

INSTITUTO FEDERAL DO PARANÁ - CAMPUS PARANAVAÍ
CURSO SUPERIOR EM TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

LUIS GUILHERME DE JESUS GOMES

**SISTEMA DE CONTROLE DE PRESTAÇÃO DE SERVIÇOS:
ESCRITÓRIO CONTÁBIL**

TRABALHO DE CONCLUSÃO DE CURSO

PARANAVAÍ
2018

LUIS GUILHERME DE JESUS GOMES

**SISTEMA DE CONTROLE DE PRESTAÇÃO DE SERVIÇOS:
ESCRITÓRIO CONTÁBIL**

Trabalho de Conclusão de Curso apresentado ao Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas da Instituto Federal do Paraná - Campus Paranavaí, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Esp. Frank Willian Cardoso de Oliveira
IFPR - Campus Paranavaí

PARANAVAÍ
2018

AGRADECIMENTOS

Agradeço aos meus pais pelo incentivo durante todos esses anos.

À Organização Apolo de Contabilidade, pela experiência e cooperação.

Ao meu orientador Frank Willian, pela ajuda, motivação e compreensão.

Aos colegas de classe e amigos pelo apoio.

Aos colaboradores do Instituto Federal do Paraná - Campus de Paranavaí pela transmissão de conhecimento.

Pouco conhecimento faz com que as pessoas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o Céu, enquanto que as cheias as baixam para a terra, sua mãe. (Leonardo da Vinci)

RESUMO

Gomes, Luis. SISTEMA DE CONTROLE DE PRESTAÇÃO DE SERVIÇOS: Escritório Contábil. 2018. 28 f. Trabalho de Conclusão de Curso – Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Federal do Paraná - Campus Paranavaí. Paranavaí, 2018.

Os serviços de um escritório contábil, devem seguir uma rotina fixa e se intercalarem entre os setores de maneira que os dados sejam distribuídos e melhor utilizados para geração de informação, para auxiliar esse serviço um sistema de informação é decisivo para a qualidade do resultado das operações. Neste trabalho será desenvolvido um sistema web em Java para gerenciar alguns dos serviços cruciais para um escritório em específico, sendo eles: o controle de entrega e devolução de documentos e o controle de recebimento de honorários contábeis.

Palavras-chave: Serviços. Contábil. Sistema.

LISTA DE FIGURAS

Figura 1 – Diagrama de Caso de Uso	10
Figura 2 – Diagrama de Classe	11
Figura 3 – Diagrama de Sequencia - Caso de uso: Manter Honorário	12
Figura 4 – Diagrama de Entidade Relacionamento	21
Figura 5 – Tabela de protocolos	26

LISTA DE QUADROS

Quadro 1 – Gerar Protocolo de Entrega de Documento.	13
Quadro 2 – Devolver Documento.	14
Quadro 3 – Manter Despesas.	15
Quadro 4 – Manter Honorários.	16
Quadro 5 – Receber Honorários.	17
Quadro 6 – Classe de despesas.	17
Quadro 7 – Classe de despesas adicionais.	18
Quadro 8 – Classe de cliente.	18
Quadro 9 – Classe de município.	18
Quadro 10 – Classe de estado.	18
Quadro 11 – Classe de honorário.	19
Quadro 12 – Classe de usuário.	19
Quadro 13 – Classe de protocolo.	19
Quadro 14 – Classe de Documento.	19
Quadro 15 – Classe de documentos de protocolos.	20
Quadro 16 – Exemplo de Anotações Hibernate	20
Quadro 17 – Tabela de clientes	22
Quadro 18 – Tabela de usuários	22
Quadro 19 – Tabela de despesas	23
Quadro 20 – Tabela de despesas adicionais	23
Quadro 21 – Tabela de documentos	24
Quadro 22 – Tabela de documentos dos protocolos	24
Quadro 23 – Tabela de estados	25
Quadro 24 – Tabela de municípios	25
Quadro 25 – Tabela de honorários	26

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 Objetivos Gerais	1
1.2 Objetivos Específicos	1
1.3 Justificativa	2
1.4 Organização	2
2 – METODOLOGIA	4
3 – FERRAMENTAS	5
3.1 Eclipse	5
3.2 GIT	5
3.3 Jaspersoft	5
4 – PLATAFORMA, LINGUAGEM e FRAMEWORKS	6
4.1 Linguagem de Programação Java e Java EE	6
4.2 Hibernate	6
4.3 JavaServer Faces	7
4.4 PrimeFaces	7
5 – MODELAGEM DO SOFTWARE	8
5.1 Requisitos de Software	8
5.1.1 Requisitos Funcionais	8
5.1.2 Requisitos Não Funcionais	9
5.2 Diagrama de Caso de Uso	10
5.3 Diagrama de Classe	11
5.4 Diagrama de Sequência	12
5.5 Descrição dos Casos de Uso	13
5.6 Descrição do Diagrama de Classes	17
5.7 Solução de Mapeamento Objeto-Relacional	20
5.8 Diagrama de Entidade Relacionamento	21
5.9 Dicionário de Dados	22
6 – CONCLUSÃO	27
6.1 TRABALHOS FUTUROS	27
Referências	28

1 INTRODUÇÃO

No escritório sobre o qual será desenvolvido este trabalho existe atualmente dois serviços que são feitos manualmente, o que acaba atrasando e dificultando a realização desses serviços. São eles: o gerenciamento de entrega e recebimento de documentos e o gerenciamento dos honorários contábeis.

Nesse escritório a entrega e devolução dos documentos como impostos, holerites, contratos, boletos e etc, é controlada pelo setor de entregas do escritório através de um modulo de software que gera protocolos de entrega de documentos, que devem ser assinados pelos clientes que receberem os documentos. Os documentos que foram entregues devem voltar para o escritório para que sejam arquivados e contabilizados, o que melhora a organização dos documentos dos clientes e torna mais fácil o acesso dos contadores a esses documentos. Essa devolução é controlada através de planilhas impressas manualmente pelos funcionários do escritório.

Já o controle de honorários é feito através de fichas em papel, sendo uma ficha pra cada cliente, ao lançar o honorário é preciso adicionar as despesas complementares que foram geradas para cada cliente e somar seu valor ao valor padrão de cada honorário, a partir desses dados são feitos os recibos que serão utilizados para cobrar os clientes do escritório, quando pago o valor deve ser anotado na ficha e subtraído do valor total devido, se o valor pago for maior que a divida do cliente na ficha deverá constar um crédito ou se for menor deverá constar um débito.

