que a digital recebida é diferente da digital armazenada.

1. O Servidor de Serviços envia a seguinte mensagem ao servidor de aplicações "A digital inserida não foi identificada. Tente novamente e caso o erro persista consulte o administrador do sistema".
2. O Servidor de Aplicações exibe a mensagem.
3. Fim de caso de uso.

2.4. Pré Condições

Usuário pré-cadastrado no sistema.

2.5. Pós Condições

Usuário autenticado no sistema.

3. Cadastro de Eventos

3.1. Atores

Gerente.

3.2. Diagrama de Casos de Uso



3.3. Inserção de Eventos

Descreve o processo de inserção de um novo evento no sistema.

3.3.1. Fluxo de Eventos

3.3.1.1. Fluxo Básico

1. O Usuário solicita a inserção de um novo evento.
2. O servidor de aplicações solicita os dados relativos do evento ao Usuário.
3. O Usuário insere o nome, a data e a descrição do evento.
4. O servidor de aplicações valida os dados inseridos.
5. O servidor de aplicações envia os dados para o servidor de serviços.
6. O servidor de serviços verifica a existência dos dados recebidos no banco de dados.
7. O servidor de serviços gera um id para o novo.
8. O servidor de serviços armazena os dados no banco de dados e retorna uma mensagem ao servidor de aplicações.
9. O Usuário finaliza a operação de inserção de novo evento.
10. Fim do caso de uso.

3.3.1.2. Fluxos Alternativos

3.3.1.2.1. Dados inseridos inválidos

Início: Após o item 4 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
2. Fim do caso de uso.

3.3.1.2.2. Dados recebidos já existentes

Início: Após o item 6 do fluxo básico.

Condição: O servidor de serviços verifica que os dados recebidos já existem no banco de dados.

1. O servidor de serviços envia a seguinte mensagem ao servidor de aplicações: "Dados inseridos já existentes no sistema. Por favor, tente novamente."
2. O servidor de aplicações exibe a mensagem.
3. Fim de caso de uso.

3.3.2. Pré Condições

Usuário autenticado no sistema.

3.3.3. Pós Condições

O evento cadastrado e apto a ser utilizado pelo o sistema.

3.4. Alteração de dados de evento

Descreve o processo de alteração de dados de um evento no sistema.

3.4.1. Fluxo de Eventos

3.4.1.1. Fluxo Básico

1. O Usuário solicita ao servidor de aplicações alterarem dados de um evento.
2. O servidor de aplicações solicita ao Usuário a entrada de um filtro determinado.
3. O servidor de aplicações solicita ao servidor de serviços a lista dos eventos do sistema, passando o valor do filtro.
4. O servidor de serviços busca os eventos no banco de dados utilizando o filtro e retorna uma lista de eventos ao servidor de aplicações.
5. O servidor de aplicações exibe a lista de eventos.
6. O Usuário escolhe um evento da lista fornecida pelo servidor de aplicações.
7. O Usuário altera algum(ns) atributo(s) do evento (o nome, data ou descrição) do evento.
8. O servidor de aplicações valida os dados inseridos.
9. O servidor de aplicações envia os dados para o servidor de serviços, solicitando a alteração dos dados do evento.
10. O servidor de serviços altera os dados do evento e retorna uma mensagem ao servidor de aplicações.
11. O Usuário finaliza a operação de alteração de dados de evento.
12. Fim do caso de uso.

3.4.1.2. Fluxos Alternativos

3.4.1.2.1. Dados inseridos inválidos

Início: Após o item 8 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
2. Fim do caso de uso.

3.4.2. Pré Condições

Usuário autenticado no sistema.

3.4.3. Pós Condições

O evento escolhido com os dados atualizados.

3.5. Visualização de dados de evento

Descreve o processo de visualização de dados de um evento no sistema.

3.5.1. Fluxo de Eventos

3.5.1.1. Fluxo Básico

1. O Usuário solicita ao servidor de aplicações visualizarem dados de um evento.
2. O servidor de aplicações solicita ao Usuário a entrada de um filtro determinado.
3. O servidor de aplicações solicita ao servidor de serviços a lista dos eventos do sistema, passando o filtro como parâmetro.
4. O servidor de serviços busca os eventos no banco de dados de acordo com o filtro e retorna uma lista de eventos ao servidor de aplicações.
5. O servidor de aplicações exibe a lista de eventos.
6. O Usuário escolhe um evento da lista fornecida pelo servidor de aplicações.
7. O Usuário finaliza a operação de visualização de dados de evento.
8. Fim do caso de uso.

3.5.2. Pré Condições

Usuário autenticado no sistema.

3.5.3. Pós Condições

O evento escolhido com os dados visualizados.

3.6. Exclusão de Evento

Descreve o processo de exclusão de um evento no sistema.

3.6.1. Fluxo de Eventos

3.6.1.1. Fluxo Básico

1. O Usuário solicita ao servidor de aplicações excluir um evento.
2. O servidor de aplicações solicita ao usuário a entrada de um filtro determinado.
3. O servidor de aplicações solicita ao servidor de serviços a lista dos eventos do sistema, passando o filtro como parâmetro.
4. O servidor de serviços busca os eventos no banco de dados e retorna uma lista de eventos ao servidor de aplicações.
5. O servidor de aplicações exibe a lista de eventos.
6. O Usuário escolhe um evento da lista fornecida pelo servidor de aplicações.
7. O servidor de aplicações pergunta se o Usuário deseja realmente excluir o evento.
8. O servidor de aplicações manda uma mensagem ao servidor de serviços passando o id do evento, solicitando a exclusão do evento.
9. O servidor de serviços exclui o evento e retorna uma mensagem ao servidor de aplicações.
10. O Usuário finaliza a operação de exclusão de evento.
11. Fim do caso de uso.

3.6.1.2. Fluxos Alternativos

3.6.1.2.1. Evento selecionado com clientes cadastrados

Início: Após o item 9 do fluxo básico.

Condição: O servidor de serviços verifica que o evento possui clientes cadastrados para esse evento.

1. O servidor de serviços manda a seguinte mensagem ao servidor de aplicações "Não foi possível a exclusão do evento, pois ele contém dados relativos a consumo de clientes."
2. O servidor de aplicações exibe a seguinte mensagem.
3. Fim do caso de uso.

3.6.2. Pré Condições

Usuário autenticado no sistema.

3.6.3. Pós Condições

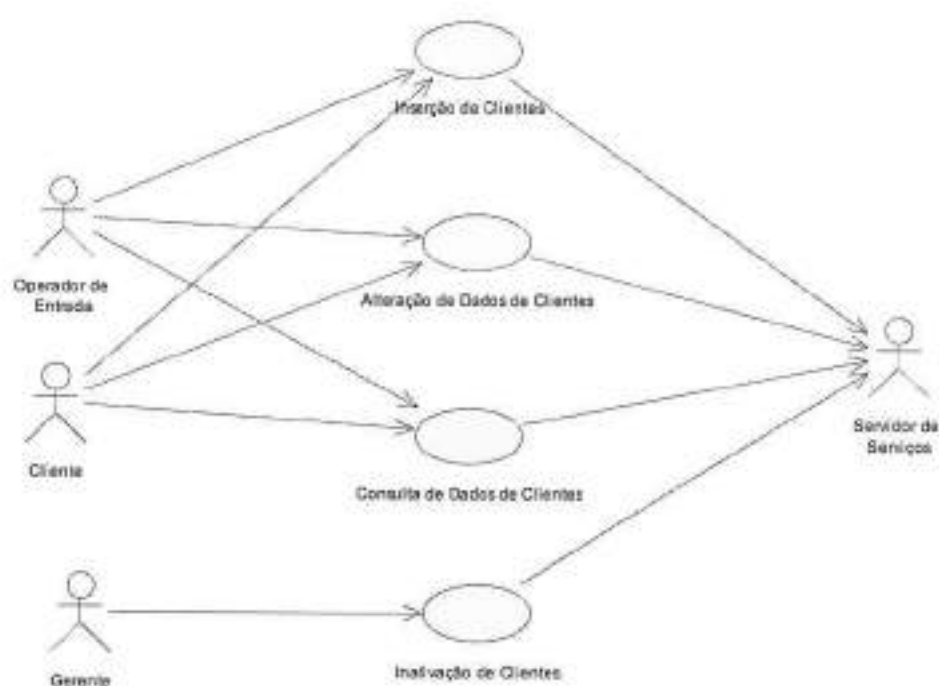
O evento excluído e não-aptos a ser utilizados.

4. Cadastro de Clientes

4.1. Atores

Gerente.
Operador de entrada.
Cliente do estabelecimento.

4.2. Diagrama de Caso de Uso



4.3. Inserção de Clientes

Descreve o processo de inserção de um novo cliente no sistema.

4.3.1. Fluxo de Eventos

4.3.1.1. Fluxo Básico

1. O operador solicita a inclusão de um novo cliente no sistema.
2. O servidor de aplicações solicita os dados relativos ao cliente.
3. O operador insere o nome, telefone, e-mail e data de nascimento do cliente.
4. O servidor de aplicações valida os dados inseridos.
5. O servidor de aplicações envia os dados cadastrais e a impressão digital do cliente para o servidor de serviços.
6. O servidor de serviços armazena os dados cadastrais e a impressão digital do cliente no banco de dados e retorna uma mensagem ao servidor de aplicações.
7. O operador finaliza a operação de inserção de novo cliente.
8. Fim do caso de uso.

4.3.1.2. Fluxos Alternativos

4.3.1.2.1. Dados inseridos inválidos

Início: Após o item 4 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
2. Fim do caso de uso.

4.3.2. Pré Condições

Operador de entrada autenticado no sistema.

O cliente não é identificado no sistema.

Impressão digital já incluída pelo cliente.

4.3.3. Pós Condições

Cliente cadastrado no sistema.

4.4. Alteração de dados de Clientes

Descreve o processo de alteração de dados de um cliente no sistema.

4.4.1. Fluxo de Eventos

4.4.1.1. Fluxo Básico

1. O operador solicita a alteração de um cliente no sistema.
2. O servidor de aplicações solicita ao operador a entrada de um filtro determinado.
3. O operador insere o(s) filtro(s) de seu interesse.
4. O servidor de aplicações solicita ao servidor de serviços a lista dos clientes, passando o valor do filtro.
5. O servidor de serviços busca os clientes no banco de dados utilizando o(s) filtro(s) e retorna uma lista de clientes ao servidor de aplicações.
6. O servidor de aplicações exibe a lista de clientes.
7. O operador escolhe um cliente da lista fornecida pelo servidor de aplicações.
8. O operador altera algum(ns) atributo(s) do cliente (o nome, telefone, e-mail, ou data de nascimento).
9. O servidor de aplicações valida os dados inseridos.
10. O servidor de aplicações envia os dados para o servidor de serviços, solicitando a alteração dos dados do cliente.
11. O servidor de serviços altera os dados do cliente e retorna uma mensagem ao servidor de aplicações.
12. O operador finaliza a operação de alteração de dados de cliente.
13. Fim do caso de uso.

