

UNIVERSIDADE LUTERANA DO BRASIL



TRILHA DE AUDITORIA

ESTAGIO SUPERVISIONADO

CARLOS ROBERTO DE SOUZA TAVARES

GUAÍBA, NOVEMBRO DE 2006.

GUAÍBA, NOVENBRO DE 2006.

SUMÁRIO

ÍNDICE DE FIGURAS	5
INTRODUÇÃO	6
1. DADOS DA EMPRESA.....	7
1.1 Identificação da Empresa.....	7
1.2 Identificação do Supervisor da Empresa.....	7
1.3 Caracterização da Empresa.....	7,8
2. MOTIVAÇÃO.....	9
3. OBJETIVOS	10
3.1 Objetivos Específicos	10
4. FRAMEWORK TOTALBANCO.....	11
5. METODOLOGIA A SER UTILIZADA.....	12,13
5.1 Diagramação.....	13-17
5.2 Análise de Informações.....	17
5.2.1 Situação Geral	17
5.2.2 Problemas Existentes	17
5.2.3 Necessidades.....	18
5.2.4 Relatórios	18
6. DESENVOLVIMENTO.....	19
6.1 Objetivos	19,20

6.2 Requisitos	21
6.2.1 Hardware.....	21
6.2.2 Software	21,22
6.3 Detalhamento da programação.....	22,23
7. PROJETO CONCEITUAL	24
7.1 Diagrama E-R	24
7.2 Requisitos Específicos	24
7.2.1 Diagrama de Classes.....	25-28
7.2.2 Diagrama em Seqüência.....	29-31
7.2.3 Dicionário de Dados.....	31-33
8. PROCESSO LÓGICO	34
8.1 Banco de Dados.....	34
8.2 Telas do Projeto.....	34
8.3 Interface TotalBanco	34
8.3.1 Telas Login.....	34
8.3.2 Tela Autenticação	35
8.3.3 Tela Definição da Abrangência	36
8.3.4 Tela Principal	37
8.3.5 Tela de Filtro	38
REFERÊNCIAS BIBLIOGRÁFICAS	39

ÍNDICE DE FIGURAS

Figura 1 – Ator, Usuário.....	14
Figura 2 – Casos de uso.....	15
Figura 3, Diagrama de sequencia.....	16
Figura 4, Diagrama de classes.	16
Figura 5, Associações.	17
Figura 6, Generalização.	17
Figura 7, Dependência	18
Figura 8, Diagrama E-R.....	24
Figura 9, Diagrama Sistema Segurança, pacote model.....	25
Figura 10, Diagrama Sistema TB, pacote control.	25
Figura 11, Diagrama Sistema TB, pacote view.....	26
Figura 12, Diagrama Sistema Framework, pacote event.....	27
Figura 13, Diagrama do Framework, pacote control.....	28
Figura 14, Diagrama Auditoria no Login	29
Figura 15, Diagrama Auditoria por Transação.....	30
Figura 16, Diagrama Usuário por Contexto	31
Figura 17, Tela Autenticação	35
Figura 18, Tela Seção do Módulo.....	36
Figura 19, Tela de Definição da Abrangência.....	36
Figura 20, Tela Principal do Sistema.....	37
Figura 21, Tela de Filtro do Sisitema.....	38

INTRODUÇÃO

Este trabalho tem por objetivo demonstrar o recurso da Trilha de Auditoria nas aplicações desenvolvidas pela TotalBanco. Atualmente as aplicações desenvolvidas pela Fábrica de Software não permitem a rastreabilidade das mudanças (inclusão, alteração e exclusão) realizadas nas bases de dados coligadas. As operações são executadas mas não são registradas em log. Existe entretanto, por parte do cliente, a necessidade de se registrar estas operações para futura auditoria, principalmente quando o cliente necessita para alguma auditoria externa

O presente trabalho tem por objetivo definir o funcionamento e arquitetura de implementação da funcionalidade de Trilha de Auditoria, á análise dos requisitos, as tabelas dos Banco de Dados, para que a Trilha pode ser desenvolvida e disponibilizada em todas as aplicações Web programadas em Java pela Fábrica de Software. Esta Trilha de Auditoria compreende o registro de todas as operações realizadas na(s) base(s) de dados utilizada(s) pelas aplicações, para futura verificação e auditoria.

1. DADOS DA EMPRESA

Este capítulo tem como objetivo apresentar os dados, características e tipos de serviços prestados pela empresa onde será realizado o estágio e os dados do supervisor responsável.

1.1 IDENTIFICAÇÃO DA EMPRESA

Nome da empresa: TotalBanco

Endereço: Rua dos Andradas, 1001 / Cj 901 9º andar,

CEP: 90020-015

Telefone: (51) 3227-6643

Site: <http://www.totalbanco.com.br>

1.2 IDENTIFICAÇÃO DO SUPERVISOR DA EMPRESA

Nome: Daniel Kafruni

Cargo: Gerente da Fabrica de Software JAVA

Graduação: Superior em andamento

E-Mail: kafruni@totalbanco.com.br

Telefone: (51) 9237 2595

1.3 CARACTERIZAÇÃO DA EMPRESA

Desde sua criação, o objetivo da TotalBanco é fornecer soluções integradas

ao mercado financeiro, juntamente com os serviços de personalização, implantação, treinamento e consultoria.

Fundada em 1988, focando o desenvolvimento no conhecimento das regras do negócio, baseado na parametrização extensiva das funcionalidades e da integração dos produtos, utilizando tecnologias abertas, as soluções evoluem continuamente, com base nos relacionamentos de parceria com seus clientes, nos mais diversos segmentos do mercado.

Em função das novas tecnologias e das necessidades do mercado, a TotalBanco esta sempre evoluindo as suas soluções. Atualmente, a empresa esta migrando para o ambiente *web*, obtendo a opção de uso de *softwares* livres.

A empresa conta atualmente com 70 profissionais de tecnologia e de mercado, no atendimento, desenvolvimento e manutenção de sistemas. Os produtos desenvolvidos operam em diversas plataformas, usando os diversas bancos de dados relacionais existentes no mercado.

Atuando em todo Brasil, a TotalBanco desenvolve, sob demanda, novas funcionalidades, de forma exclusiva, personalizando soluções para atender às necessidades específicas de cada cliente, oferecendo ampla flexibilidade de parametrização, permitindo adequações a outros mercados.

2. MOTIVAÇÃO

Atualmente as aplicações desenvolvidas pela Fábrica de Software Totalbanco não permitem a rastreabilidade das mudanças (inclusão, alteração e exclusão) realizadas nas bases de dados coligadas. As operações são executadas mas não são registradas em log. Existe entretanto, por parte do cliente, a necessidade de se registrar estas operações para futura auditoria, principalmente quando o cliente necessita para alguma auditoria externa.

Devido ao fato de já haver inúmeras aplicações desenvolvidas, inclusive em produção em clientes, se torna inviável a implementação da Trilha de Auditoria de maneira que seja necessária a manutenção destes sistemas. Assim, a melhor estratégia de implementação da Trilha de Auditoria é implementá-la no Framework Java (Biblioteca utilizada pela Fábrica), já que o mesmo é utilizado como infraestrutura básica de codificação em todas as aplicações.