Para o controle desses serviços, está atualmente disponível dois módulos de um software que gere alguns dos serviços do escritório, porém a utilização desse sistema especificamente nos módulos desses serviços é muito complexa e pouco prática, por isso o sistema acaba não sendo utilizado ou sendo utilizado parcialmente. então os controles acabam sendo feitos manualmente.

1.1 Objetivos Gerais

Este trabalho tem como objetivo desenvolver de um sistema para um escritório de contabilidade para facilitar a gestão do recebimento financeiro de serviços prestados e o controle de documentos que são repassados aos seus clientes.

1.2 Objetivos Específicos

- Gestão de despesas possíveis dos clientes, como abertura e fechamento de empresa, contratos, alterações de contrato, recálculo de tributos e livros;
- Gestão de honorários contábeis por competência, emissão de recibos para cobrança, lançamento de despesas a serem cobradas, descontos, créditos, reajustes, juros e multa;

- Gestão de recebimento e devolução de documentos dos clientes, como notas fiscais, impostos, contratos, documentos, despesas de água, luz, telefone e aluguel;
- Emissão de recibo de honorários;
- Emissão de relatórios de controle dos honorários: pagos, em aberto ou vencidos;
- Emissão de relatório de movimentação de documentos: entregue, devolvido, arquivado;
- Guia de Ajuda, informando como usar cada funcionalidade do sistema.
- Controle de Acesso via usuário e senha, contando com usuários com níveis de acesso diferentes: (Empregado e Gerente);

1.3 Justificativa

O controle dos honorários, despesas e documentos dentro do escritório contábil segundo [LONDERO \(2005\)](#) envolve um ciclo mensal e contínuo, e conta com vários funcionários para a geração e utilização de informações, a junção dessas informações deve ser controlada de forma com que todos que puderem acessá-la possam visualizar e tomar ações a partir dos dados coletados, com o auxílio de um sistema de informação, esses dados se tornam mais claros e confiáveis, além de aumentar a velocidade com que essas informações são compartilhadas e gerenciadas.

Atualmente no escritório grande parte dos controles nos setores de entrega e financeiro são feitos via papel e caneta, e cada funcionário tem seu próprio controle individual. Ao marcar uma despesa, ou serviço nesse documento ele deve ser passado ao gerente no fim do período para que ele possa contabilizar as despesas e lançar nos honorários. Dessa forma os processos da prestação de serviço do escritório se torna lento, árduo e passível de falhas, podendo ocasionar informações errôneas por perda de dados, dificuldade de entendimento dos dados e atrasos.

Ao fazer esse controle em um sistema, além de aumentar a velocidade com que essas informações são gravadas e transmitidas, diminuirá o risco de perda de informações e também facilitará o entendimento delas pelo gerente, visto que a qualidade da informação sem perda de dados é maior.

1.4 Organização

Este trabalho será organizado da seguinte forma: a princípio foi apresentada a introdução que descreveu diretamente a situação inicial do Escritório, quais serviços serão informatizados e como funcionavam antes do trabalho, em seguida, foram apresentados os objetivos gerais e específicos desse trabalho seguido da justificativa.

Após a seção de organização será apresentada a metodologia, que irá informar sobre os passos tomados para a progressão desse trabalho, adiante estão descritas as ferramentas que serão utilizadas de forma a auxiliar a produção do software e controle de arquivos, a frente está a seção de Plataforma, Linguagem e Frameworks, que dará uma breve descrição da linguagem de programação, e dos projetos de código que serão utilizados para facilitar e agilizar

o desenvolvimento do sistema. a seguir está a modelagem do Software que apresenta todos os documentos que representam o sistema seguido dos resultados e uma breve informação sobre o que ainda deve ser feito.

2 METODOLOGIA

Para o desenvolvimento desse trabalho serão executados os seguintes procedimentos:

- Pesquisa Bibliográfica: outros trabalhos técnicos parecidos, artigos e livros sobre programação e engenharia de software.
- Entrevistas com funcionários do escritório alvo.
- Modelagem do Software: elicitación de requisitos, criação de diagramas de classe, casos de uso, sequencia, entidade - relacionamento e prototipação do sistema interagindo diretamente com funcionários e gerentes do escritório e com o orientador para a validação dos dados.
- Documentação: criação e atualização do documento técnico enquanto o trabalho for progredindo.
- Desenvolvimento: criação, desenvolvimento e configuração do sistema de informação web que implementará as funcionalidades descritas na documentação.

3 FERRAMENTAS

As ferramentas facilitam o trabalho de um desenvolvedor, pois visam o maior aproveitamento do tempo e segurança do processo, utilizamos essas tecnologias com mais frequência com o passar do tempo devido a essas vantagens, temos como exemplos o Eclipse, o Git e o Jaspersoft, nos tópicos a seguir será detalhada a forma como essas ferramentas atuam em um projeto de software, entre outros detalhes.

3.1 Eclipse

Eclipse é um famoso ambiente integrado de desenvolvimento (IDE) de software livre para java, também bom para C/C++ ([ECLIPSE-FUNDATION, 2018](#)), pode-se combinar facilmente o suporte a idiomas e outros recursos através de pacotes padrão. Neste trabalho será utilizado para digitar, compilar, executar e testar os códigos, além de permitir a manipulação do servidor de forma automatizada, desta forma agilizando e facilitando o trabalho.

3.2 GIT

O GIT é um sistema de controle de versões de arquivos gratuito e de código aberto ([GIT, 2018](#)) e um dos seus diferenciais é seu sistema de ramificações onde pode-se criar vários ramos independentes entre si, e depois decidir se deseja compartilhá-los, mesclá-los ao seu ramo principal ou excluí-los. Esse sistema ainda traz como vantagem a segurança de que cada desenvolvedor terá um backup do servidor, guardado localmente, e que a cada mudança poderá ser salvo, substituindo a versão principal.

3.3 Jaspersoft

O jaspersoft é uma ferramenta para auxiliar a criação de modelos de relatórios, consultas de relatório, e componentes visuais com mais de 50 tipos de gráficos, mapas, tabelas e muito mais ([TIBCO, 2018](#)).

4 PLATAFORMA, LINGUAGEM e FRAMEWORKS

A escolha da plataforma de desenvolvimento de um software é indispensável, e assim a plataforma e o framework devem funcionar satisfatoriamente com a linguagem, e a realidade atual facilita cada vez mais os processos que envolvem o desenvolvimento dos projetos de software.