4.4.1.2. Fluxos Alternativos

4.4.1.2.1. Dados inseridos inválidos

Início: Após o item 9 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
2. Fim do caso de uso.

4.4.2. Pré Condições

Operador de entrada autenticado no sistema.

Cliente cadastrado no sistema.

4.4.3. Pós Condições

Cliente selecionado com os dados atualizados.

4.5. Visualização de dados de Cliente

Descreve o processo de visualização de dados de um cliente no sistema.

4.5.1. Fluxo de Eventos

4.5.1.1. Fluxo Básico

1. O operador solicita a visualização de um cliente no sistema.
2. O servidor de aplicações solicita ao operador a entrada de um filtro determinado.
3. O operador insere o(s) filtro(s) de seu interesse.
4. O servidor de aplicações solicita ao servidor de serviços a lista dos clientes, passando o valor do filtro.
5. O servidor de serviços busca os clientes no banco de dados utilizando o(s) filtro(s) e retorna uma lista de clientes ao servidor de aplicações.
6. O servidor de aplicações exibe a lista de clientes.
7. O operador escolhe um cliente da lista fornecida pelo servidor de aplicações.
8. O operador visualiza as informações cadastrais do cliente.
9. O operador finaliza a operação de visualização de dados de cliente.
10. Fim do caso de uso.

4.5.2. Pré Condições

Operador de entrada autenticado no sistema.
O cliente cadastrado no sistema.

4.5.3. Pós Condições

O cliente selecionado com os dados visualizados.

4.6. Inativação de Cliente

Descreve o processo de inativação de um cliente no sistema.

4.6.1. Fluxo de Eventos

4.6.1.1. Fluxo Básico

1. O gerente solicita ao servidor de aplicações a inativação de um cliente.
2. O servidor de aplicações solicita ao gerente a entrada de um filtro determinado.
3. O gerente insere o(s) filtro(s) de seu interesse.
4. O servidor de aplicações solicita ao servidor de serviços a lista dos clientes, passando o valor do filtro.
5. O servidor de serviços busca os clientes no banco de dados utilizando o(s) filtro(s) e retorna uma lista de clientes ao servidor de aplicações.
6. O servidor de aplicações exibe a lista de clientes.
7. O gerente escolhe um cliente da lista fornecida pelo servidor de aplicações.
8. O servidor de aplicações pergunta se o gerente deseja realmente inativar o cliente.
9. O servidor de aplicações manda uma mensagem ao servidor de serviços passando o id do cliente, solicitando a inativação do cliente.
10. O servidor de serviços muda o status do cliente para Inativo e retorna uma mensagem ao servidor de aplicações.
11. O gerente finaliza a operação de inativação de cliente.
12. Fim do caso de uso.

4.6.2. Pré Condições

Gerente autenticado no sistema.
O cliente cadastrado no sistema.

4.6.3. Pós Condições

O cliente com o status de inativo no sistema.

5. Solicitação de Identificação no Sistema

Este caso de uso descreve o processo de identificação de um cliente no sistema.

5.1. Atores

Operador.
Cliente.

5.2. Diagramas de Casos de Uso



5.3. Fluxo de Eventos

5.3.1. Fluxo Básico

1. O usuário solicita a operação de identificação no sistema.
2. O Servidor de Aplicações solicita que o cliente insira a digital no leitor biométrico.
3. O Servidor de Aplicações faz a leitura da digital do cliente e envia uma mensagem ao Servidor de Serviços, solicitando a identificação do cliente, passando como parâmetro o dado lido pelo leitor biométrico.
4. O Servidor de Serviços busca no banco de dados os dados do cliente utilizando o dado recebido e retorna os dados encontrados ao Servidor de Aplicações.
5. O Servidor de Aplicações mostra os dados do cliente.
6. Fim do caso de uso.

5.3.2. Fluxos Alternativos

5.3.2.1. Cliente não encontrado

Início: Após o item 4 do fluxo básico.

Condição: O servidor de serviços não encontra os dados do cliente no banco de dados.

1. O servidor de serviços manda a seguinte mensagem para o Servidor de Aplicações "Não foi encontrado nenhum cliente com essa digital. Tente novamente."
2. O Servidor de Aplicações exibe a mensagem.

3. Fim do caso de uso.

5.3.2.2. Mais de um cliente encontrado

Início: Após o item 4 do fluxo básico.

Condição: O servidor de serviços não encontra os dados do cliente no banco de dados.

1. O servidor de serviços manda a seguinte mensagem para o Servidor de Aplicações "Foi encontrado mais de um cliente com essa digital. Tente novamente."

2. O Servidor de Aplicações exibe a mensagem.

3. Fim do caso de uso.

5.4. Pré Condições

Operador estar autenticado no sistema.

5.5. Pós Condições

Cliente autenticado no sistema.

6. Entrada no Estabelecimento

Descreve o processo de entrada de um cliente no estabelecimento.

6.1. Atores

Operador de entrada.
Cliente do estabelecimento.

6.2. Diagrama de Caso de Uso



6.3. Fluxo de Eventos

6.3.1. Fluxo Básico

1. Operador de entrada solicita a entrada de um cliente no estabelecimento.
2. O servidor de aplicação solicita a inserção da impressão digital do cliente no leitor biométrico de digitais.
3. O cliente insere a impressão digital no leitor biométrico de digitais.
4. O servidor de aplicação captura a informação da impressão digital e envia para o servidor de serviços.
5. O servidor de serviços valida a informação da biometria e retorna ao servidor de aplicação uma mensagem.
6. O servidor de aplicação solicita a confirmação da entrada do cliente na casa.
7. O operador de entrada confirma ao servidor de aplicação a entrada do cliente no estabelecimento.
8. O servidor de aplicação envia ao servidor de serviços a confirmação de entrada do cliente.
9. O servidor de serviços associa o cliente ao evento do dia.
10. Fim do caso de uso.

	<p>SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA POR PLATAFORMA HETEROGÊNEA (J2EE E .NET)</p> <p>Documento Integrado de Casos de Uso</p>	
--	---	--

6.3.2. Fluxos Alternativos

6.3.2.1. Cliente não identificado no servidor de serviços

Início: Após o item 5 do fluxo básico.

Condição: o servidor de serviços não valida a informação da biometria do cliente.

1. É invocado o caso de uso Inserção de Clientes.
2. Retorna ao passo 6 do fluxo básico.

6.3.2.2. Cliente com dados desatualizados

Início: Após o item 5 do fluxo básico.

Condição: as informações do cliente estão desatualizadas no banco de dados do estabelecimento.

1. É invocado o caso de uso Alteração de Dados de Clientes.
2. Retorna ao passo 6 do fluxo básico.

6.4. Pré Condições

Operador de entrada autenticado no sistema.
O evento do dia cadastrado no sistema

6.5. Pós Condições

O cliente está apto a consumir os eventos e produtos do estabelecimento no evento do dia.

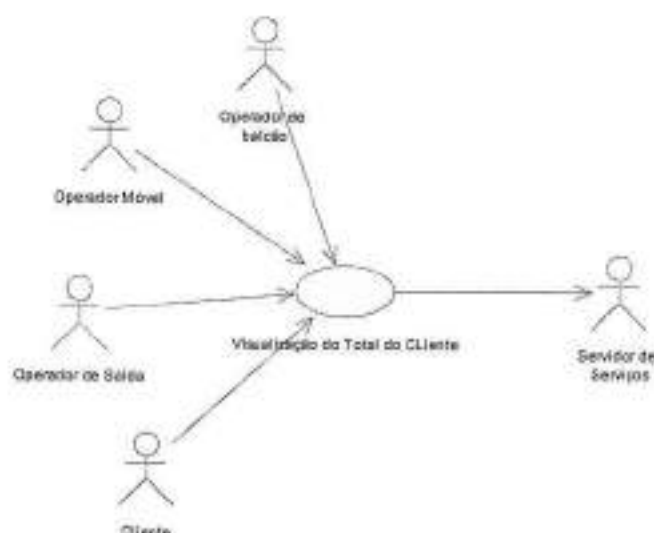
7. Visualização do Total do Cliente

Descreve o processo de visualização do total consumido pelo cliente.

7.1. Atores

Operador de balcão.
Operador móvel.
Operador de saída.
Cliente do estabelecimento.

7.2. Diagrama de Caso de Uso



7.3. Fluxo de Eventos

7.3.1. Fluxo Básico

1. O operador solicita a visualização do total consumido pelo cliente.
2. O servidor de aplicações solicita a inserção da impressão digital do cliente no leitor biométrico de digitais.
3. O cliente insere a impressão digital no leitor biométrico de digitais.
4. O servidor de aplicações captura a informação da impressão digital e envia para o servidor de serviços.
5. O servidor de serviços valida a informação da biometria do cliente.
6. O servidor de serviços calcula o total consumido pelo cliente.
7. O servidor de serviços retorna ao servidor de aplicações as informações do cliente, os produtos consumidos e o total consumido.
8. O servidor de aplicações apresenta todas as informações do cliente.
9. O servidor de aplicações lista os produtos consumidos pelo cliente e apresenta o total.

	SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA POR PLATAFORMA HETEROGÊNEA (J2EE E .NET) Documento Integrado de Casos de Uso	
--	---	--

10. Fim do caso de uso.

7.3.2. Fluxos Alternativos

7.3.2.1. Cliente não identificado no servidor de serviços

Início: Após o item 5 do fluxo básico.

Condição: o servidor de serviços não valida a informação da biometria do cliente.

1. É invocado novamente o Caso de Uso de Visualização do Total do Cliente.

7.4. Pré Condições

Operador de balcão autenticado no sistema.

Operador móvel autenticado no sistema.

Operador de saída autenticado no sistema.

Cliente estar vinculado ao evento do dia.

7.5. Pós Condições

O sistema apresentar as informações do cliente, a discriminação dos produtos e o total consumido pelo cliente.