Implementando a trilha de auditoria dentro do Framework Java, será possível compreender o registro de todas as operações realizadas na(s) base(s) de dados utilizada(s) pela aplicações, para futura verificação e auditoria. Fazendo-se assim também não será necessário codificar este recurso em todos os produtos da Fábrica, como citado anteriormente, apenas no Framework que é uma biblioteca utilizada por todos aplicativos, gerando segurança e menor codificação.

3. OBJETIVOS

O objetivo deste trabalho é implementar a funcionalidade da Trilha de Auditoria, a ser desenvolvida e disponibilizada em todas as aplicações Web programadas em Java pela Fábrica de Software, atendendo a necessidade do cliente Totalbanco, que é a necessidade de saber todas as operações realizadas na Base de Dados.

Outro objetivo importante a ser alcançado com o desenvolvimento deste trabalho é o de colocar em prática teorias e conhecimentos obtidos na Universidade.

3.1 OBJETIVOS ESPECIFICOS

- Analisar estrutura atual do Framework;
- Desenvolver trilha de auditoria na biblioteca TBFramework;
- Homologar trilha de auditoria nas aplicações Totalbanco;
- Testar trilha de auditoria;
- Disponibilizar trilha no cliente.

4. FRAMEWORK TOTALBANCO

Dê uma forma geral um framework é um conjunto de classes e interfaces que cooperam para resolver um tipo de problema de software. Este conceito se aplica a qualquer linguagem de programação, mas quanto mais amplos forem os recursos para o desenvolvimento orientado a objetos da linguagem, maiores serão as possibilidades de se desenvolver um framework com melhores aplicabilidades.

A Totalbanco possui uma tradição na implementação de sistemas utilizando frameworks. Seguindo esta tradição foi desenvolvido o TBFramework, uma biblioteca de componentes Java para auxiliar no desenvolvimento dos sistemas Web. Apesar de estar em constante evolução, este framework já possui inúmeros recursos à disposição dos programadores, tornando o trabalho não só mais ágil, como também de melhor qualidade.

5. METODOLOGIA A SER UTILIZADA

Para a migração deste Sistema e desenvolvimento no novo padrão será utilizada a UML (Linguagem de Modelagem Unificada) [UML06] , ao qual é um modelo de linguagem para modelagem de dados orientados a objetos, usada para especificar, construir, visualizar, construir e documentar um sistema de software. Com ela podemos fazer uma modelagem visual de maneira que os relacionamentos entre os componentes do sistema sejam melhor visualizados, compreendidos e documentados.

Os aspectos desta modelagem de dados podem ser definidos através dos seguintes diagramas:

- Diagrama de Caso de Uso: Um caso de uso representa uma unidade discreta da interação entre um usuário (humano ou máquina) e um sistema. Um caso de uso é uma unidade de um trabalho significativo. Cada Caso de Uso tem uma descrição a qual descreve a funcionalidade que irá ser construída no sistema proposto.
- Diagrama de classes: É uma representação da estrutura e relações das classes que servem de modelo para objetos, ou seja defini todas as classes que o sistema necessita possuir.
- Diagrama de seqüência: Representa a seqüência de processos (mais especificamente, de mensagens passadas entre objetos) num programa de

computador. Através dele pode- se descrever a maneira como o grupo de objetos colaboram em algum comportamento ao longo do tempo.

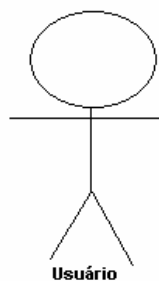
Este trabalho foi desenvolvido em 6 partes distintas que são:

1. Definição de escopo
2. Levantamento e análise de requisitos
3. Especificação do sistema
4. Desenvolvimento do sistema
5. Teste de sistema
6. Homologação de sistema

5.1 DIAGRAMAÇÃO

Atores: Representam um conjunto coerente de papéis que os usuários de casos de uso desempenham quando interagem com esses casos de uso. Tipicamente, um ator representa um papel que um ser humano, um dispositivo de hardware ou até outro sistema desempenha com o sistema.

Figura1, Ator



Caso de uso: Descreve a funcionalidade proposta para o novo sistema. Um caso de uso representa uma unidade discreta da interação entre um usuário

(humano ou máquina) e o sistema. Um Caso de Uso é uma unidade de um trabalho significativo. Por exemplo: o "login para o sistema", "selecionar módulo" e "sair sistema" são todos Casos de Uso. Cada Caso de Uso tem uma descrição o qual descreve a funcionalidade que irá ser construída no sistema proposto.

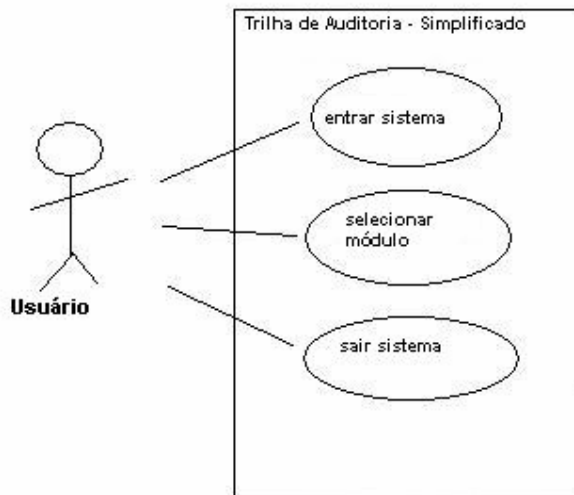


Figura 2, Casos de uso

Diagrama de Seqüência: é um tipo de diagrama usado em UML (*Unified Modeling Language*), representado a seqüência de processos (mais especificamente, de mensagens passadas entre objetos) num programa de computador.

Um diagrama de seqüência descreve a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo. Ele registra o comportamento de um único caso de uso. Ele exhibe os objetos e as mensagens passadas entre esses objetos no caso de uso. Um *design* pode ter uma grande quantidade de métodos em classes diferentes. Isso torna difícil determinar a seqüência global do comportamento. Esse diagrama é simples e lógico, a fim de tornar óbvios a seqüência e o fluxo de controle.

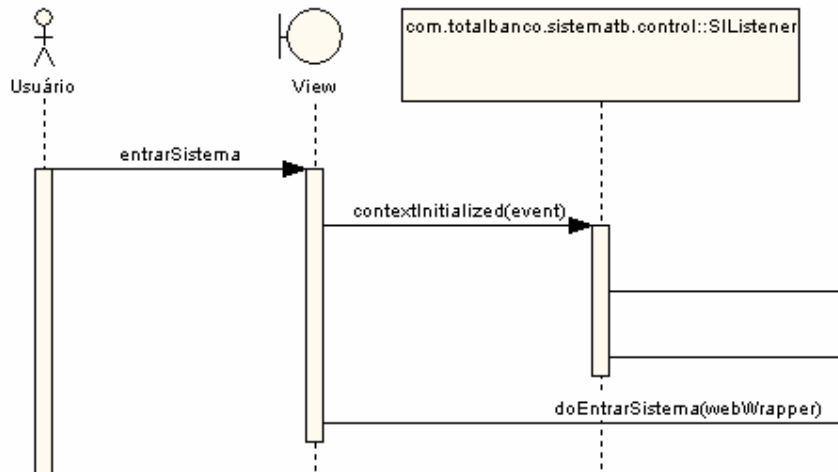


Figura 3, Diagrama de seqüência

Diagrama de Classes: é uma representação da estrutura e relações das classes que servem de modelo para objetos. É uma modelagem muito útil para o sistema, define todas as classes que o sistema necessita possuir e é a base para a construção dos diagramas de comunicação, seqüência e estados. Define os atributos e métodos de uma classe.