4.1 Linguagem de Programação Java e Java EE

DEITEL e DEITEL (2010) afirmam que Java é uma linguagem de programação fortemente orientada a objetos, com a popularização de computadores pessoais, modificaram a vida de todos na época, desde pessoas a grandes corporações. Com a linguagem de programação Java, pode-se trabalhar com várias situações, por ser uma linguagem versátil a mesma possui três principais versões, mas a que será utilizada para desenvolver o sistema será o Java EE (Java Enterprise Edition), java para aplicações de grande porte, voltada para corporações, e desenvolvimento web

O Java EE, que será utilizado no desenvolvimento do Sicon tem como maior diferença as aplicações web, nesse projeto o Java será executado em um servidor, que poderá suportar várias conexões, o Tomcat em conjunto com outros frameworks.

As aplicações java por ser executadas na JVM (Java Virtual Machine) segundo o que SIERRA e BATES (2010) apresentam torna-se uma linguagem de programação multiplataforma, sendo limitada apenas a possibilidade de instalação da JVM, dessa forma independentemente do sistema, todas as aplicações Java funcionam, e ainda pode ser portada por outro sistema sem a necessidade de alteração do código.

4.2 Hibernate

Devido a dificuldade de trabalhar com aplicações orientadas a objetos e banco de dados relacionais e a demanda por desenvolvimento mais rápido no mercado, segundo SILVA (2016), surgiram várias soluções para facilitar a integração entre elas e os banco de dados relacionais e resolver este problema bem como poupar o programador de muito trabalho manual, para Primo a principal e mais popular solução foi o Hibernate que é uma ferramenta ou framework de mapeamento de objeto relacional ORM object relational mapping que ajuda a persistir objetos java em um banco de dados relacional,ele faz acesso ao banco e gera todos os códigos sql necessários.

4.3 JavaServer Faces

Java Server Faces também conhecido como JSF, Segundo ([ANDRADE, 2013](#)), foi criado com a intenção de facilitar a produtividade no desenvolvimento de interfaces visuais, é um framework baseado em componentes de interface com o usuário; e estruturado na arquitetura do padrão de projeto MVC (Model View Controller). Os componentes são a base para a construção da interface com o usuário, são demonstrados por meios de tags em um JSP, e após “transformados” em HTML. O JSF é um Framework dirigido a eventos, como por exemplo, a chamada de um método; este framework é considerado um padrão de mercado na atualidade e está presente na especificação JEE 5.0.

4.4 PrimeFaces

O PrimeFaces segundo [SCHIECK \(2015\)](#) é considerado uma biblioteca de componentes para aplicações com JSF, ou seja, a ferramenta possui uma ampla quantidade de componentes, desde botões, campos de texto em geral, agregando também o ajax, que é um plug-in integrado, que facilita o desenvolvimento, não havendo a necessidade do programador configurar os componentes para o ajax, o próprio PrimeFaces já efetua a configuração deste componente. A grande vantagem na utilização da plataforma é que possibilita uma maior opção de componentes estilizados e possui um padrão de componentes dos mesmos.

5 MODELAGEM DO SOFTWARE

Neste capítulo será apresentada a documentação de modelagem que auxilia a produção e conferência do software que será desenvolvido, essa documentação abrange os requisitos de software, o diagrama de casos de uso, o diagrama de classes e o diagrama de entidade e relacionamento.

5.1 Requisitos de Software

Requisitos, de acordo com MEDEIROS (2013), são além de funções, objetivos, propriedades, também restrições que o sistema deve possuir para satisfazer normas, padrões ou especificações de acordo com o usuário final. Ainda de acordo com MEDEIROS (2013), de forma geral, os requisitos são uma condição necessária para satisfazer um Objetivo.

5.1.1 Requisitos Funcionais

Os requisitos funcionais refletem as funções e informações do sistema. (MEDEIROS, 2013).

Para este projeto os requisitos funcionais são:

- I - O sistema possibilita manter o cadastro de empresas clientes;
- II - O sistema possibilita manter o cadastro de despesas;
- III - O sistema possibilita manter o cadastro de honorários contábeis;
- IV - O sistema possibilita manter o cadastro de documentos;
- V - O sistema possibilita manter o cadastro de usuários;
- VI - O sistema possibilita manter o cadastro de protocolos de entrega;
- VII - O sistema deve gerar Relatório de despesas;
- VIII - O sistema deve gerar Recibos de honorários contábeis;
- IX - O sistema deve gerar Relatório de documentos;
- X - O sistema deve gerar Relatório de clientes;
- XI - O sistema deve gerar Relatório em forma Protocolos de Entrega;
- XII - O sistema deve gerar Relatório de Documentos entregues, devolvidos e não devolvidos;
- XIII - O sistema deve gerar protocolos de entrega, para documentos que serão entregues aos clientes;
- XIV - O sistema possibilita o controle da devolução dos documentos que foram entregues aos clientes;
- XV - O sistema possibilita o controle do pagamento dos honorários com status (pago, em aberto, vencido);

- XVI - O sistema deve controlar o valor dos honorários contábeis por competência, através dos créditos e débitos e das despesas de mesma referência;
- XVII - O valor do honorário que não for pago deverá ser somado ao valor do honorário subsequente;
- XVIII - Cada um dos documentos inseridos no protocolo poderão ser devolvidos individualmente;

5.1.2 Requisitos Não Funcionais

Já os requisitos não funcionais referem-se aos critérios que qualificam os requisitos funcionais(MEDEIROS, 2013).

Requisitos não funcionais desse projeto:

- I - O sistema deverá ser acessado pelo navegador;
- II - O sistema deve ser desenvolvido com base nas tecnologias Java;
- III - O tempo máximo para a geração de relatórios é de 10 segundos;
- IV - O tempo máximo para a transição de telas é de 10 segundos;
- V - O sistema deve conter um guia de ajuda ao usuário.
- VI - Honorários só poderão ter o status de pagamento alterado pelo gerente;

5.2 Diagrama de Caso de Uso

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Ou seja, ele descreve as principais funções do sistema e como essas funções interagem com o usuário. (RIBEIRO, 2012).