8. Solicitação de saída do estabelecimento

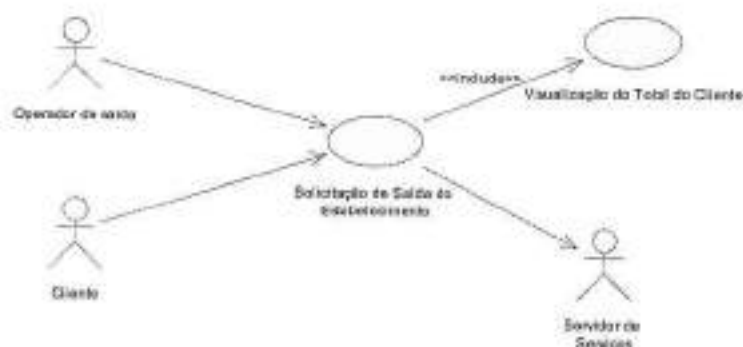
Descreve o processo de saída de um cliente do estabelecimento.

8.1. Atores

Operador de saída.

Cliente do estabelecimento.

8.2. Diagrama de Caso de Uso



8.3. Fluxo de Eventos

8.3.1. Fluxo Básico

1. O sistema invoca o Caso de Uso Visualização do Total do Cliente.
2. O servidor de aplicações apresenta as formas de pagamento aceitas pelo estabelecimento.
3. O operador apresenta ao cliente o valor total a ser pago e as possíveis formas de pagamento.
4. O cliente paga o total consumido.
5. O operador informa ao sistema a forma de pagamento escolhida pelo cliente.
6. O operador confirma o pagamento.
7. O servidor de aplicações envia a confirmação de pagamento ao servidor de serviços.
8. O servidor de serviços fecha a comanda do cliente para o evento do dia.
9. Fim do caso de uso.

8.4. Pré Condições

Operador de saída autenticado no sistema.
Cliente estar vinculado ao evento do dia.

8.5. Pós Condições

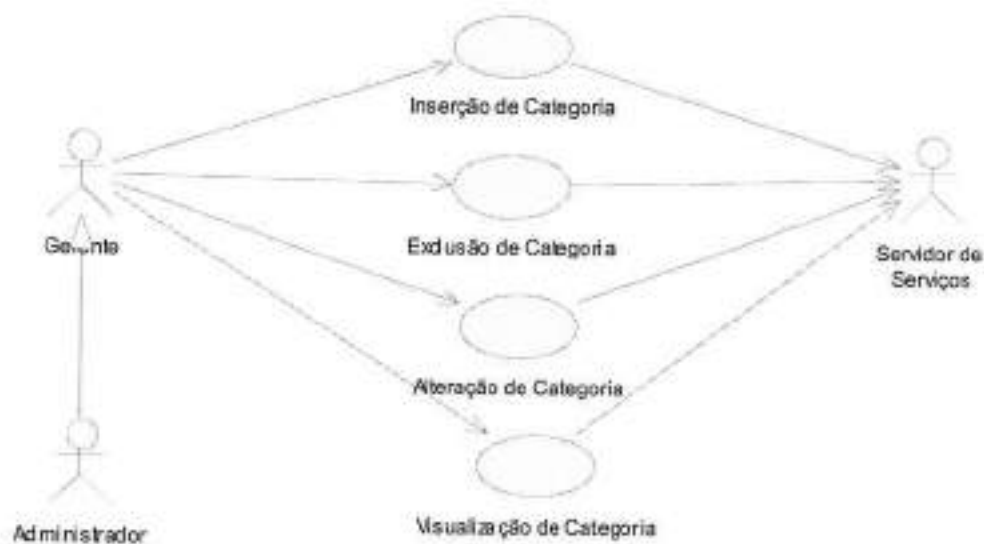
O cliente não está mais presente no evento e sua comanda de consumo está fechada.

9. Cadastro de Categoria

9.1. Atores

Administrador.
Gerente.

9.2. Diagrama de Casos de Uso



9.3. Inserção de Categoria

O usuário do sistema insere um uma nova categoria de produto.

9.3.1. Fluxo de Eventos

9.3.1.1. Fluxo Básico

1. O usuário solicita inserção de uma nova categoria de produto ao sistema.
2. O servidor de aplicação solicita ao usuário os dados relativos à inserção de uma nova categoria de produto:
 - a. Nome da categoria de produto
 - b. Descrição da categoria de produto.
3. O usuário fornece os dados relativos à categoria de produto.
4. O usuário submete os dados ao servidor de aplicações.
5. O servidor de aplicações valida os dados relativos à categoria de produto e envia ao servidor de serviços.
6. O servidor de serviços valida os dados relativos à categoria de produto.
7. O servidor de serviços insere os dados na base de dados.
8. O servidor de serviços retorna uma mensagem de sucesso ao servidor de aplicações.
9. O servidor de aplicações retorna uma mensagem de sucesso ao usuário.
10. Fim do caso de uso.

9.3.1.2. Fluxos Alternativos

9.3.1.2.1. Dados inseridos inválidos

Início: Após o item 7 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
2. Fim do caso de uso.

9.3.1.2.2. Duplicação de Dados

Início: Após o item 6 do fluxo básico.

Condição: O servidor de serviços verifica que os dados recebidos já existem no banco de dados.

1. O servidor de serviços envia a seguinte mensagem ao servidor de aplicações: "Dados inseridos já existentes no sistema. Por favor, tente novamente."
2. O servidor de aplicações exibe a mensagem.
3. Fim de caso de uso.

9.3.2. Pré Condições

Usuário autenticado no sistema e com permissões de Administrador ou Gerente.

9.3.3. Pós Condições

A categoria de produto deve estar disponível no sistema para o cadastro de um novo produto de consumo, cadastro de um novo produto de estoque ou para o cadastro de uma nova subcategoria de produto.

9.4. Alteração de dados de categoria

O usuário do sistema altera dados de uma categoria de produto.

9.4.1. Fluxo de Eventos

9.4.1.1. Fluxo Básico

1. O usuário solicita ao sistema a alteração de uma categoria de produto.
2. O servidor de aplicações solicita ao servidor de serviços a lista das categorias de produto cadastradas no sistema.
3. O servidor de serviços lista do banco de dados as categorias de produto.
4. O servidor de serviços retorna ao servidor de aplicações uma lista com as categorias de produto.
5. O servidor de aplicações exibe ao usuário a lista das categorias de produto.
6. O usuário seleciona a categoria de produto que deseja alterar.
7. O usuário edita os dados da categoria de produto e submete ao servidor de aplicações.
8. O servidor de aplicações solicita ao usuário confirmação de alteração da categoria de produto.
9. O usuário confirma a alteração da categoria de produto.
10. O servidor de aplicações requisita ao servidor de serviços a alteração da categoria de produto.
11. O servidor de serviços altera a categoria de produto do banco de dados.
12. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
13. O servidor de aplicações retorna uma mensagem ao usuário.
14. Fim do caso de uso.

9.4.1.2. Fluxos Alternativos

9.4.1.2.1. Dados inseridos inválidos

Início: Após o item 12 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
1. Fim do caso de uso.
2. Fim de caso de uso.

9.4.1.2.2. O Usuário desiste da operação

Início: Em qualquer ponto dos fluxos básico e alternativo.

Condição: o Usuário desistiu da transação.

1. Fim do caso de uso.

9.4.2. Pré Condições

Usuário autenticado no sistema e com permissões de Administrador ou Gerente.
Existir uma categoria de produto cadastrada no sistema.

9.4.3. Pós Condições

A categoria de produto alterada deve estar disponível para o cadastro de um novo produto de consumo, produto de estoque ou para o cadastro de uma nova subcategoria de produto.

9.5. Visualização de dados de Categoria

O usuário do sistema deseja visualizar uma categoria de produto.

9.5.1. Fluxo de Eventos

9.5.1.1. Fluxo Básico

1. O usuário solicita ao sistema a visualização de uma categoria de produto.
2. O servidor de aplicações solicita ao servidor de serviços a lista das categorias de produto cadastradas no sistema.
3. O servidor de serviços lista do banco de dados as categorias de produto.
4. O servidor de serviços retorna ao servidor de aplicações uma lista com as categorias de produto.
5. O servidor de aplicações exibe ao usuário a lista das categorias de produto.
6. O usuário seleciona a categoria de produto que deseja visualizar.
7. O usuário submete ao servidor de aplicações.
8. O servidor de aplicação exibe os dados da categoria de produto selecionada.
 - a. Nome da categoria de produto.
 - b. Descrição da categoria de produto.
9. Fim do caso de uso.

9.5.1.2. Fluxos Alternativos

9.5.1.2.1. Não existem dados que satisfaçam o filtro aplicado.

Início: Após o item 5 do fluxo básico.

Condição: Não existem dados que satisfaçam o filtro aplicado, o Servidor de Serviços não pôde devolver o resultado da requisição ao servidor de aplicações.

1. Servidor de Serviços envia a seguinte mensagem ao Servidor de Serviços: "Não há categorias que satisfaçam o critério desejado. Por favor, tente novamente".
2. Servidor de aplicações exibe a mensagem.
3. Fim de caso de uso.

9.5.2. Pré Condições

Usuário autenticado no sistema e com permissões de Administrador ou Gerente.
Existir uma categoria de produto cadastrada no sistema.

9.5.3. Pós Condições

A categoria de produto deve estar disponível para visualização.

10. Cadastro de Produto de Consumo

10.1. Atores

Administrador.
Gerente.

10.2. Diagrama de Casos de Uso



10.3. Inserção de Produto de Consumo

O usuário do sistema insere um novo produto de consumo de consumo.

10.3.1. Fluxo de Eventos

10.3.1.1. Fluxo Básico

1. O usuário solicita a inserção de um novo produto de consumo ao sistema.
2. O servidor de aplicação solicita ao usuário os dados relativos à inserção de um produto de consumo:
 - a. Nome do produto de consumo
 - b. Descrição do produto de consumo
 - c. Seleção da categoria de produto a qual pertencerá o produto de consumo.
 - d. O operador cadastra a composição desse produto. É possível compor o produto com ingredientes de estoque. Isso permite realizar o controle de ingredientes de estoque e níveis mínimos de estoque
 - e. Valor de consumo do produto
 - f. Valor de promoção do produto
 - g. Taxa de entrega para esse produto
 - h. Imagem para representar o produto
3. O usuário submete os dados ao servidor de aplicações.
4. O servidor de aplicações valida os dados relativos ao produto de consumo e envia ao servidor de serviços.
5. O servidor de serviços valida os dados relativos ao produto de consumo.
6. O servidor de serviços insere os dados na base de dados.
7. O servidor de serviços retorna uma mensagem de sucesso ao servidor de aplicações.
8. O servidor de aplicações retorna uma mensagem de sucesso ao usuário.
9. Fim do caso de uso.