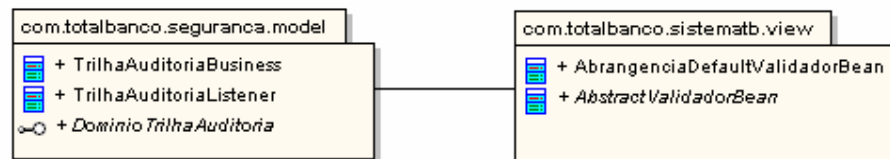


Figura 4, Diagrama de Classes

Um diagrama de classes consiste em classes e os relacionamentos entre elas. Os relacionamentos mais comuns são: Associações, Generalizações, Dependências e Refinamentos.

DIAGRAMA E-R: O Diagrama entidade relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Ele é a principal representação do Modelo de Entidades Relacionamento. Ele é usado para mostrar visualmente o relacionamento entre as tabelas de um bancos de dados. As relações são construídas através da associação de um ou mais atributos destas tabelas.

Associações: é uma ligação entre classes, portanto uma ligação entre objetos e suas respectivas classes.



Figura 5, Associações

Generalização: é o relacionamento entre um elemento mais geral e um mais específico. O mais específico pode conter apenas informações adicionais.



Figura 6, Generalização

Dependência: é um tipo de relacionamento onde um elemento é dependente e o outro independente. Uma mudança no elemento independente afeta o dependente.



Figura 7, Dependência

Refinamento: é o relacionamento entre 2 descrições da mesma coisa mas em níveis diferentes de abstração.

5.2 ANÁLISE DE INFORMAÇÕES

A análise das informações foi realizada junto a empresa TotalBanco.

5.2.1 SITUAÇÃO ATUAL

Nas aplicações desenvolvidas pela Fábrica de Software não há nada que possa fazer auditoria referente a operações em banco de dados. Caso fosse necessária alguma auditoria, isto teria que ser feito de forma manual, diretamente no Banco de Dados.

5.2.2 PROBLEMAS EXISTENTES

Atualmente as aplicações desenvolvidas pela Fábrica de Software não permitem a rastreabilidade das mudanças (inclusão, alteração e exclusão) realizadas nas bases de dados coligadas.

5.2.3 NECESSIDADES

- Registrar operações de log;
- Registrar operações de acesso a funções e transações;
- Registrar operações no banco de dados (inclusão, alteração, exclusão).

5.2.4 RELATÓRIOS

Para uma melhor visualização dos resultados, serão realizados relatórios de consultas. Estes relatórios demonstrarão de uma forma clara para o auditor, quais campos foram alterados, quais eram os valores anteriores, quais são os novos valores na tabela e quais são as chaves.

6. DESENVOLVIMENTO

Para o desenvolvimento desse projeto foram definidos os objetivos, requisitos de hardware e requisitos de software.

6.1 OBJETIVOS

A característica principal deste projeto é implementar no Sistema *Framework* TotalBanco o recurso de auditoria, que nada mais é que uma necessidade que o cliente possui para futura auditoria, onde possa ser registradas as operações realizadas em um banco de dados por determinado usuário.

O projeto terá as seguintes funções:

- a) **Registro de autenticação:** É registrado o acesso ao módulo por parte do usuário, seja uma tentativa de autenticação no sistema (com sucesso ou falha) ou sua saída do sistema, onde serão armazenados dados desta autenticação, conforme tabela abaixo:

DADO	TIPO	DOMÍNIO / MÁSCARA
Data-hora do acesso	Data/Hora	DD/MM/YYYY HH:MM:SS
Código do Usuário	Textual	
Perfil do Usuário	Textual	
Sigla do Módulo acessado	Textual	
Código da Unidade	Numérico	
IP do Computador do Usuário	Numérico Formatado	###.###.###.###
IP do Servidor do Módulo	Numérico Formatado	###.###.###.###
Domínio da rede usada	Textual	
Tipo de acesso realizado	Caractere	A - Acesso ao módulo (login); S - Saída do módulo (logout); F - Tentativa falha de acesso (erro);

- b) **Registro de acesso a função:** É registrado o acesso a determinada função de uma transação do módulo. São coletadas apenas a identificação da função, da transação e documento, uma vez que a identificação do usuário já foi coletada no nível anterior. Os dados a serem coletados são:

DADO	TIPO	DOMÍNIO / MÁSCARA
Data-hora do acesso	Data/Hora	DD/MM/YYYY HH:MM:SS
Sigla do documento acessado	Textual	
Sigla da transação executada	Textual	
Sigla da função executada	Textual	

- c) **Registrar modificações realizadas no Banco de Dados:** São coletadas e registradas as modificações realizadas na base de dados, de maneira que possibilite a verificação futura por pessoas não familiarizadas com a estrutura da base de dados. Assim, ao invés de se utilizar o nome da coluna constante no banco de dados, é utilizado sempre que possível a descrição da coluna, que é bem mais legível (e.g.: “Código do Usuário”, ao invés de “cd_usr”). Os dados a serem coletados são:

DADO	TIPO	DOMÍNIO / MÁSCARA
Data-hora da operação	Data/Hora	DD/MM/YYYY HH:MM:SS
Tipo da operação realizada	Caractere	I = Inclusão (insert); A - Alteração (update); E - Exclusão (delete);
Chave primária do registro ¹	Textual	coluna1=valor coluna2=valor ...
Conteúdo anterior do registro ²	Textual	coluna1=valor coluna2=valor ...
Conteúdo posterior do registro	Textual	coluna1=valor coluna2=valor ...

¹ A chave primária, que identifica o registro modificado, é facultativa no caso da operação de inclusão (insert), mas obrigatória nas operações de alteração (update) e exclusão (delete).

² O conteúdo anterior do registro é inexistente na operação de inclusão, e facultativo na operação de exclusão, mas obrigatório na operação de alteração.

6.2 REQUISITOS

Para o desenvolvimento desse projeto serão necessários os seguintes requisitos de hardware e software que estão descritos abaixo:

6.2.1 HARDWARE

➤ Equipamento de desenvolvimento

- Computador AMD Sempron(tm) 2600+ 1.83 GHZ
- 1 GHZ de RAM
- HD de 20 GB
- Placa de rede 10/100
- Monitor, teclado, mouse
- Impressora

6.2.2 SOFTWARE

A implementação deste sistema será realizada com o uso das seguintes tecnologias: Servidor Web Apache Tomcat, Linguagens Web XML, JAVA[SIE05], Javascript e JSP (Java Server Page). Padronizando para a nova biblioteca o sistema passa a ser multiplataforma no que diz respeito a Banco de Dados, portanto não será referenciado um banco de dados específico.

- Sistema Operacional Windows XP Versão 2002 SP 2 [STA06]
- J2SDK1.4.2_12
- Ferramenta IDE para Edição de Linguagem JAVA, Eclipse 3.2.0 [ECL05]

- Servidor Web Apache Tomcat 5.5.17 [TOM06]
- Linguagem XML[ME04], JavaScript e Linguagem JSP
- Aqua Data Studio 3.7.6 como ferramenta client para teste em banco de dados.