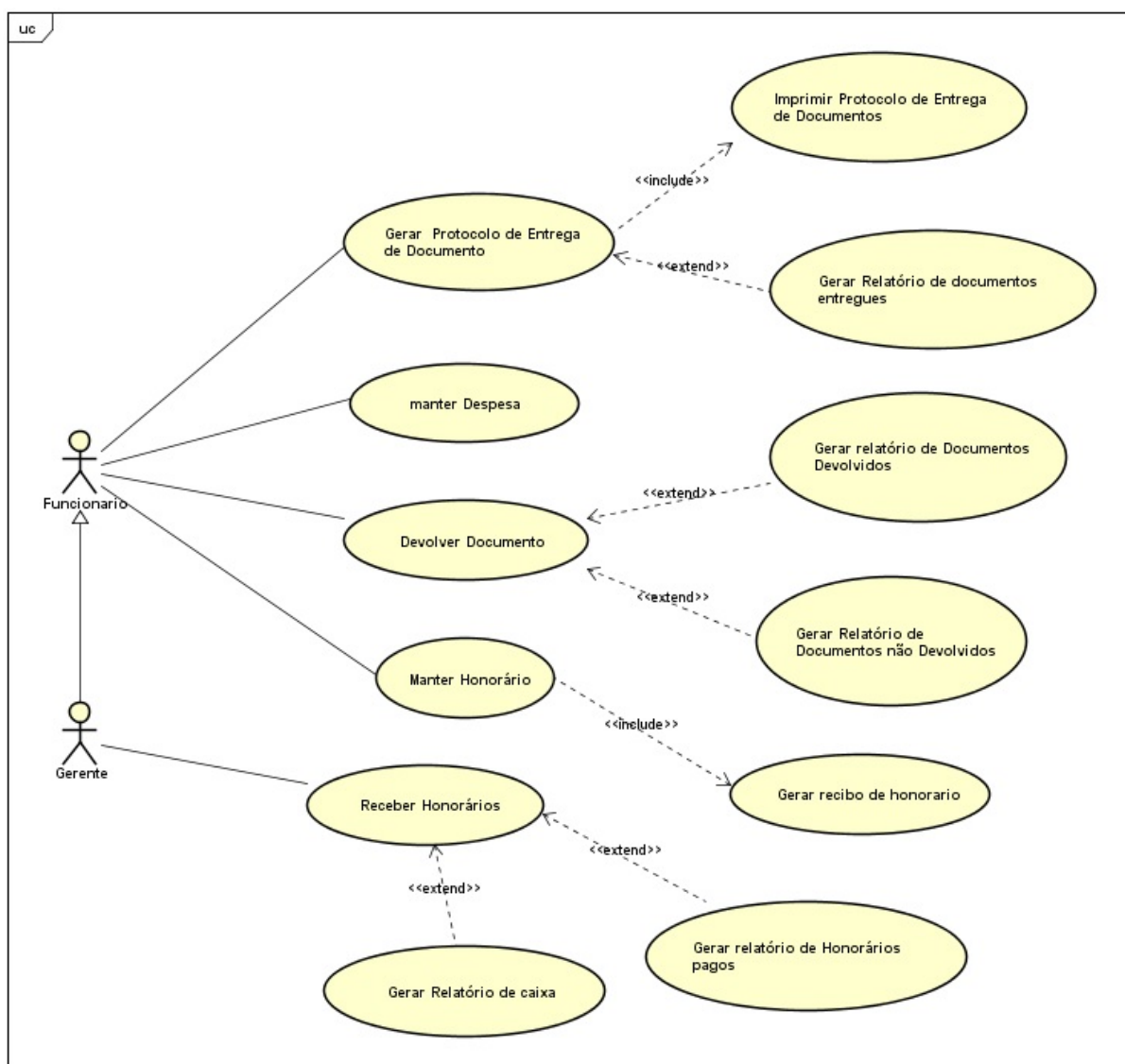
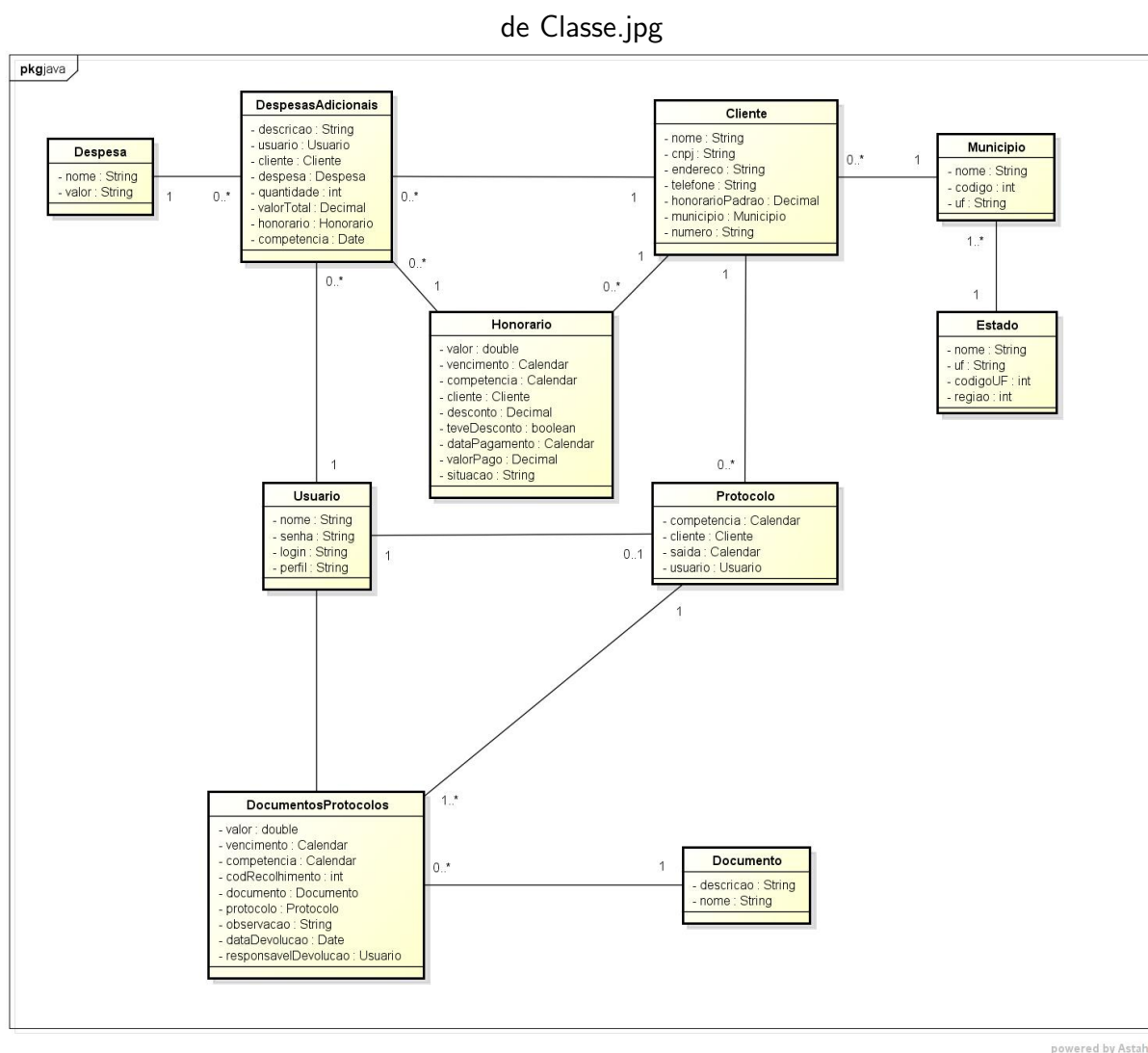