10.3.1.2. Fluxos Alternativos

10.3.1.2.1. Dados inseridos inválidos

Início: Após o item 4 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
1. Fim do caso de uso.

10.3.2. Pré Condições

Usuário autenticado no sistema e com permissões de Administrador ou Gerente.
Categorias de produto cadastradas no sistema.

10.3.3. Pós Condições

O produto de consumo deve estar cadastrado com sucesso (sistema exibe uma mensagem de sucesso) e disponível para manipulação no sistema.

10.4. Alteração de dados de Produto de Consumo

O usuário do sistema altera dados de um produto de consumo.

10.4.1. Fluxo de Eventos

10.4.1.1. Fluxo Básico

1. O usuário solicita ao sistema a alteração de um produto de consumo.
2. O servidor de aplicação solicita ao usuário o tipo de filtro para listagem dos produtos de consumo sujeitas à alteração.
3. O usuário submete ao servidor de aplicação o tipo de filtro e os dados de busca para produto de consumo.
4. O servidor de aplicações solicita ao servidor de serviços a lista dos produtos de consumo cadastrados no sistema, que satisfazem ao critério de busca.
5. O servidor de serviços lista do banco de dados os produtos de consumo, que satisfazem o critério de busca.
6. O servidor de serviços retorna ao servidor de aplicações uma lista com os produtos de consumo, que satisfazem o critério de filtro.
7. O servidor de aplicações exibe ao usuário a lista dos produtos de consumo.
8. O usuário seleciona o produto de consumo que deseja alterar.
9. O usuário edita os dados do produto de consumo e submete ao servidor de aplicações.
10. O servidor de aplicação solicita ao usuário confirmação de alteração do produto de consumo.
11. O usuário confirma a alteração do produto de consumo.
12. O servidor de aplicações requisita ao servidor de serviços a alteração do produto de consumo.
13. O servidor de serviços altera os dados do produto de consumo do banco de dados.
14. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
15. O servidor de aplicações retorna uma mensagem ao usuário.
16. Fim do caso de uso.

10.4.1.2. Fluxos Alternativos

10.4.1.2.1. Dados Inseridos Inválidos

Início: Após o item 12 do fluxo básico.

Condição: O servidor de aplicações verifica que os dados inseridos são inválidos ou inconsistentes.

1. O servidor de aplicações exibe a seguinte mensagem "Os dados fornecidos estão inconsistentes. Por favor, verifique a validade dos mesmos."
1. Fim do caso de uso.

10.4.2. Pré Condições

Usuário autenticado no sistema e com permissões de Administrador ou Gerente.
Existir um produto de consumo cadastrado no sistema.

10.4.3. Pós Condições

O produto de consumo alterado deve estar disponível para o cadastro de um novo pedido.

10.5. Visualização de dados de Produto de Consumo

O usuário do sistema deseja visualizar um produto de consumo.

10.5.1. Fluxo de Eventos

10.5.1.1. Fluxo Básico

1. O usuário solicita ao sistema a visualização de um produto de consumo.
2. O servidor de aplicação solicita ao usuário o tipo de filtro para listagem dos produtos de consumo deseja visualizar.
3. O usuário submete ao servidor de aplicação o tipo de filtro e os dados de busca para o produto de consumo.
4. O servidor de aplicações solicita ao servidor de serviços a lista dos produtos de consumo cadastrados no sistema, que satisfazem ao critério de busca.
5. O servidor de serviços lista do banco de dados os produtos de consumo, que satisfazem o critério de busca.
6. O servidor de serviços retorna ao servidor de aplicações uma lista com os produtos de consumo, que satisfazem o critério de filtro.
7. O servidor de aplicações exibe ao usuário a lista dos produtos de consumo.
8. O usuário seleciona o produto de consumo que deseja visualizar.
9. O servidor de aplicação exibe os dados do produto de consumo selecionado:
 - a. Nome / Descrição / Seleção da categoria de produto a qual pertencerá o produto de consumo.
 - b. O operador cadastra a composição desse produto. É possível compor o produto com ingredientes de estoque. Isso permite realizar o controle de ingredientes de estoque e níveis mínimos de estoque.
 - c. Valor de consumo do produto / Valor de promoção do produto
10. Fim do caso de uso.

10.5.1.2. Fluxos Alternativos

10.5.1.2.1. Não existem dados que satisfaçam o filtro aplicado.

Início: Após o item 5 do fluxo básico.

Condição: Não existem dados que satisfaçam o filtro aplicado, o Servidor de Serviços não pôde devolver o resultado da requisição ao servidor de aplicações.

1. Servidor de Serviços envia a seguinte mensagem ao Servidor de Serviços: "Não há subcategorias que satisfaçam o critério desejado. Por favor, tente novamente".
2. Servidor de aplicações exibe a mensagem.
3. Fim de caso de uso.

10.5.2. Pré Condições

Usuário autenticado no sistema e com permissões de Administrador ou Gerente.
Existir um produto de consumo cadastrado no sistema.

10.5.3. Pós Condições

Os dados do produto de consumo devem estar disponíveis para visualização.

10.6. Exclusão de Produto de Consumo

O usuário do sistema exclui um produto de consumo de consumo existente.

10.6.1. Fluxo de Eventos

10.6.1.1. Fluxo Básico

1. O usuário solicita ao sistema a exclusão de um produto de consumo.
2. O servidor de aplicação solicita ao usuário o tipo de filtro para listagem dos produtos de consumo sujeitos à exclusão.
3. O usuário submete ao servidor de aplicação o tipo de filtro e os dados de busca para o produto de consumo.
4. O servidor de aplicações solicita ao servidor de serviços a lista dos produtos de consumo cadastrados no sistema, que satisfazem o critério de busca.
5. O servidor de serviços lista do banco de dados os produtos de consumo, que satisfazem o critério de filtro.
6. O servidor de serviços retorna ao servidor de aplicações uma lista com os produtos de consumo, que satisfazem o critério de filtro.
7. O servidor de aplicações exibe ao usuário a lista dos produtos de consumo.
8. O usuário seleciona os produtos de consumo que deseja excluir.
9. O usuário submete ao servidor de aplicações.
10. O servidor de aplicações solicita ao usuário confirmação de exclusão do produto de consumo.
11. O usuário confirma a exclusão do produto de consumo.
12. O servidor de aplicações requisita ao servidor de serviços a exclusão do produto de consumo.
13. O servidor de serviços exclui o produto de consumo do banco de dados.
14. O servidor de aplicações retorna uma mensagem ao usuário.
15. Fim do caso de uso.

10.6.2. Pré Condições

Usuário autenticado no sistema e com permissões de Administrador ou Gerente.
Existir um produto de consumo cadastrado no sistema.

10.6.3. Pós Condições

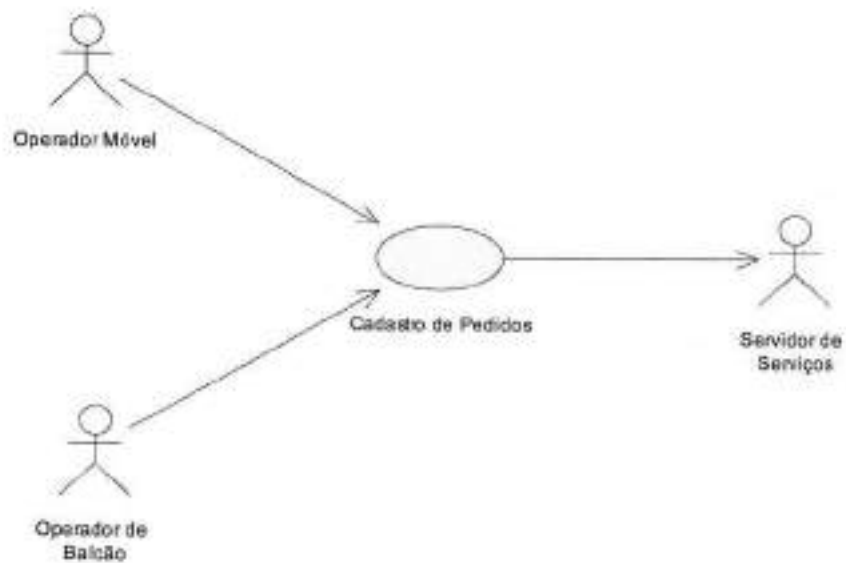
O produto de consumo deverá ser removido do consumo e não estar mais disponível para o usuário.

11. Cadastro de Pedidos

11.1. Atores

Operador de Balcão.
Operador Móvel.

11.2. Diagrama de Caso de Uso



11.3. Inclusão de Pedidos

Descreve o processo de inclusão de um novo pedido no sistema.

11.3.1. Fluxo de Eventos

11.3.1.1. Fluxo Básico

1. O operador solicita a inclusão de um novo pedido no sistema.
2. O servidor de aplicações envia a solicitação de inclusão para o servidor de serviços.
3. O servidor de serviços retorna os produtos de consumo disponíveis no evento do dia.
4. O servidor de aplicações apresenta os produtos de consumo disponíveis no evento do dia.
5. O operador seleciona o(s) produto(s) solicitado(s) pelo cliente.
6. O servidor de aplicações solicita a inserção da impressão digital do cliente no leitor biométrico de digitais para a confirmação do pedido.
7. O cliente insere a impressão digital no leitor biométrico de digitais.
8. O servidor de aplicações captura a informação da impressão digital do cliente.
9. O servidor de aplicações envia a impressão digital do cliente e o pedido para o servidor de serviços.
10. O servidor de serviços valida a informação da biometria do cliente.
11. O servidor de serviços associa o(s) produto(s) solicitado(s) a comanda do cliente e retorna uma mensagem ao servidor de aplicações.
12. O operador finaliza a operação de inserção de novo produto.
13. Fim do caso de uso.

11.3.1.2. Fluxos Alternativos

11.3.1.2.1. Cliente não identificado no servidor de serviços

Início: Após o item 10 do fluxo básico.

Condição: o servidor de serviços não valida a informação da biometria do cliente.

1. O cliente insere novamente a impressão digital no leitor biométrico de digitais.
2. Fim de caso de uso.

11.3.2. Pré Condições

Operador de balcão autenticado no sistema.
Operador móvel autenticado no sistema.
Cliente estar vinculado ao evento do dia.

11.3.3. Pós Condições

Pedidos realizados e cadastrados na comanda do cliente.

11.4. Cancelamento de Pedidos

Descreve o processo de cancelamento de pedidos realizados.