6.3 DETALHAMENTO DA PROGRAMAÇÃO

A implementação da Trilha de Auditoria foi realizada em sete etapas, apresentadas abaixo:

- **1) Criação das tabelas da Trilha de Auditoria** - necessárias para o registro das informações coletadas.
- **2) Mapeamento das tabelas em Java** - através da criação de classes de entidades para cada uma das tabelas manipuladas pelo Framework Java.
- **3) Registro do acesso do usuário ao módulo** - também realizada no instante da autenticação do usuário, logo após a seleção do módulo e unidade (abrangência).
- **4) Carga das Trilhas de Auditoria definidas** - realizada no instante da autenticação do usuário, para identificar as transações e funções a serem rastreadas.
- **5) Registro da função executada** - verificada a cada requisição do usuário ao sistema, e realizada através de uma classe de filtro.
- **6) Registro das operações realizadas na base de dados** - implementada diretamente nas classes do Framework Java, para cada seqüência documento / transação / função identificados anteriormente.

- **7) Programação da tela de consulta** - para pesquisa e exibição das trilhas de auditoria de maneira mais legível.

7. PROJETO CONCEITUAL

Para desenvolvimento da trilha de auditoria foram utilizados os seguintes diagramas, que ilustram o seu conceito: Diagrama E-R, Dicionário de Dados, Diagrama de Classes e Diagrama de Seqüência.

7.1 DIAGRAMA E-R

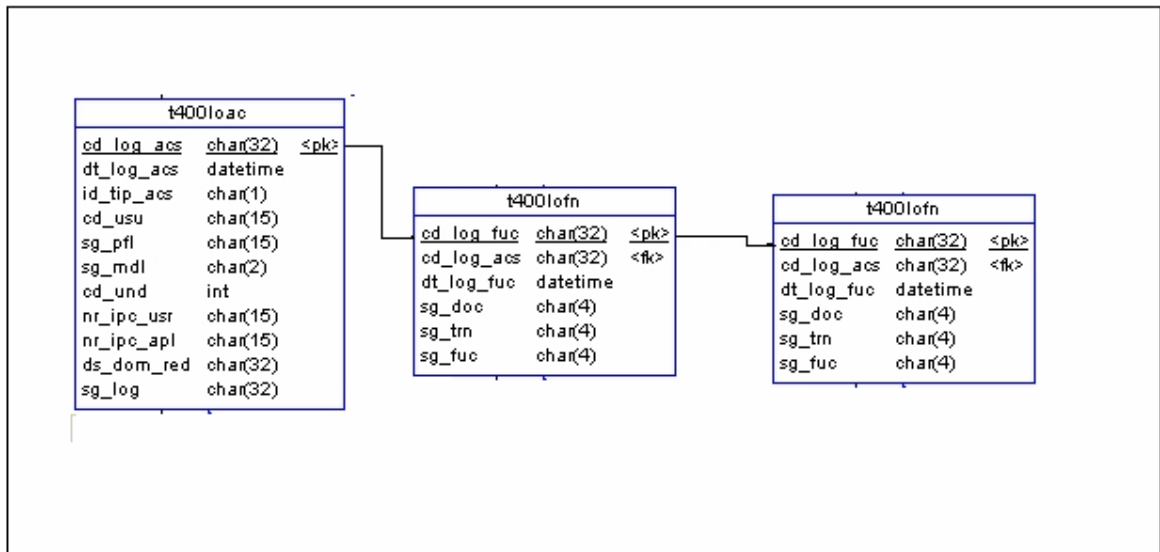


Figura 8, Diagrama E-R

7.2 REQUISITOS ESPECIFICOS

Neste tópico será mostrado os diagramas de classes, tal como os diagramas de seqüência e o Dicionário de Dados.