Figura 1 – Diagrama de Caso de Uso

5.3 Diagrama de Classe

O Diagrama de classe é uma representação da estrutura e relações das classes que servem de modelo para objetos. Simplificando a classe é um conjunto de objetos com as mesmas características, assim podemos identificar objetos e agrupá-los, de forma a classificá-los (TYBEL, 2016).



5.4 Diagrama de Sequência

O diagrama de sequencia segundo [MELO \(2009\)](#) é um diagrama de interação, que detalha a interação entre objetos do sistema dinamicamente, passando uma noção sobre as mensagens passadas entre os objetos durante o tempo de execução.

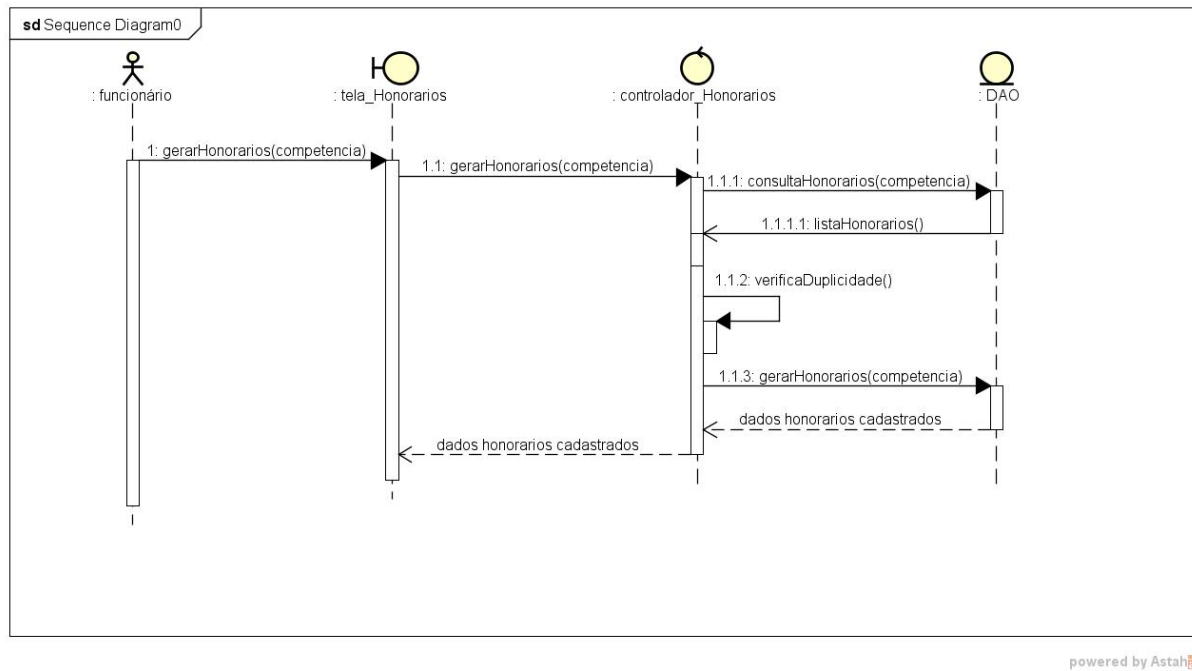


Figura 3 – Diagrama de Sequencia - Caso de uso: Manter Honorário

5.5 Descrição dos Casos de Uso

A descrição dos casos explica detalhadamente como funciona cada caso de uso, através dos passos seguidos pelos atores, em conjunto com o sistema, aponta o fluxo principal de uso e os fluxos alternativos e de exceção, além de apontar os requisitos que aquele caso de uso se refere.

Quadro 1 – Gerar Protocolo de Entrega de Documento.

Objetivo	Criar o Protocolo de Entrega de Documentos
Requisitos	VI, XII e XIII
Atores	Funcionário
Fluxo Principal	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção protocolo 2. O sistema exibe a tela de gerenciamento de protocolos contendo: <ul style="list-style-type: none"> - Botão de Novo Protocolo - Botão de Buscar Protocolo - Lista dos últimos protocolos 3. O funcionário seleciona o botão Novo Protocolo 4. O sistema exibe a tela de novo protocolo 5. O funcionário preenche os dados e seleciona salvar 6. O sistema exibe o Relatório 7. O funcionário seleciona Imprimir
Fluxo Alternativo 01 - Gerar Relatório de Documentos Entregues	<ol style="list-style-type: none"> 1. após o passo 2 do fluxo principal o funcionário clica no botão "Relatório" 2. O sistema exibe a pagina para definir os parâmetros do relatório 3. O funcionário preenche os dados e seleciona o botão "Gerar" 4. Volta ao passo 06 do fluxo principal
Fluxo Alternativo 02 - Imprimir protocolo já criado	<ol style="list-style-type: none"> 1. após o passo 2 do fluxo principal o funcionário clica no botão "Buscar" 2. O sistema exibe a pagina para definir os parâmetros do relatório 3. O funcionário preenche os dados e seleciona o botão "Gerar" 4. Volta ao passo 06 do fluxo principal

Quadro 2 – Devolver Documento.