11.4.1. Fluxo de Eventos

11.4.1.1. Fluxo Básico

1. O operador solicita o cancelamento de um pedido no sistema.
2. O servidor de aplicações solicita a inserção da Impressão digital do cliente no leitor biométrico de digitais.
3. O cliente insere a impressão digital no leitor biométrico de digitais.
4. O servidor de aplicações captura a informação da impressão digital e envia para o servidor de serviços.
5. O servidor de serviços valida a informação da biometria do cliente.
6. O servidor de serviços retorna o último pedido realizado pelo cliente.
7. O servidor de aplicações apresenta o pedido a ser cancelado.
8. O operador confirma o cancelamento.
9. O servidor de aplicações envia para o servidor de serviços a confirmação de cancelamento de pedido.
10. O servidor de serviços cancela o pedido e retorna uma mensagem ao servidor de aplicações.
11. O operador finaliza a operação de cancelamento de pedido.
12. Fim do caso de uso.

11.4.1.2. Fluxos Alternativos

11.4.1.2.1. Nenhum pedido a ser cancelado

Início: Após o item 6 do fluxo básico.

Condição: o servidor de serviços não encontra nenhum pedido realizado pelo cliente para ser cancelado.

1. Fim de caso de uso.

11.4.2. Pré Condições

Operador de balcão autenticado no sistema.

Operador móvel autenticado no sistema.

Cliente estar vinculado ao evento do dia.

11.4.3. Pós Condições

Pedido cancelado no sistema.

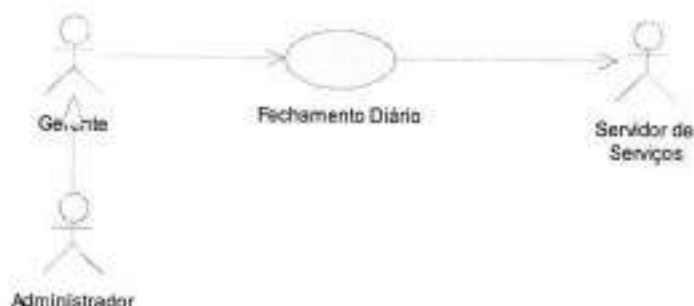
12. Fechamento Diário

O usuário do sistema deseja solicitar o fechamento diário do estabelecimento.

12.1. Atores

Administrador.
Gerente.

12.2. Diagramas de Casos de Uso



12.3. Fluxo de Eventos

12.3.1.1. Fluxo Básico

1. O usuário do sistema inicia a solicitação de geração de relatório de fechamento diário.
2. Servidor de aplicações solicita ao servidor de serviços geração de relatório de fechamento diário:
 - a. Todos os produtos consumidos agrupados por produto e por categoria de produto
 - b. Quantidades
 - c. Total de lucro
 - d. Status de produtos de estoque
 - e. Nome do usuário que gerou o relatório
 - f. Data/Hora de geração do relatório
3. O servidor de aplicações solicita ao servidor de serviços que gere um relatório.
4. O servidor de serviços consulta os dados na base de dados e retorna ao servidor de aplicações os dados do relatório.
5. O servidor de aplicações exibe ao usuário o relatório gerado.
6. O usuário solicita ao sistema impressão do relatório de solicitação de compra de produto de estoque.
7. O servidor de aplicações solicita ao servidor de serviços que a impressão seja realizada.
8. O servidor de serviços realiza a impressão do relatório.
9. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
10. O servidor de aplicações retorna ao usuário uma mensagem.
11. Fim do caso de uso.

12.4. Pré Condições

Usuário autenticado no sistema e com permissões de Gerente ou Administrador.
Existir um estoque e produtos de estoque cadastrados no sistema.

12.5. Pós Condições

Fechamento Diário impresso.

13. Cadastro de Produtos de Estoque

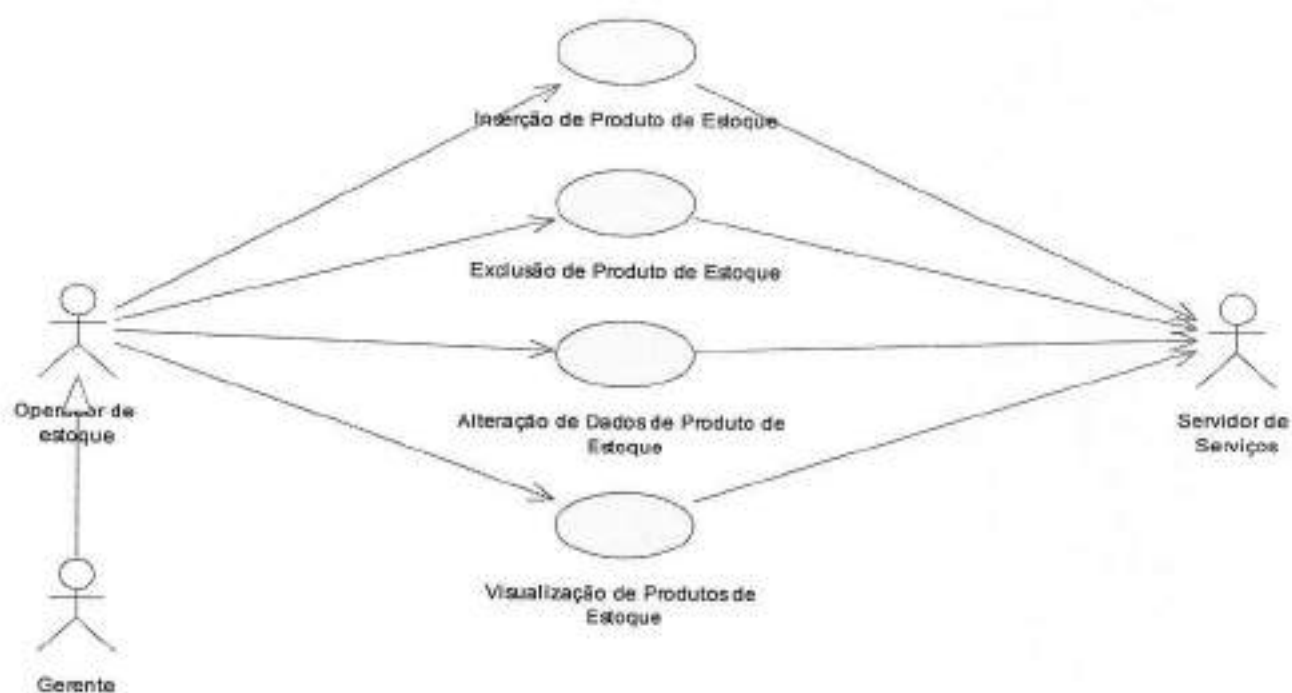
Descreve o processo de cadastro de produto de estoque.

13.1. Atores

Operador de estoque do sistema.

Usuário gerente do sistema.

13.2. Diagrama de Caso de Uso



13.3. Inserção de Produto de Estoque

O usuário insere um novo produto de estoque no sistema.

13.3.1. Fluxo de Eventos

13.3.1.1. Fluxo Básico

1. O usuário solicita o cadastro de um novo produto de estoque no sistema.
2. O servidor de aplicações solicita que o usuário forneça os dados relativos ao produto de estoque:
 - a. Nome do produto.
 - b. Descrição.
 - c. Categoria.
 - d. Unidade de medida.
 - e. Quantidade mínima.
3. O usuário fornece os dados relativos ao produto de estoque.
4. O usuário submete os dados ao servidor de aplicações.
5. O servidor de aplicações valida os dados relativos ao produto de estoque e envia ao servidor de serviços.
6. O servidor de serviços valida os dados relativos ao produto de estoque e insere na base de dados.
7. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
8. O servidor de aplicações retorna uma mensagem ao usuário.
9. Fim do caso de uso.

13.3.1.2. Fluxos Alternativos

13.3.1.2.1. Produto já cadastrado

Início: Passo 6 do fluxo básico

Condição: O produto fornecido já está cadastrado no sistema.

1. O servidor de serviços envia ao servidor de aplicações a mensagem "Nome de produto já cadastrado no sistema"
2. Retorna ao passo 9.

13.3.2. Pré Condições

Usuário autenticado no sistema.
Categoria já cadastrada no sistema.

13.3.3. Pós Condições

Produto de estoque cadastrado no sistema.

13.4. Exclusão de Produtos de Estoques

O usuário exclui o produto de estoque da casa cadastrado no sistema.

13.4.1. Fluxo de Eventos

13.4.1.1. Fluxo Básico

1. O usuário solicita a exclusão de um produto de estoque cadastrado no sistema.
2. O servidor de aplicações solicita ao servidor de serviços a lista de categoria de produtos de estoque cadastrados no sistema.
3. O servidor de serviços lista do banco de dados as categorias de produto de estoque cadastradas no sistema.
4. O servidor de serviços retorna ao servidor de aplicações as categorias dos produtos de estoque.
5. O servidor de aplicações solicita ao usuário que selecione a categoria de produto de estoque que se encontra o produto que se deseja excluir.
6. O usuário seleciona a categoria de produto de estoque que se encontra o produto que se deseja excluir e submete ao servidor de aplicações.
7. O servidor de aplicações solicita ao servidor de serviços a lista dos produtos cadastrados na categoria selecionada pelo usuário.
8. O servidor de serviços consulta no banco de dados a lista de produtos cadastrados na categoria.
9. O servidor de serviços retorna ao servidor de aplicações a lista de produtos cadastrados na categoria.
10. O servidor de aplicações solicita ao usuário que selecione os produtos a serem excluídos do estoque.
11. O usuário seleciona os produtos a serem excluídos e submete ao servidor de aplicações.
12. O servidor de aplicações solicita ao servidor de serviços a exclusão da lista de produtos selecionada pelo usuário.
13. O servidor de serviços valida a lista enviada pelo servidor de aplicações.
14. O servidor de serviços exclui da base de dados os produtos da lista.
15. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
16. O servidor de aplicações exibe ao usuário a mensagem enviada pelo servidor de serviços.
17. Fim do caso de uso.

13.4.2. Pré Condições

Usuário autenticado no sistema.

Produto de estoque cadastrado no sistema.

13.4.3. Pós Condições

Produto de estoque excluído do sistema.

13.5. Alteração de Dados de Produtos de Estoque

O usuário altera os dados de um fornecedor cadastrado no sistema.