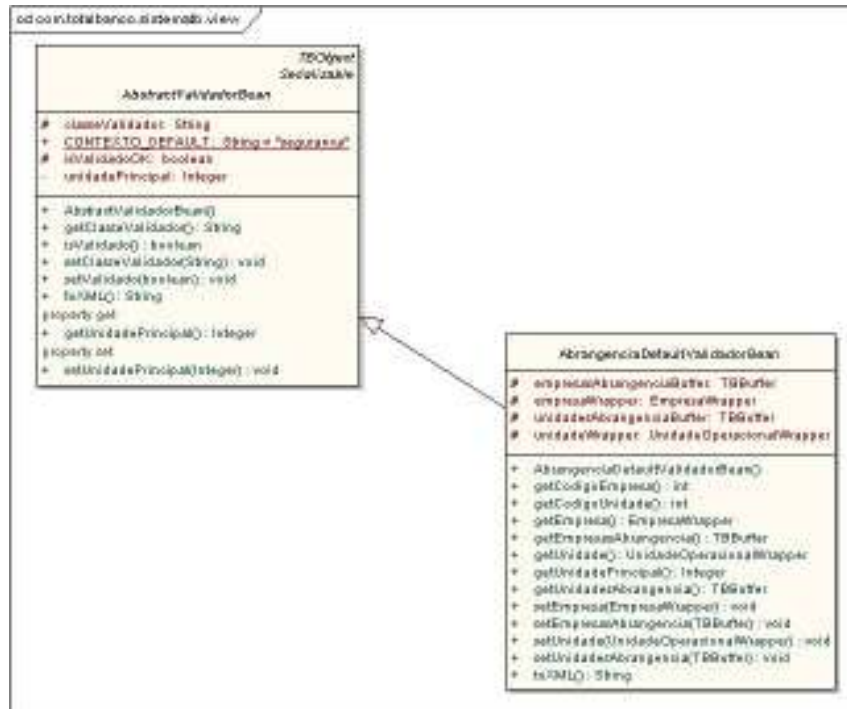


Figura 11, Diagrama Sistema TB, pacote view

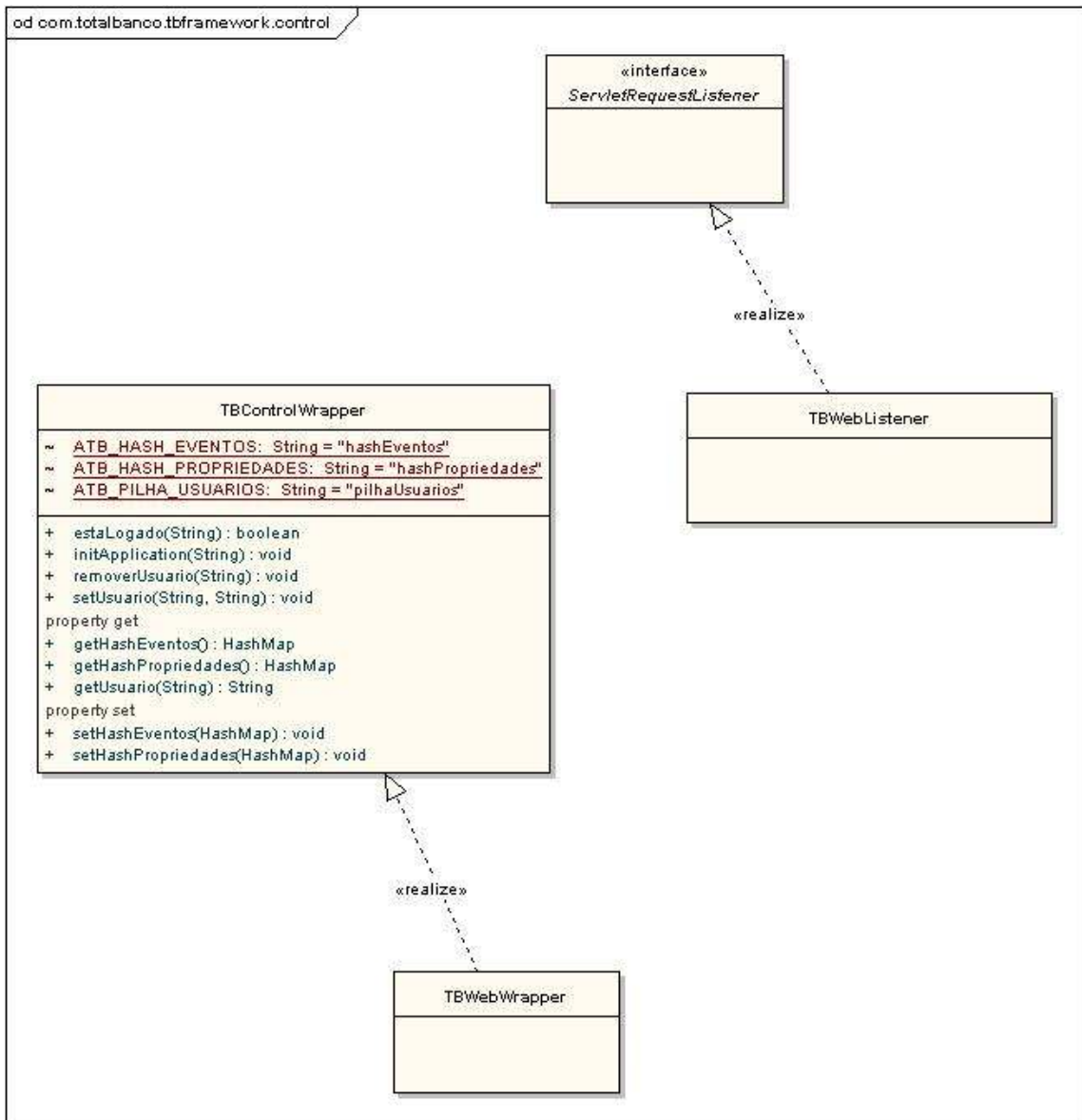


Figura 13, Diagrama do Framework, pacote control

7.2.2 DIAGRAMA DE SEQÜÊNCIA

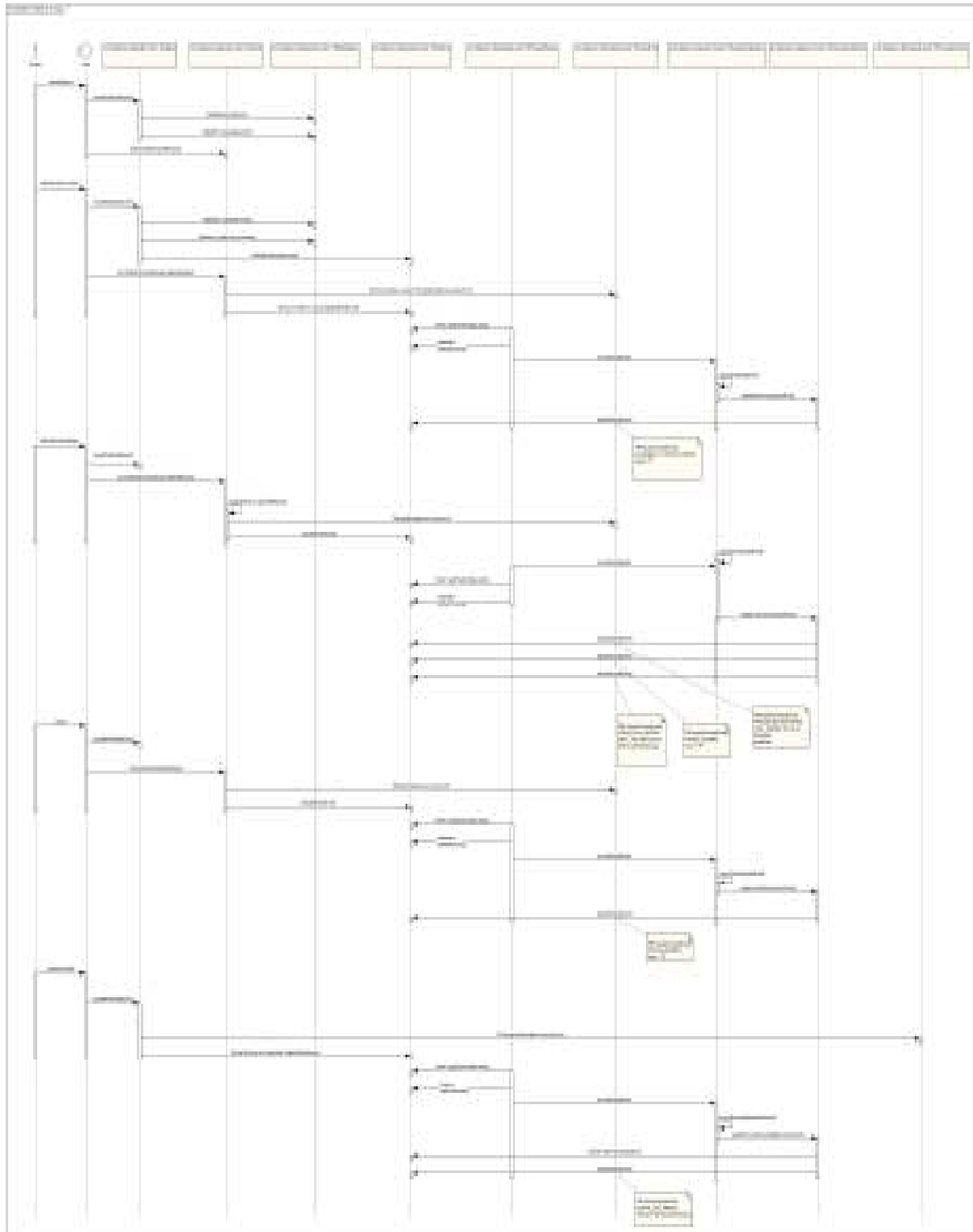


Figura 14, Diagrama Auditoria no Login

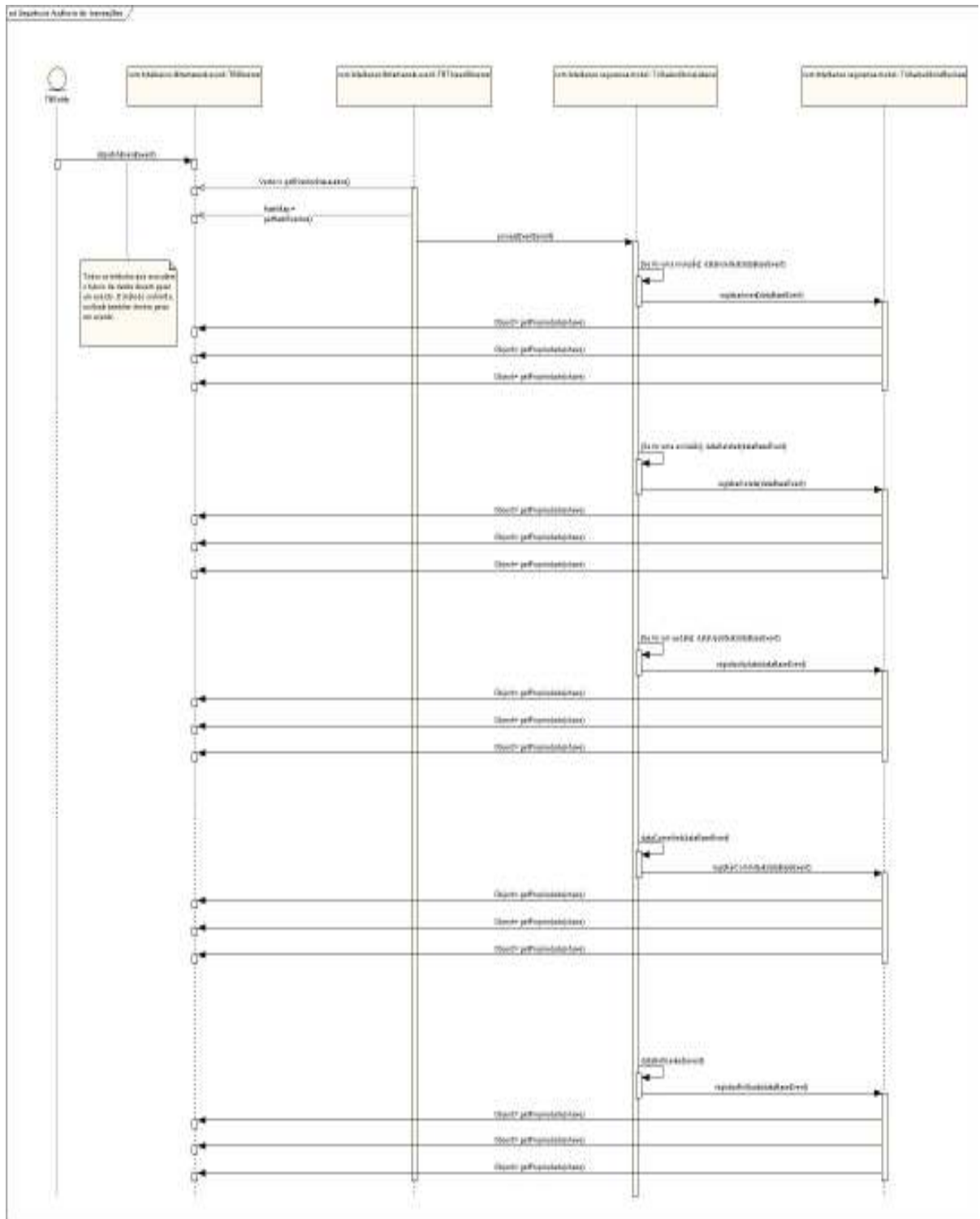


Figura 15, Diagrama auditoria por transação

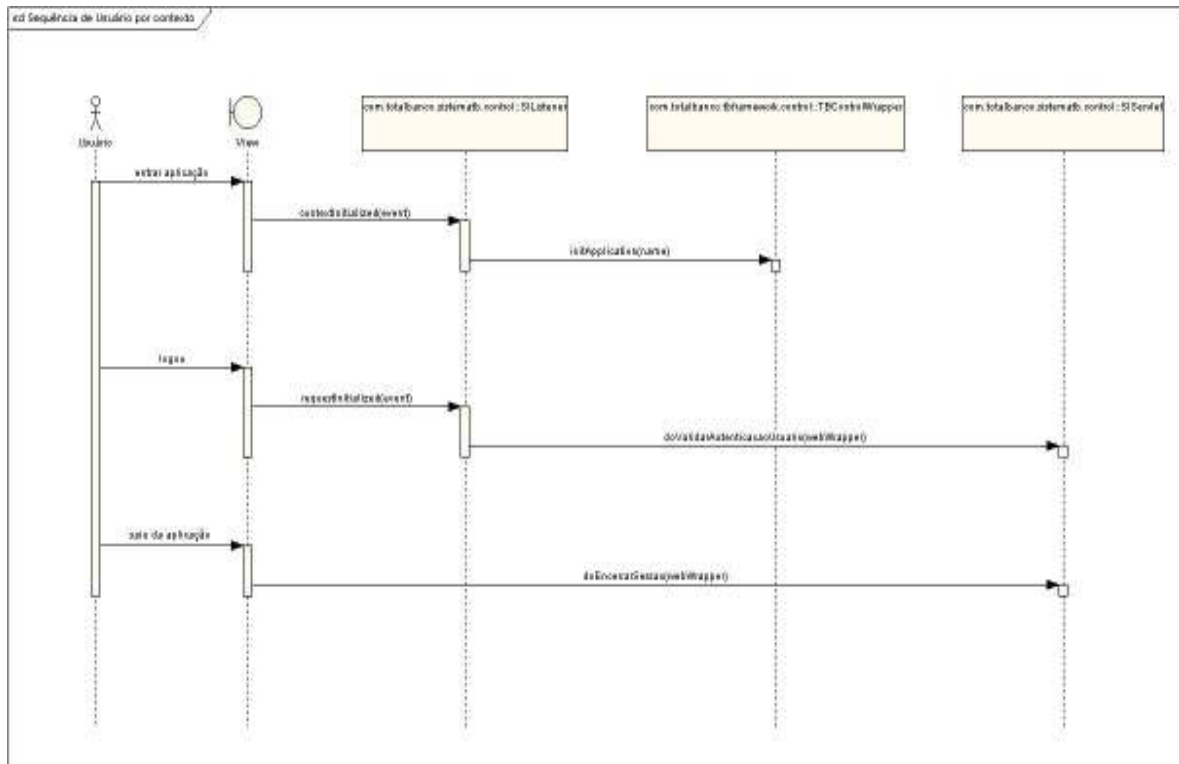


Figura 16, Diagrama usuário por contexto

7.2.3 DICIONÁRIO DE DADOS

Através da ferramenta AppModeler, foi criada as tabelas no Banco do Segurança. Abaixo esta o dicionário de dados desta tabela:

Tabela Log de Acesso (t400loac)

Coluna	Descrição	Tipo	Domínio	Obrigatória
* cd_log_acs	Código do log de acesso	char(32)		Sim
dt_log_acs	Data-hora de log do acesso	Datetime		Sim
id_tip_acs	Indicador do tipo de acesso realizado	char(1)	[A, S, F]	Sim
cd_usu	Código do usuário	char(15)		Sim
sg_pfl	Sigla do perfil do usuário no momento do acesso	char(15)		Não
sg_mdl	Sigla do módulo acessado	char(2)		Não
cd_und	Código da unidade da abrangência	Int		Não
nr_ipc_usr	Número IP do computador do usuário (cliente)	char(15)		Não

nr_ipc_apl	Número IP do computador da aplicação (servidor)	char(15)		Não
ds_dom_red	Domínio da rede usada	char(32)		Não
sg_log	Sigla da trilha definida	char(15)		Não

Tabela Log de Função (t400lofn)

Coluna	Descrição	Tipo	Domínio	Obrigatória
cd_log_fuc	Código do log de função	char(32)		Sim
dt_log_fuc	Data-hora de log da execução da função	Datetime		Sim
sg_doc	Sigla do documento acessado	char(4)		Sim
sg_trn	Sigla da transação acessada	char(4)		Sim
sg_fuc	Sigla da função executada	char(4)		Não