Objetivo	Gravar no sistema a informação de que o documento foi devolvido
Requisitos	VI, XIV e XVIII
Atores	Funcionário
Pré-condições	Estar autenticado no sistema
Fluxo Principal	<ol style="list-style-type: none">1. O funcionário seleciona a opção "Documentos"2. O sistema exibe a tela de gerenciamento de documentos3. O funcionário seleciona o botão "Devolução"4. O sistema exibe a tela de devolução de documentos5. O funcionário preenche os dados e seleciona o botão "Devolver"6. O sistema exibe uma mensagem confirmando realização da operação
Fluxo Alternativo 01 - Gerar relatório de documentos devolvidos	<ol style="list-style-type: none">1. Após o passo 2 do fluxo principal o funcionário seleciona "Relatório"2. O sistema exibe a tela de relatório de documentos3. O funcionário preenche os dados e clica em "Gerar"4. O sistema exibe o relatório
Fluxo Alternativo 02 - Gerar relatório de documentos não devolvidos	<ol style="list-style-type: none">1. Após o passo 2 do fluxo principal o funcionário seleciona "Relatório"2. O sistema exibe a tela de relatório de documentos3. O funcionário preenche os dados e clica em "Gerar"4. O sistema exibe o relatório

Quadro 3 – Manter Despesas.

Objetivo	Manter Cadastro de Despesas
Requisitos	II e VII
Atores	Funcionário
Pré-condições	Estar autenticado no sistema cliente estar cadastrado no sistema
Fluxo Principal	1. O funcionário seleciona a opção Despesas 2. O sistema exibe a tela de gerenciamento de Despesas 3. O funcionário seleciona o botão Nova Despesa 4. O sistema exibe a tela de Nova Despesa 5. O funcionário seleciona e preenche os dados e clica em Salvar 6. O sistema exibe uma mensagem confirmando o sucesso da operação
Fluxo Alternativo 01 - Alterar Despesa	1. Após o passo 2 do fluxo principal o funcionário preenche os dados de busca e clica no botão "Alterar Despesa" 5. volta ao passo 5 do fluxo principal
Fluxo Alternativo 02 - Excluir Despesa	1. Após o passo 2 do fluxo principal o funcionário preenche os dados de busca e clica no botão "Excluir Despesa" 5. volta ao passo 5 do fluxo principal

Quadro 4 – Manter Honorários.

Objetivo	Manter o cadastro dos Honorários Contábeis dos Clientes
Requisitos	III, VIII, XVI e XVII
Atores	Funcionário
Pré-condições	Estar autenticado no sistema cliente estar cadastrado no sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção Honorário 2. O sistema exibe a tela de gerenciamento de Honorário 3. O funcionário seleciona o botão Gerar Honorários 4. O sistema exibe uma mensagem de sucesso e atualiza a lista com os últimos Honorários gerados
Fluxo Alternativo 01 Alterar Honorário	<ol style="list-style-type: none"> 1. Após o passo 2 do fluxo principal o funcionário clica em "Buscar Honorário" 2. O sistema exibe a tela de Busca de Honorários 3. O funcionário preenche os dados de busca e clica no botão "Buscar" 4. O sistema Exibe a lista correspondente a busca do funcionário na tela 5. O funcionário seleciona o Honorário na lista e clica em "Alterar Honorário" 6. O sistema exibe a tela de alteração de honorário 7. O funcionário preenche os dados e clica no botão "Salvar" 8. volta ao passo 4 do fluxo principal
Fluxo Alternativo 02 - Gerar Recibo de honorário	<ol style="list-style-type: none"> 1. Após o passo 2 do fluxo principal o funcionário preenche os dados de busca e clica no botão "Buscar" 2. O sistema Exibe a lista correspondente a busca do funcionário na tela 3. O funcionário clica em "Gerar Recibo" 4. O sistema exibe a tela de relatórios preenchida com os recibos gerados 5. O funcionário clica em "Imprimir"

Quadro 5 – Receber Honorários.

Objetivo	Marcar Honorários como recebidos no sistema
Requisitos	XV e XVII
Atores	Funcionário
Pré-condições	Estar autenticado no sistema Honorário estar cadastrado no sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O funcionário seleciona a opção "Honorário" 2. O sistema exibe a tela de gerenciamento de Honorário 3. O funcionário seleciona o botão "Receber Honorários" 4. O sistema exibe a tela de Busca de Honorários 5. O funcionário seleciona o Honorário e clica em "Receber" 6. O sistema exibe a tela de Recebimento de honorário 7. O funcionário preenche os dados e clica no botão "Receber" 8. O sistema Exibe uma mensagem de confirmação 9. O funcionário clica em 'Sim' 10. O sistema Exibe uma mensagem de sucesso
Fluxo Alternativo 01 - Confirmação Negada	<ol style="list-style-type: none"> 1. Após o passo 8 do fluxo principal o funcionário clica em 'Não' 2. O sistema exibe uma mensagem de cancelamento da operação
Fluxo Alternativo 01 - Gerar Relatório de Caixa	<ol style="list-style-type: none"> 1. Após o passo 2 do fluxo principal o funcionário clica em "Relatório" 2. O sistema exibe a tela de Geração de Relatórios 3. O funcionário preenche os dados e clica em "Gerar"

5.6 Descrição do Diagrama de Classes

A Descrição do Diagrama de Classes visa explicar para que servem cada um dos itens das classes, facilitando o entendimento sobre as classes e seus relacionamentos. A seguir está a descrição de cada uma das classes desse sistema.

Despesas	Armazena as despesas modelo no sistema
Descrição dos atributos da classe	
nome	guarda o nome das despesas
valor	guarda o valor das despesas

Quadro 6 – Classe de despesas.

DespesasAdicionais	Armazena as despesas adicionais de cada cliente
Descrição dos atributos da classe	
descrição	guarda alguma informação útil sobre a despesa
usuário	guarda que usuário gerou a despesa no sistema
cliente	guarda a que cliente a despesa pertence
despesa	guarda o padrão da despesa
quantidade	guarda a quantidade de despesas do mesmo padrão
valorTotal	guarda orma a soma das quantidades do valor da despesa
honorario	guarda a que honorário a despesa pertence
competencia	guarda a referência da despesa

Quadro 7 – Classe de despesas adicionais.

Cliente	Armazena os dados dos clientes do escritório
Descrição dos atributos da classe	
nome	guarda o nome da empresa cliente
cnpj	guarda o CNPJ do cliente
endereço	guarda o nome da rua do cliente
telefone	guarda o telefone do cliente
honorarioPadrao	guarda o valor padrão do honorário daquele cliente
municipio	guarda o nome da cidade do cliente
numero	guarda o número do endereço do cliente

Quadro 8 – Classe de cliente.