13.5.1. Fluxo de Eventos

13.5.1.1. Fluxo Básico

1. O usuário solicita a alteração de dados de um produto de estoque cadastrado no sistema.
2. O servidor de aplicações solicita ao servidor de serviços a lista de categoria de produtos de estoque cadastrados no sistema.
3. O servidor de serviços lista do banco de dados as categorias de produto de estoque cadastradas no sistema.
4. O servidor de serviços retorna ao servidor de aplicações a lista das categorias dos produtos de estoque.
5. O servidor de aplicações solicita ao usuário que selecione a categoria de produto de estoque que se encontra o produto que se deseja alterar os dados.
6. O usuário seleciona a categoria de produto de estoque que se encontra o produto que se deseja alterar os dados.
7. O servidor de aplicações solicita ao servidor de serviços a lista dos produtos cadastrados na categoria selecionada pelo usuário.
8. O servidor de serviços consulta no banco de dados a lista de produtos cadastrados na categoria.
9. O servidor de serviços retorna ao servidor de aplicações a lista de produtos cadastrados na categoria.
10. O servidor de aplicações solicita ao usuário que selecione o produto a ser alterado os dados.
11. O servidor de aplicações permite que o usuário altere os seguintes dados do produto de estoque:
 - a. Nome do produto.
 - b. Descrição.
 - c. Categoria.
 - d. Unidade de medida.
 - e. Quantidade mínima.
12. O usuário altera os dados e submete ao servidor de aplicações.
13. O servidor aplicações valida os dados alterados e solicita ao servidor de serviços a alteração dos dados do produto editado.
14. O servidor de serviços valida os dados do produto e altera na base de dados.
15. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
16. O servidor de aplicações exibe uma mensagem ao usuário.
17. Fim do caso de uso.

13.5.2. Pré Condições

Usuário autenticado no sistema.

Produto de estoque cadastrado no sistema.

13.5.3. Pós Condições

Produto de estoque com os dados alterados no sistema.

13.6. Visualização de Produtos de Estoque

O usuário visualiza os produtos de estoque cadastrados no sistema.

13.6.1. Fluxo de Eventos

13.6.1.1. Fluxo Básico

1. O usuário solicita a consulta de dados de produtos de estoque do sistema.
2. O servidor de aplicações solicita ao servidor de serviços a lista de categorias de produto de estoque cadastradas no sistema.
3. O servidor de serviços lista do banco de dados as categorias de produto de estoque cadastradas no sistema.
4. O servidor de serviços retorna ao servidor de aplicações a lista das categorias de produto de estoque.
5. O servidor de aplicações solicita ao usuário que selecione a categoria de produto de estoque que se encontra o produto que se deseja alterar os dados.
6. O usuário seleciona a categoria de produto de estoque que se encontra o produto que se deseja alterar os dados.
7. O servidor de aplicações solicita ao servidor de serviços a lista dos produtos cadastrados na categoria selecionada pelo usuário.
8. O servidor de serviços consulta no banco de dados a lista de produtos cadastrados na categoria.
9. O servidor de serviços retorna ao servidor de aplicações a lista de produtos cadastrados na categoria.
10. O servidor de aplicações exibe ao usuário a lista dos produtos de estoque cadastrados na categoria com os seguintes campos:
 - a. Nome do produto.
 - b. Descrição.
 - c. Categoria.
 - d. Unidade de medida.
 - e. Quantidade mínima.
 - f. Quantidade no estoque.
11. Fim do caso de uso.

13.6.2. Pré Condições

Usuário autenticado no sistema.
Produtos de estoque cadastrados no sistema.

13.6.3. Pós Condições

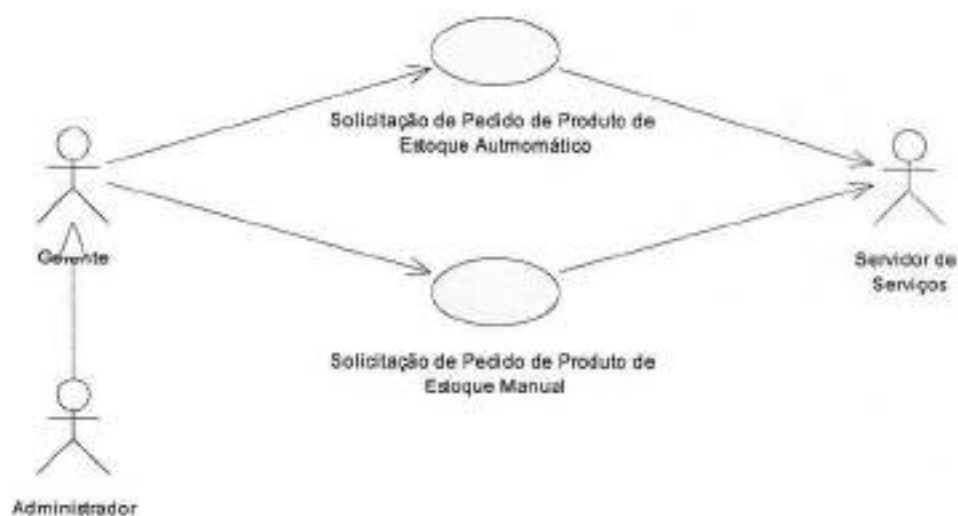
Dados dos produtos de estoque na tela.

14. Solicitação de Pedido de Produto de Estoque

14.1. Atores

Administrador
Gerente.

14.2. Diagrama de Casos de Uso



14.3. Solicitação de Pedido de Produto de Estoque Manual

O usuário do sistema deseja solicitar o pedido de compras dos produtos de estoque de forma manual.

14.3.1. Fluxo de Eventos

14.3.1.1. Fluxo Básico

1. O usuário do sistema inicia a solicitação de pedido de compra de produtos manual.
2. O servidor de aplicações solicita ao servidor de serviços uma lista com todos os produtos de estoque que estão cadastrados no sistema.
3. O servidor de serviços lista do banco de dados os produtos de estoque do sistema, separando-os por categoria.
4. O servidor de serviços retorna ao servidor de aplicações a lista com todos os produtos de estoque.
5. O usuário seleciona os produtos que deseja realizar compra, mesmo não estando abaixo de estoque mínimo.
6. O usuário confirma o desejo de gerar um relatório de compra.
7. O servidor de aplicações solicita ao servidor de serviços que gere um relatório com os produtos de estoque selecionados.
8. O servidor de serviços consulta na base de dados os dados referentes aos produtos de estoque e retorna ao servidor de aplicações os dados do relatório:
 - a. Listagem dos produtos de estoque
 - b. Valor atual do produto de estoque
 - c. Quantidade atual do produto de estoque
 - d. Fornecedores para os produtos de estoque
 - e. Nome do usuário que gerou o relatório
 - f. Data/Hora em que o relatório foi gerado
9. O servidor de aplicações exibe ao usuário o relatório gerado.
10. O usuário solicita ao sistema impressão do relatório de solicitação de compra de produto de estoque.
11. O servidor de aplicações realiza a impressão do documento.
12. O servidor de aplicações retorna ao usuário uma mensagem.
13. Fim do caso de uso.

14.3.2. Pré Condições

Usuário autenticado no sistema e com permissões de Gerente ou Administrador.
Existir um estoque e produtos de estoque cadastrados no sistema.

14.3.3. Pós Condições

Produto de Estoque com compra programada.

15. Cadastro de Compra de Produto de Estoque

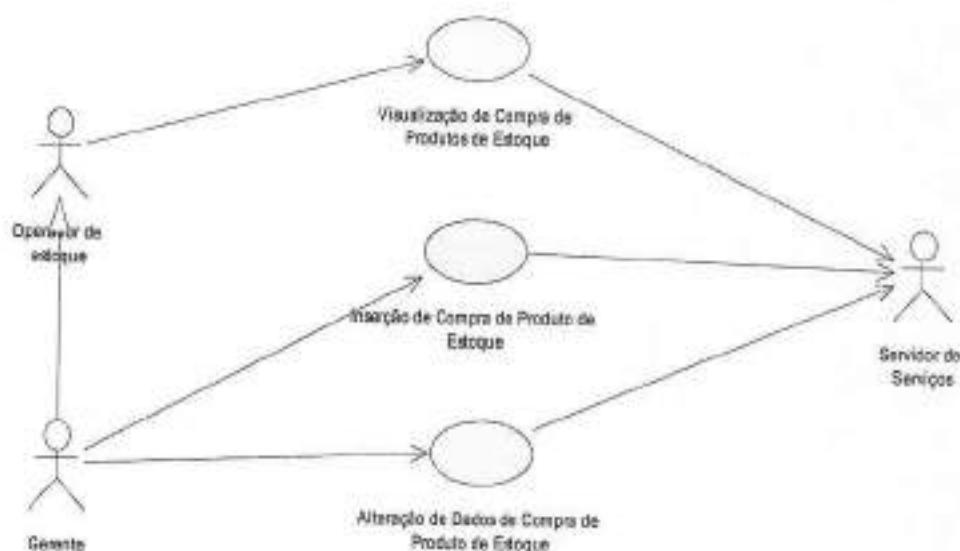
Descreve o processo de cadastro de compra de produto de estoque.

15.1. Atores

Usuário gerente do sistema.

Operador de estoque do sistema.

15.2. Diagrama de Caso de Uso



15.3. Inserção de Compras de Produto de Estoque

O usuário insere os dados uma compra de produtos de estoque para a casa no sistema.

15.3.1. Fluxo de Eventos

15.3.1.1. Fluxo Básico

1. O usuário solicita o cadastro de uma compra de produtos de estoque no sistema.
2. O servidor de aplicações solicita que o gerente forneça os dados relativos ao fornecedor:
 - f. Identificador da Compra
 - g. Fornecedor.
 - h. Data da compra.
 - i. Data da chegada dos produtos.
 - j. Lista de produtos:
 - i. Produto
 - ii. Quantidade solicitada
 - iii. Quantidade entregue
 - iv. Preço unitário
3. O usuário fornece os dados relativos a compra e submete ao servidor de aplicações.
4. O servidor de aplicações valida os dados da compra e solicita ao servidor de serviços o cadastro da compra.
5. O servidor de serviços valida os dados da compra e insere na base de dados.
6. O servidor de serviços retorna uma mensagem ao servidor aplicações.
7. O servidor de aplicações retorna uma mensagem ao usuário contendo o identificador da compra.
8. Fim do caso de uso.

15.3.1.2. Fluxos Alternativos

15.3.1.2.1. Identificador de compra já cadastrado no sistema

Início: Passo 5 do fluxo básico.

Condição: O usuário tenta cadastrar uma compra com um identificador já existente no sistema.

1. Servidor de serviços envia ao servidor de aplicações a mensagem "Identificador de compra já cadastrado no sistema."
2. Retorna ao passo 7.

15.3.2. Pré Condições

Usuário autenticado no sistema.