Tabela Log de Dados (t400loda)

Coluna	Descrição	Tipo	Domínio	Obrigatória
cd_log_dad	Código do log de dados	char(32)		Sim
dt_log_dad	Data-hora de log da operação sobre os dados	datetime		Sim
id_tip_ope	Indicador do tipo da operação realizada	char(1)	[I, A, E]	Sim
ds_chv_reg	Chave primária do registro	varchar(1024)		Não
ds_ant_reg	Conteúdo anterior do registro	varchar(1024)		Não
ds_pos_reg	Conteúdo posterior do registro	varchar(1024)		Sim

Tabela de Auditoria (t400audi)

Coluna	Descrição	Tipo	Domínio	Obrigatória
sg_log	Sigla da trilha definida	char(15)		Sim
de_log	Descrição da trilha	char(40)		Não
nm_arq	Nome do arquivo de log	char(8)		Não
dt_ini	Data do início da validade	datetime		Não
dt_fim	Data do final da validade	datetime		Não

Tabela de Transações Auditadas (t400lotr)

Coluna	Descrição	Tipo	Domínio	Obrigatória
sg_log	Sigla da trilha definida	char(15)		Sim
sg_mdl	Sigla do módulo auditado	char(2)		Não
sg_doc	Sigla do documento auditado	char(4)		Não
sg_trn	Sigla da transação auditada	char(4)		Não

Tabela de Unidades Auditadas (t400loag)

Coluna	Descrição	Tipo	Domínio	Obrigatória
sg_log	Sigla da trilha definida	char(15)		Sim
cd_und	Código da unidade auditada	int		Não

Tabela de Perfil Auditado (t400lope)

Coluna	Descrição	Tipo	Domínio	Obrigatória
sg_log	Sigla da trilha definida	char(15)		Sim
sg_pfl	Sigla do perfil auditado	char(15)		Não
cd_usu	Código do usuário auditado	char(15)		Não

** Campos definidos como chave primária (primary key) da tabela.*

8. PROJETO LÓGICO

8.1 BANCO DE DADOS

O projeto da Trilha de Auditoria não está restrito há nenhum banco de dados, pois a empresa trabalha com SQL ANSI, estando os comandos aptos a rodarem em qualquer Base de Dados.

8.2 TELAS DO PROJETO

Por motivos da Trilha de Auditoria, não ser uma aplicação em si, mas um recurso as aplicações, este projeto não dispõe de telas próprias, pois faz uso das telas dos Sistemas TotalBanco para realizar as suas operações.

8.3 INTERFACES TOTALBANCO

A TotalBanco tem como padrão trabalhar com os mesmos tipos de telas em suas aplicações, adotando é claro às suas interfaces os estilos, imagens, transações referentes a cada cliente. Abaixo as principais telas do Sistema TotalBanco:

8.3.1 TELA AUTENTICAÇÃO

A tela de autenticação do usuário é a primeira tela a ser mostrada ao abrir um sistema no padrão TotalBanco, conforme figura 17. Caso der alguma falha ao acessar com usuário, o sistema de Trilha de auditoria, gravará um registro,

marcando a ocorrência de falha. Caso o usuário seja válido é verificado quais módulos terá acesso, para mostrar estes na próxima tela.

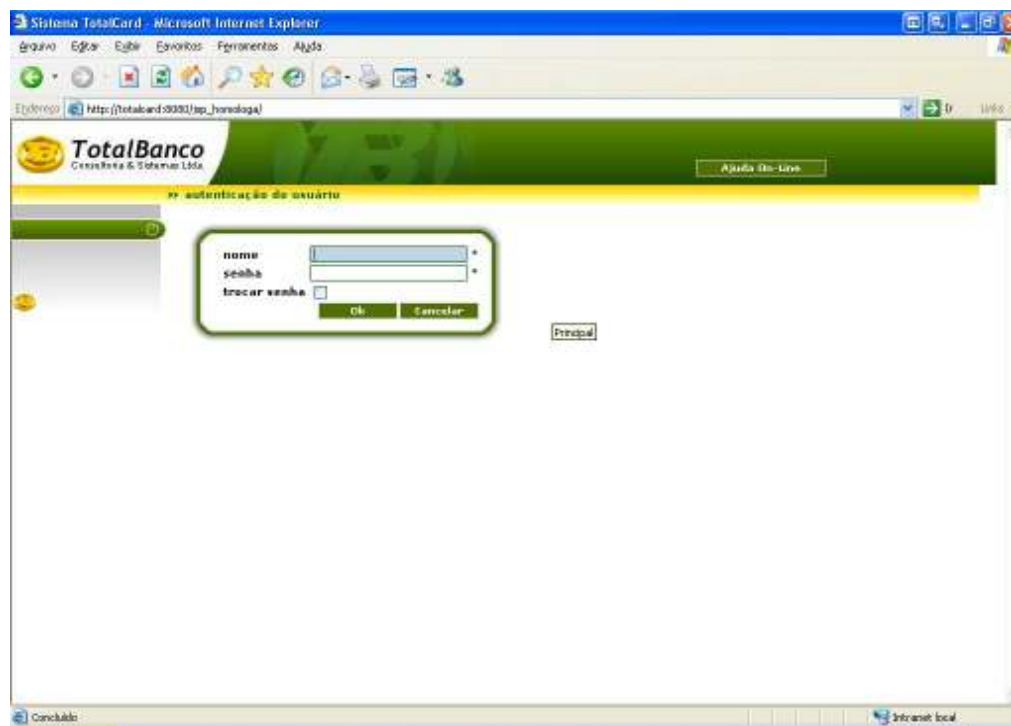


Figura 17, tela autenticação

8.3.2 TELA SELEÇÃO DO MÓDULO

Na tela de seleção do módulo conform figura 18, é demonstrado os módulos ao qual o cliente possui acesso.

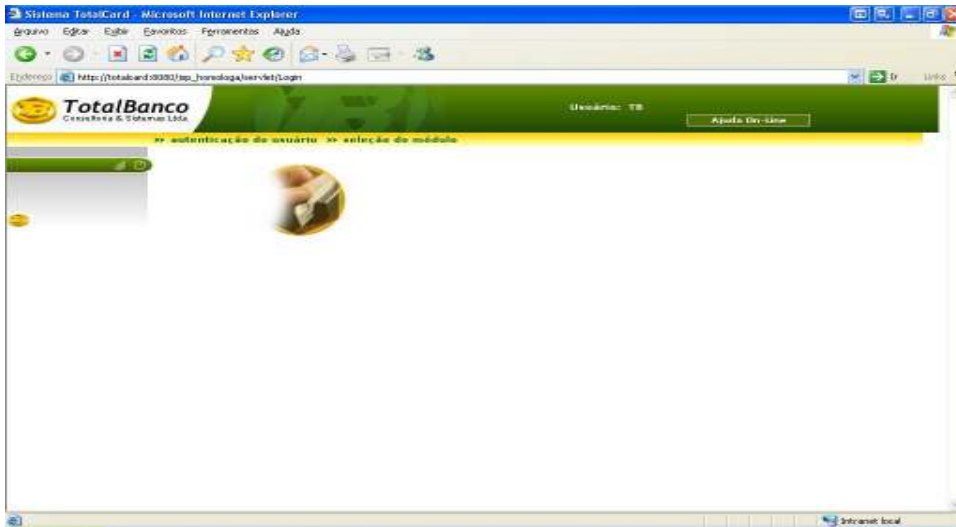


Figura 18, tela seleção do módulo

8.3.3 TELA DEFINIÇÃO DA ABRANGÊNCIA

Na figura 19 esta sendo demonstrado a tela de seleção de empresa/ponto de atendimento, ou seja a abrangência ao qual o usuário estará definindo o seu acesso. No momento em que o usuário escolhe a abrangência, caso ele seja um usuário auditável, será cadastrada na tabela t400loac, o acesso com sucesso realizado.