Municipio	Armazena as cidades do Brasil
Descrição dos atributos da classe	
nome	guarda o nome da cidade
codigo	guarda o código do IBGE da cidade
uf	guarda a sigla do estado

Quadro 9 – Classe de município.

Estado	Armazena os estados do Brasil
Descrição dos atributos da classe	
uf	guarda a sigla do estado
codigoUF	guarda o código UF do Estado
regiao	guarda o código da região do estado

Quadro 10 – Classe de estado.

Honorario	Armazena os honorários de cada cliente
Descrição dos atributos da classe	
valor	guarda o valor total do honorário
vencimento	guarda a data de vencimento do honorário
competência	guarda a referência do honorário
cliente	informa o cliente ao qual o honorário pertence
desconto	guarda o valor de desconto do honorário (se houver)
dataPagamento	guarda a data de pagamento do honorário
valorPago	guarda o valor pago pelo cliente
situacao	informa a situação do honorário (Em aberto, pago, atrasado)

Quadro 11 – Classe de honorário.

Usuario	Armazena os dados dos usuarios do sistema
Descrição dos atributos da classe	
nome	guarda o nome do usuário
senha	guarda a senha do usuário
login	guarda a informação de entrada do usuário no sistema
perfil	informa o nível de acesso do usuário ao sistema

Quadro 12 – Classe de usuário.

Protocolo	obj
Descrição dos atributos da classe	
competência	guarda a referência do protocolo
cliente	informa a que cliente o protocolo pertence
saída	guarda a data de criação do protocolo
usuário	informa que usuário criou o protocolo

Quadro 13 – Classe de protocolo.

Documento	Armazena os documentos modelo que o escritório entrega
Descrição dos atributos da classe	
tipo	guarda o tipo de documento
nome	guarda o nome do documento

Quadro 14 – Classe de Documento.

Documentos	Protocolos
Armazena os documentos que foram inseridos no protocolo	
Descrição dos atributos da classe	
valor	guarda o valor do documento (se houver)
vencimento	guarda a data de vencimento do documento
competencia	guarda a referência do documento
codigoDeRecolhimento	guarda o código de pagamento do documento é usado para identificar o tipo do documento
documento	informa qual o documento modelo para esse documento
protocolo	informa a que protocolo pertence esse documento
observacao	guarda alguma informação sobre esse documento
dataDevolucao	guarda a data de devolução do documento (para os documentos devolvidos)
usuarioResponsavelDevolucao	informa que usuário lançou a devolução no sistema

Quadro 15 – Classe de documentos de protocolos.

5.7 Solução de Mapeamento Objeto-Relacional

Os frameworks de mapeamento objeto relacional oferecem a solução para a diferença entre paradigmas que é um problema comum ao desenvolver software orientado a objetos. Esses frameworks facilitam a integração entre os paradigmas, automatizando grande parte do trabalho de criação de tabelas e estruturas de bancos de dados e ainda na persistência e manutenção dos dados.

O framework utilizado nesse projeto é o Hibernate com Java Persistence API (JPA), essas ferramentas unidas geram o código SQL em tempo de execução para o banco de dados, sendo necessário apenas a utilização de anotações para guiar o framework sobre o que o programador deseja que seja feito.

Exemplo de anotações e suas funções:

@Entity	Indica que esta classe será uma tabela do banco de dados
@Id	Indica que o atributo id é a chave primária da tabela, ou seja, é o atributo que identifica o objeto nessa tabela.
@GeneratedValue	Indica que o atributo (geralmente o Id) seguirá algum padrão de implementação, como por exemplo o auto-increment, que a cada novo registro somará por padrão 1 ao id atual, iniciando por 0.
@Named	indica qual o nome que a tabela do banco de dados terá.

Quadro 16 – Exemplo de Anotações Hibernate

Para se comunicar com o banco de dados algumas configurações devem ser definidas no projeto, através do arquivo *persistence.xml* é possível definir o banco de dados a ser utilizado, o seu local, usuário, senha, e etc.

5.9 Dicionário de Dados

O dicionário de dados informa detalhadamente como estão definidos cada atributo das tabelas de banco de dados utilizadas no projeto, além do tipo de dados, se pode ou não ser nulo e as ligações entre as tabelas (foreign keys).

cliente

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (<i>Primária</i>)	bigint(20)	Não				
honorario_padrao	decimal(10,2)	Não				
nome	varchar(255)	Sim	<i>NULL</i>			
status	bit(1)	Não				
municipio_id	int(11)	Não		municipio -> id		

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo	Comentário
PRIMARY	BTREE	Sim	Não	id	5	A	Não	
municipio_id	BTREE	Não	Não	municipio_id	2	A	Não	

Quadro 17 – Tabela de clientes

usuario

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (<i>Primária</i>)	bigint(20)	Não				
login	varchar(255)	Sim	<i>NULL</i>			
nome	varchar(255)	Sim	<i>NULL</i>			
senha	varchar(255)	Sim	<i>NULL</i>			
status	bit(1)	Não				
perfil	varchar(255)	Sim	<i>NULL</i>			

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	5	A	Não

Quadro 18 – Tabela de usuários

despesa

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (Primária)	bigint(20)	Não				
nome	varchar(255)	Sim	NULL			
status	bit(1)	Não				
valor	decimal(10,2)	Não				

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	6	A	Não

Quadro 19 – Tabela de despesas

despesasadicionais

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (Primária)	bigint(20)	Não				
competencia	date	Sim	NULL			
descricao	varchar(255)	Sim	NULL			
quantidade	int(11)	Não				
valorTotal	decimal(10,2)	Sim	NULL			
cliente_id	bigint(20)	Sim	NULL	cliente -> id		
honorario_id	bigint(20)	Sim	NULL	honorario -> id		
usuario_id	bigint(20)	Sim	NULL	usuario -> id		
despesa_id	bigint(20)	Sim	NULL	despesa -> id		

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	9	A	Não
FK_96e41hru4xxdmviuyak2gie6u	BTREE	Não	Não	cliente_id	9	A	Sim
FK_29tlo3mxex6y78qeeve5ohjk	BTREE	Não	Não	honorario_id	9	A	Sim
FK_5jtxwk36n6gskpe3d50aa0jdv	BTREE	Não	Não	usuario_id	2	A	Sim
FK_j87yjjvw3ih8am84v36sl6l59y	BTREE	Não	Não	despesa_id	9	A	Sim

Quadro 20 – Tabela de despesas adicionais

documento

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (Primária)	bigint(20)	Não				
descricao	varchar(255)	Sim	NULL			
nome	varchar(255)	Sim	NULL			
status	bit(1)	Não				