15.3.3. Pós Condições

Compra de produtos de estoque cadastrada no sistema.

15.4. Alteração de Dados de Compra de Produtos de Estoque

O usuário altera os dados de uma compra de produtos de estoque cadastrada no sistema.

15.4.1. Fluxo de Eventos

15.4.1.1. Fluxo Básico

1. O usuário solicita a alteração de dados de uma compra de produtos de estoque.
2. O servidor de aplicações solicita ao usuário o identificador da compra.
3. O usuário digita o identificador da compra e submete ao servidor de aplicações.
4. O servidor de aplicações solicita ao servidor de serviços os dados relativos ao identificador de compra digitado.
5. O servidor de serviços lista do banco de dados os dados relativos ao identificador de compras fornecido.
6. O servidor de serviços retorna ao servidor de aplicações os dados relativos a compra.
7. O servidor de aplicações exibe ao usuário as informações referentes ao identificador de compras fornecido.
8. O usuário altera os seguintes dados relativos a compra:
 - k. Identificador da Compra
 - l. Fornecedor.
 - m. Data da compra.
 - n. Data da chegada dos produtos.
 - o. Lista de produtos:
 - i. Produto
 - ii. Quantidade solicitada
 - iii. Quantidade entregue
 - iv. Preço unitário
9. O usuário submete ao servidor de aplicações.
10. O servidor de aplicações valida os dados relativos a compra e solicita ao servidor de serviços a alteração dos dados da compra.
11. O servidor de serviços valida os dados relativos a compra e altera na base de dados.
12. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
13. O servidor de aplicações retorna uma mensagem ao usuário.
14. Fim do caso de uso.

15.4.1.2. Fluxos Alternativos

15.4.1.2.1. Identificador de compra já cadastrado no sistema

Início: Passo 5 do fluxo básico.

Condição: O usuário tenta cadastrar uma compra com um identificador já existente no sistema.

1. Servidor de serviços envia ao servidor de aplicações a mensagem "Identificador de compra já cadastrado no sistema."
2. Retorna ao passo 13.

15.4.1.2.2. Identificador de compra não cadastrado no sistema

Início: Passo 5 do fluxo básico.

Condição: O usuário tenta cadastrar uma compra com um identificador já existente no sistema.

1. Servidor de serviços envia ao servidor de aplicações a mensagem "Identificador de compra não encontrado no sistema."
2. Retorna ao passo 13.

15.4.2. Pré Condições

Usuário autenticado no sistema.
Compra cadastrada no sistema.

15.4.3. Pós Condições

Fornecedor com os dados alterados no sistema.

15.5. Visualização de Dados de Compra de Produtos de Estoque

O usuário visualiza os dados relativos a uma compra de produtos de estoque.

15.5.1. Fluxo de Eventos

15.5.1.1. Fluxo Básico

1. O usuário solicita a consulta de dados de uma compra de produtos de estoque no sistema.
2. O servidor de aplicações solicita ao usuário o identificador da compra.
3. O usuário digita o identificador da compra e submete ao servidor de aplicações.
4. O servidor de aplicações consulta solicita ao servidor de serviços os dados relativos ao identificador de compra digitado.
5. O servidor de serviços lista do banco de dados os dados relativos ao identificador de compras fornecido.
6. O servidor de serviços retorna ao servidor de aplicações os dados relativos a compra.
7. O servidor de aplicações exibe ao usuário as informações referentes ao identificador de compras fornecido.
8. Fim do caso de uso.

15.5.1.2. Fluxos Alternativos

15.5.1.2.1. Identificador de compra não cadastrado no sistema

Início: Passo 5 do fluxo básico.

Condição: O usuário tenta cadastrar uma compra com um identificador já existente no sistema.

1. Servidor de serviços envia ao servidor de aplicações a mensagem "Identificador de compra não encontrado no sistema."
2. Retorna ao passo 13.

15.5.2. Pré Condições

Usuário autenticado no sistema.

Compra de produtos de estoque cadastrada no sistema.

15.5.3. Pós Condições

Dados da compra de produtos de estoque na tela do usuário.

16. Devolução de Produtos de Estoque

Descreve o processo de devolução de um produto de estoque.

16.1. Atores

Usuário gerente do sistema.

16.2. Diagrama de Caso de Uso



16.3. Fluxo de Eventos

16.3.1. Fluxo Básico

1. O usuário solicita o cadastro de devolução de um produto de estoque ao fornecedor
2. O servidor de aplicações solicita o identificador da compra em que o produto foi incluído.
3. O usuário digita o identificador da compra e submete ao servidor de aplicações.
4. O servidor de aplicações solicita ao servidor de serviços os dados relativos ao identificador de compra fornecido.
5. O servidor de serviços consulta na base de dados os dados relativos ao identificador de compra fornecido.
6. O servidor de serviços retorna ao servidor de aplicações os dados relativos a compra.
7. O servidor de aplicações exibe ao usuário as seguintes informações referentes a compra:
 - p. Identificador da Compra
 - q. Fornecedor.
 - r. Data da compra.
 - s. Data da chegada dos produtos.
 - t. Lista de produtos:
 - i. Produto
 - ii. Quantidade solicitada
 - iii. Quantidade entregue
 - iv. Preço unitário
8. O usuário seleciona os produtos que deseja cadastrar a devolução digita a quantidade de produtos que deseja devolver.

	SOLUÇÃO DE IDENTIFICAÇÃO BIOMÉTRICA GERENCIADA POR PLATAFORMA HETEROGÊNEA (J2EE E .NET) Documento Integrado de Casos de Uso	
--	---	--

9. O usuário submete ao servidor de aplicações.
10. O servidor de aplicações valida os dados relativos a devolução e solicita ao servidor de serviços o cadastro da devolução ao fornecedor dos produtos.
11. O servidor de serviços valida os dados e insere na base de dados.
12. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
13. O servidor de aplicações exibe a mensagem ao usuário.
14. Fim do caso de uso.

16.3.2. Fluxos Alternativos

16.3.2.1. Identificador de compra não cadastrado no sistema

Início: Passo 5 do fluxo básico.

Condição: O usuário tenta cadastrar uma compra com um identificador já existente no sistema.

1. Servidor de serviços envia ao servidor de aplicações a mensagem "Identificador de compra não encontrado no sistema."
2. Retorna ao passo 13.

16.4. Pré Condições

Usuário autenticado no sistema.

Compra cadastrada no sistema.

16.5. Pós Condições

Devoluções de produtos ao fornecedor cadastrado no sistema.

17. Relatório de inventário de estoque

Descreve o processo de geração de relatórios estatísticos pelo operador do sistema.

17.1. Atores

Operador de estoque do sistema.

Usuário gerente do sistema.

17.2. Diagrama de Caso de Uso



17.3. Fluxo de Eventos

17.3.1. Fluxo Básico

1. O usuário solicita a visualização de inventário de estoque
2. O servidor de aplicações solicita ao servidor de serviços a geração de um relatório de inventário de estoque.
3. O servidor de serviços consulta na base de dados os dados relativos aos produtos do estoque.
4. O servidor de serviços retorna ao servidor de aplicações os dados relativos aos produtos de estoque.
5. O servidor de aplicações exibe na tela do usuário um relatório contendo as informações:
 - a. Categoria do Produto:
 - i. Nome do produto.
 - ii. Descrição.
 - iii. Unidade de medida.
 - iv. Quantidade mínima.
 - v. Quantidade no estoque.
 - vi. Preço de custo médio.
6. Fim do caso de uso.

17.3.2. Fluxos Alternativos

17.3.2.1. Usuário deseja consultar os detalhes de entradas e saídas do produto

Início: Após o item 5 do fluxo básico.

Condição: O operador deseja consultar os detalhes de entrada e saída do produto de estoque.

1. O servidor de aplicações solicita ao servidor de serviços a consulta do relatório de detalhes de entrada e saída do produto de estoque passando o código do produto.
2. O servidor de serviços consulta na base de dados os detalhes de inventário do produto.
3. O servidor de serviços retorna ao servidor de aplicações o relatório de detalhes do produto.
4. O servidor de aplicações exibe na tela:
 - a. Nome do produto.
 - b. Descrição.
 - c. Categoria.
 - d. Unidade de medida.
 - e. Quantidade mínima.
 - f. Quantidade no estoque.
 - g. Preço de custo médio.
 - h. Dados de entrada e saída:
 - i. Data do movimento.
 - ii. Histórico
 - iii. Entradas:
 1. Quantidade
 2. Custo Unitário
 3. Total
 - iv. Saídas:
 1. Quantidade
 2. Custo médio
 3. Total
 - v. Saldos:
 1. Quantidade
 2. Custo médio
 3. Total
5. Fim do caso de uso.

17.4. Pré Condições

Usuário autenticado no sistema.

Dados de produtos de estoque cadastrado no sistema.

17.5. Pós Condições

Relatório gerado na tela do operador.

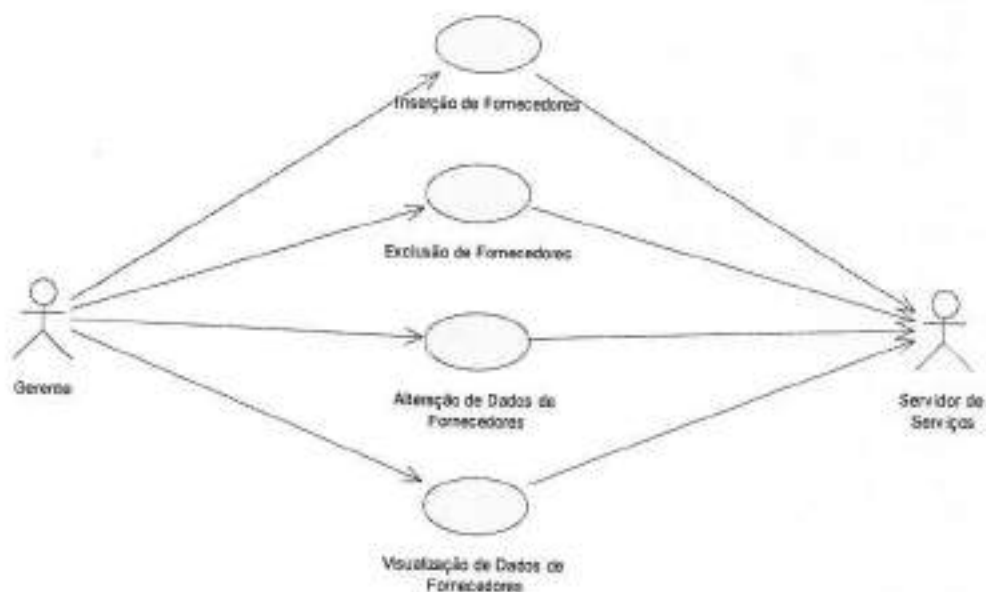
18. Cadastro de Fornecedores

Descreve o processo de cadastro de fornecedores no sistema.