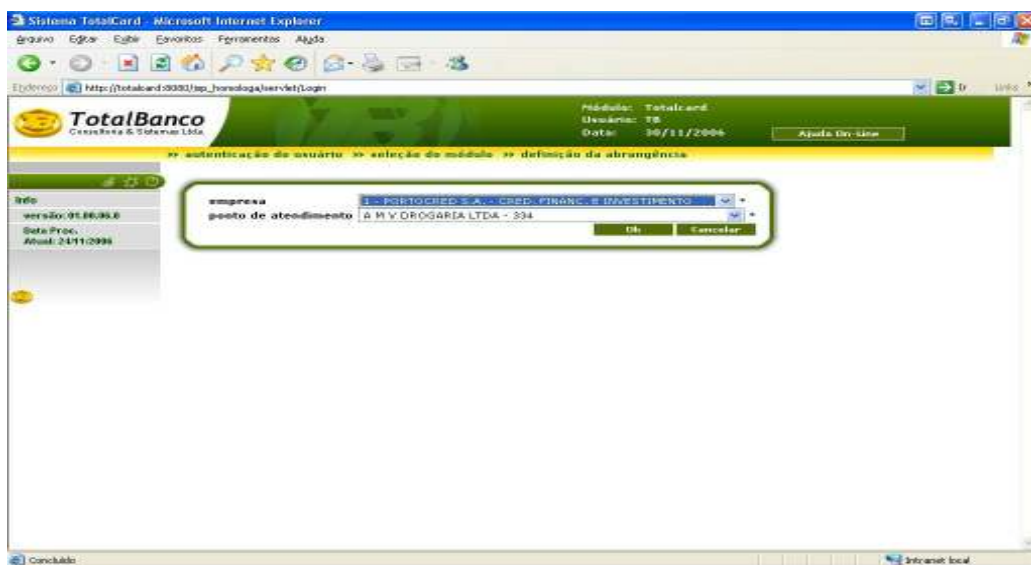


Figura 19, tela de definição da Abrangência

8.3.4 TELA PRINCIPAL

A figura 20 demonstra a tela principal dos Sistemas TotalBanco, onde o usuário pode escolher as transações disponíveis para este módulo nas opções do lado esquerdo, tal como na barra verde superior esquerda, trocar de usuário, módulo, abrangência ou sair do sistema.

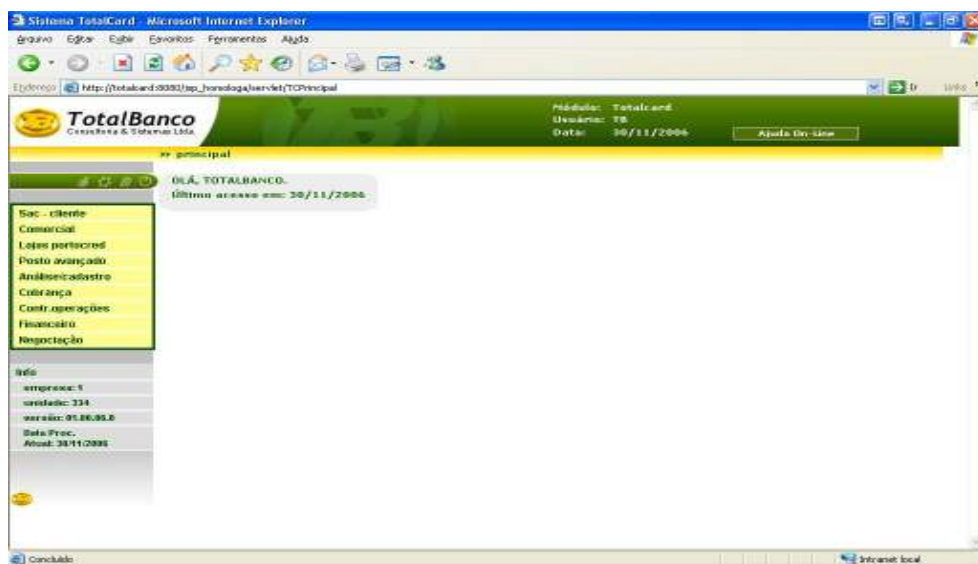


Figura 20, tela principal do Sistema

8.3.5 TELA DE FILTRO

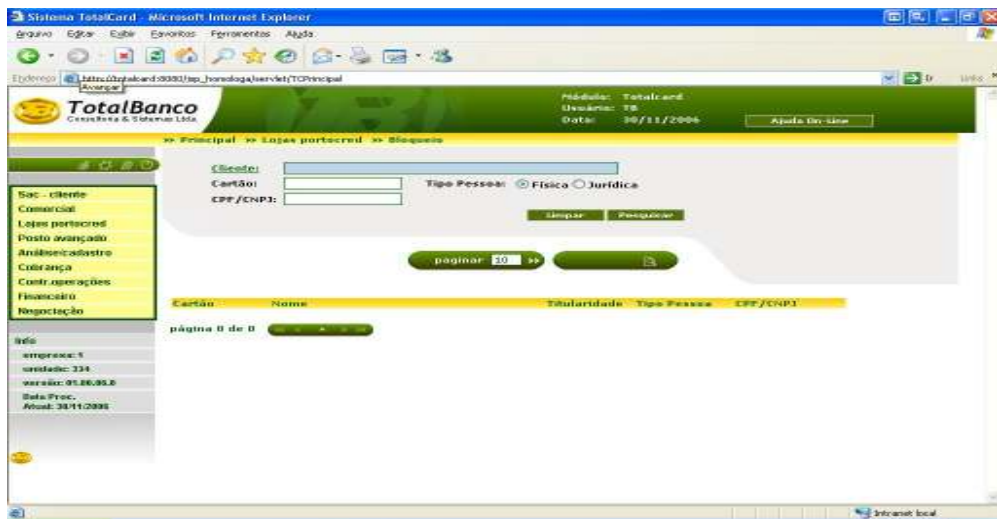


Figura 21, tela de filtro do sistema

BIBLIOGRAFIA

1. [UML06] UML. Disponível por: <http://pt.wikipedia.org/wiki/UML/> em 27/11/2006.
2. [STA06] STANEK, William. Microsoft Windows XP Professional. Ed. Bookman: 2006.
3. [TOM06] TOMCAT. Disponível por: jakarta.apache.org/tomcat/ em 25/08/2006.
4. [ECL06] ECLIPSE 3.0.1. Disponível por: www.eclipse.org/ em 25/08/2006.
5. [ME04] MENDES, Antônio. Programando com XML. Ed. Campus: 2004.
6. [SIE05] SIERRA, Kathy. JAVA 2 – Certificação Sun para programadores e desenvolvedores