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	2	A	Não

Quadro 21 – Tabela de documentos

documentosprotocolos

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (Primária)	bigint(20)	Não				
competencia	date	Sim	NULL			
observacao	varchar(255)	Sim	NULL			
valor	double	Não				
vencimento	date	Sim	NULL			
documento_id	bigint(20)	Sim	NULL	documento -> id		
protocolo_id	bigint(20)	Sim	NULL	protocolo -> id		
codRecolhimento	int(11)	Não				
dataDevolucao	date	Sim	NULL			
responsavelDevolucao_id	bigint(20)	Sim	NULL	usuario -> id		
devolvido	bit(1)	Não				

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	18	A	Não
FK_iew014b6jutqxn3ws118718pn	BTREE	Não	Não	documento_id	4	A	Sim
FK_fbtbarvw14pifu8179qd6ipd8	BTREE	Não	Não	protocolo_id	18	A	Sim
FK_tnjdmvf2ryoaokue48ebxrds	BTREE	Não	Não	responsavelDevolucao_id	2	A	Sim

Quadro 22 – Tabela de documentos dos protocolos

estado

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
Id (<i>Primária</i>)	int(11)	Não				
CodigoUf	int(11)	Não				
Nome	varchar(50)	Não				
Uf	char(2)	Não				
Regiao	int(11)	Não				

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	Id	27	A	Não

Quadro 23 – Tabela de estados

municipio

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (<i>Primária</i>)	int(11)	Não				
Codigo	int(11)	Não				
Nome	varchar(255)	Não				
estado_id	int(11)	Não		estado -> Id		

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	5570	A	Não
estado_fk	BTREE	Não	Não	estado_id	54	A	Não

Quadro 24 – Tabela de municípios

honorario

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (Primária)	bigint(20)	Não				
competencia	datetime	Sim	NULL			
desconto	decimal(10,2)	Sim	NULL			
status	bit(1)	Não				
valor	decimal(10,2)	Não				
vencimento	datetime	Sim	NULL			
cliente_id	bigint(20)	Sim	NULL	cliente -> id		
dataPagamento	datetime	Sim	NULL			
valorPago	decimal(10,2)	Sim	NULL			
pago	bit(1)	Não				
teveDesconto	bit(1)	Não				

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	49	A	Não
FK_75naydjkd019q2kgcp4hkxjf	BTREE	Não	Não	cliente_id	12	A	Sim

Quadro 25 – Tabela de honorários

protocolo

Coluna	Tipo	Nulo	Predefinido	Ligações para	Comentários	MIME
id (Primária)	bigint(20)	Não				
competencia	date	Sim	NULL			
saida	date	Sim	NULL			
status	bit(1)	Não				
cliente_id	bigint(20)	Sim	NULL	cliente -> id		
usuario_id	bigint(20)	Sim	NULL	usuario -> id		

Índices

Nome da chave	Tipo	Único	Pacote	Coluna	Quantidade	Agrupamento (Collation)	Nulo
PRIMARY	BTREE	Sim	Não	id	19	A	Não
FK_fgug70d4nnninge9114ohhop	BTREE	Não	Não	cliente_id	9	A	Sim
FK_87qr1hh1w7kq35tn7gr2bxrpd	BTREE	Não	Não	usuario_id	2	A	Sim

Figura 5 – Tabela de protocolos

6 CONCLUSÃO

Com a utilização desse sistema é esperado uma melhora nos serviços, de forma que o tempo necessário para cumprir as obrigações seja reduzido, também é esperado uma diminuição da perda de informações, já que devido a velocidade e facilidade de inserção de informação, os dados poderão ser gravados logo após a sua geração, e que o sistema permita uma maior visibilidade das informações facilitando o entendimento das informações e melhorando o trabalho de todos os setores envolvidos.

6.1 TRABALHOS FUTUROS

Para a conclusão desse trabalho, é necessário a criação do restante dos relatórios, a integração do endereço dos clientes, bem como uma nova entrevista e apresentação do sistema aos funcionários do escritório para validação e correção de funcionalidades.

Referências

- ANDRADE, F. T. **Java EE 7 com JSF, PrimeFaces e CDI**. São Paulo: AlgaWorks, 2013. Citado na página 7.
- DEITEL, P.; DEITEL, H. **Java como programar**. 8. ed. Nova Jersey: Pearson Prentice Hall, 2010. Citado na página 6.
- ECLIPSE-FUNDATION. **Desktops IDEs**. 2018. Disponível em: <<https://www.eclipse.org/ide/>>. Acesso em: 23 de maio de 2018. Citado na página 5.
- GIT. **About: Branching and Merging**. 2018. Disponível em: <<https://git-scm.com/about/>>. Acesso em: 23 de maio de 2018. Citado na página 5.
- LONDERO, B. A. A contabilidade na administração de empresas. **Simpósio de iniciação Científica dos Cursos de Ciências Contábeis de Santa Maria - Universidade Federal de Santa Maria**, Santa Maria, 2005. Citado na página 2.
- MEDEIROS. **Introdução a Requisitos de Software**. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>>. Acesso em: 21 de maio de 2018. Citado 2 vezes nas páginas 8 e 9.
- MELO, A. C. A contabilidade na administração de empresas. **Revista Engenharia de Software 15**, São Paulo, 2009. Citado na página 12.
- RIBEIRO. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. 2012. Disponível em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em: 22 de maio de 2018. Citado na página 10.
- SCHIECK. **Introdução ao PrimeFaces**. 2015. Disponível em: <<https://www.devmedia.com.br/introducao-ao-primefaces/33139>>. Acesso em: 20 de maio de 2018. Citado na página 7.
- SIERRA, K.; BATES, B. **Use a Cabeça Java**. 2. ed. Rio de Janeiro: Alta Books, 2010. Citado na página 6.
- SILVA, I. P. **Desenvolvendo com Hibernate**. 2016. Disponível em: <<http://www.devmedia.com.br/artigo-java-magazine-73-desenvolvendo-com-hibernate/14756>>. Acesso em: 20 de maio de 2018. Citado na página 6.
- TIBCO. **Jaspersoft Studio**. 2018. Disponível em: <<https://community.jaspersoft.com/project/jaspersoft-studio>>. Acesso em: 24 de maio de 2018. Citado na página 5.
- TYBEL. **Orientações básicas na Elaboração de um Diagrama de Classes**. 2016. Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 22 de maio de 2018. Citado na página 11.