18.1. Atores

Usuário gerente do sistema.

18.2. Diagrama de Caso de Uso



18.3. Inserção de Fornecedores

O usuário gerente insere um novo fornecedor de produtos para a casa no sistema.

18.3.1. Fluxo de Eventos

18.3.1.1. Fluxo Básico

1. O usuário gerente solicita o cadastro de um novo fornecedor no sistema.
2. O servidor de aplicações solicita que o usuário gerente forneça os dados relativos ao fornecedor:
 - a. Nome do fornecedor.
 - b. CNPJ.
 - c. Ramo.
 - d. Endereço.
 - e. CEP.
 - f. Cidade.
 - g. Telefone.
 - h. Email.
 - i. Nome para contato.
3. O usuário gerente fornece os dados relativos ao fornecedor.
4. O usuário gerente submete os dados ao servidor de aplicações.
5. O servidor de aplicações valida os dados relativos ao fornecedor e envia ao servidor de serviços.
6. O servidor de serviços valida os dados relativos ao fornecedor.
7. O servidor de serviços insere os dados na base de dados.
8. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
9. O servidor de aplicações retorna uma mensagem ao usuário gerente.
10. Fim do caso de uso.

18.3.1.2. Fluxos Alternativos

18.3.1.2.1. CNPJ já cadastrado

Início: Passo 6 do fluxo básico

Condição: O CNPJ fornecido já está cadastrado no sistema.

1. O servidor de serviços envia ao servidor de aplicações a mensagem "CNPJ já cadastrado no sistema"
2. Retorna ao passo 9.

18.3.1.2.2. O usuário gerente desiste da operação

Início: Em qualquer ponto dos fluxos básicos e alternativos (antes de digitar a senha e submeter a transação)

Condição: o usuário gerente desistiu da transação

1. Fim do caso de uso.

18.3.2. Pré Condições

Usuário gerente autenticado no sistema.

18.3.3. Pós Condições

Fornecedor cadastrado no sistema.

18.4. Exclusão de Fornecedores

O usuário gerente exclui fornecedor da casa cadastrado no sistema.

18.4.1. Fluxo de Eventos

18.4.1.1. Fluxo Básico

1. O usuário gerente solicita a exclusão de um fornecedor cadastrado no sistema.
2. O servidor de aplicações solicita ao servidor de serviços a lista dos fornecedores cadastrados no sistema.
3. O servidor de serviços lista do banco de dados os fornecedores cadastrados no sistema.
4. O servidor de serviços retorna ao servidor de aplicações uma lista com os fornecedores cadastrados no sistema.
5. O servidor de aplicações exibe ao usuário gerente a lista dos fornecedores cadastrados no sistema.
6. O usuário gerente seleciona o fornecedor que deseja excluir.
7. O usuário gerente submete ao servidor de aplicações.
8. O servidor de aplicações pergunta ao usuário se ele deseja realmente excluir o fornecedor.
9. O usuário submete ao servidor de aplicações.
10. O servidor de aplicações requisita ao servidor de serviços a exclusão do fornecedor selecionado.
11. O servidor de serviços exclui o fornecedor do banco de dados.
12. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
13. O servidor de aplicações retorna uma mensagem ao usuário gerente.
14. Fim do caso de uso.

18.4.2. Pré Condições

Usuário gerente autenticado no sistema.
Fornecedor cadastrado no sistema.

18.4.3. Pós Condições

Fornecedor excluído do sistema.

18.5. Alteração de Dados de Fornecedores

O usuário gerente altera os dados de um fornecedor cadastrado no sistema.

18.5.1. Fluxo de Eventos

18.5.1.1. Fluxo Básico

1. O usuário gerente solicita a alteração de dados de um fornecedor.
2. O servidor de aplicações solicita ao servidor de serviços a lista dos fornecedores cadastrados no sistema.
3. O servidor de serviços lista do banco de dados os fornecedores cadastrados no sistema.
4. O servidor de serviços retorna ao servidor de aplicações uma lista com os fornecedores cadastrados no sistema.
5. O servidor de aplicações exibe ao usuário gerente a lista dos fornecedores cadastrados no sistema.
6. O usuário gerente seleciona o fornecedor que deseja alterar os dados.
7. O servidor de aplicações exibe na tela os dados do fornecedor que o usuário gerente deseja alterar:
 - a. Nome do fornecedor.
 - b. Ramo.
 - c. Endereço.
 - d. CEP.
 - e. Cidade.
 - f. Telefone.
8. O usuário gerente altera os dados do fornecedor selecionado.
9. O usuário gerente submete ao servidor de aplicações.
10. O servidor de aplicações requisita ao servidor de serviços a alteração dos dados do fornecedor selecionado.
11. O servidor de serviços altera os dados do fornecedor no banco de dados.
12. O servidor de serviços retorna uma mensagem ao servidor de aplicações.
13. O servidor de aplicações retorna uma mensagem ao usuário gerente.
14. Fim do caso de uso.

18.5.2. Pré Condições

Usuário gerente autenticado no sistema.
Fornecedor cadastrado no sistema.

18.5.3. Pós Condições

Fornecedor com os dados alterados no sistema.

18.6. Visualização de Dados de Fornecedores

O usuário gerente altera os dados de um fornecedor cadastrado no sistema.

18.6.1. Fluxo de Eventos

18.6.1.1. Fluxo Básico

1. O usuário gerente solicita a consulta de dados de um fornecedor do sistema.
2. O servidor de aplicações solicita ao servidor de serviços a lista dos fornecedores cadastrados no sistema.
3. O servidor de serviços lista do banco de dados os fornecedores cadastrados no sistema.
4. O servidor de serviços retorna ao servidor de aplicações uma lista com os fornecedores cadastrados no sistema.
5. O servidor de aplicações exibe ao usuário gerente a lista dos fornecedores cadastrados no sistema.
6. O usuário gerente seleciona o fornecedor que deseja visualizar os dados.
7. O servidor de aplicações exibe na tela os dados do fornecedor que o usuário gerente deseja visualizar:
 - a. Nome do fornecedor.
 - b. CNPJ.
 - c. Ramo.
 - d. Endereço.
 - e. CEP.
 - f. Cidade.
 - g. Telefone.
8. Fim do caso de uso.

18.6.2. Pré Condições

Usuário gerente autenticado no sistema.
Fornecedor cadastrado no sistema.

18.6.3. Pós Condições

Dados do fornecedor na tela do usuário gerente.

19. Solicitação de Balanço

Descreve o processo de visualização do balanço pelo gerente do sistema

19.1. Atores

Gerente do sistema.

19.2. Diagrama de Caso de Uso



19.3. Fluxo de Eventos

19.3.1. Fluxo Básico

1. O gerente solicita a visualização do balanço da casa.
2. O gerente seleciona o período que deseja visualizar o relatório:
 - a. 15 dias.
 - b. 30 dias.
 - c. Período.
3. O gerente seleciona o período desejado.
4. O gerente submete ao servidor de aplicações.
5. O servidor de aplicações solicita ao servidor de serviços o relatório com os dados relativos a opção selecionada.
6. O servidor de serviços valida os dados da solicitação e retorna o relatório ao servidor de aplicações.
7. O servidor de aplicações retorna ao gerente do sistema o relatório solicitado.
8. Fim do caso de uso.

20. Geração de Relatórios Estatísticos

Descreve o processo de geração de relatórios estatísticos pelo operador do sistema.

20.1. Atores

Operador do sistema.

20.2. Diagrama de Caso de Uso



20.3. Fluxo de Eventos

20.3.1. Fluxo Básico

1. O operador solicita a visualização de relatórios estatísticos
2. O servidor de aplicações solicita que o servidor de aplicações selecione a opção de relatório estatístico que ele deseja visualizar:
 - a. Relatório por produtos.
 - b. Relatório por clientes.
3. O operador seleciona a opção de relatório estatístico que deseja visualizar.
4. O operador seleciona o período que deseja visualizar o relatório.
5. O operador submete ao servidor de aplicações.
6. O servidor de aplicações solicita ao servidor de serviços uma lista com os dados relativos a opção de relatório selecionada.
7. O servidor de serviços valida os dados da solicitação e retorna o relatório ao servidor de aplicações.
8. O servidor de aplicações retorna ao operador do sistema o relatório solicitado.
9. Fim do caso de uso.

20.4. Pré Condições

Operador autenticado no sistema.

20.5. Pós Condições

Relatório gerado na tela do operador.

21. Geração de Mala Direta

Descreve o processo de envio de mala direta de um gerente do sistema para os clientes da casa.

21.1. Atores

Usuário gerente do sistema.

21.2. Diagrama de Caso de Uso



21.3. Fluxo de Eventos

21.3.1. Fluxo Básico

1. O gerente solicita o envio de mala direta para os clientes da casa. O servidor de aplicações solicita que o gerente selecione a opção de mala direta que deseja enviar:
 - a. Promoções de festas.
 - b. Aniversariantes.
2. O gerente seleciona a opção de mala direta que deseja enviar.
3. O servidor de aplicações solicita que o cliente selecione a página html a ser enviada.
4. O gerente seleciona a página html a ser enviada.
5. O gerente submete ao servidor de aplicações.
6. O servidor de aplicações solicita ao servidor de serviços a lista dos clientes para os quais serão enviados dada a opção selecionada.
7. O servidor de serviços consulta na base de dados a lista de clientes seguindo o filtro passado.
8. O servidor de serviços retorna ao servidor de aplicações a lista dos clientes listados do banco de dados.
9. O servidor de aplicações formata o email e solicita o envio do email para o servidor de e-mails.

10. O servidor de e-mails envia o email para os clientes da casa.
11. O servidor de aplicações retorna uma mensagem ao cliente.
12. Fim do caso de uso.

21.3.2. Fluxos Alternativos

21.3.2.1. Página html não encontrada no sistema

Início: Após o item 5 do fluxo básico

Condição: O gerente do sistema não encontra nenhuma página html para enviar para os clientes.

1. Fim do caso de uso.

21.4. Pré Condições

Gerente autenticado no sistema.

Clientes com email cadastrados no sistema.

Página html com a propaganda a ser enviada preparada.

21.5. Pós Condições

Email enviado para os clientes da casa.