Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

Paulo Henrique Silva dos Santos

Self Pay

Software gerenciador de folha de pagamento

Assis
2014

Paulo Henrique Silva dos Santos

Self Pay

Sistema Gerenciador de folha de pagamento

Trabalho de Conclusão de Curso apresentado à Fundação Educacional do Município de Assis, como requisição do Curso de Graduação de Tecnologia em Processamento de Dados.

Orientado: Paulo Henrique Silva dos Santos
Orientador: Dr. Alex Sandro Romeo de Souza Poletto

Assis
2014

Agradecimentos

Primeiramente a Deus por sempre estar ao nosso lado nos guiando e protegendo.

Ao Professor Alex Sandro Romeo de Souza Poletto pela orientação, paciência e incentivos dados para a realização do projeto.

Aos meus pais pela ajuda em conceitos e incentivos durante a realização do trabalho.

A minha namorada Isadora Cruz, pela paciente e ajuda em todos os momentos para que não desistisse e concluísse o projeto.

RESUMO

Tendo em vista a demanda para pequenas ou grandes empresas que possuem a tarefa de realizar o pagamento de seus funcionários. Um Software de automação para gerenciar o pagamento desses funcionários faz com que simplifique e facilite a tarefa. O software é focado no modulo de calculo de folha de pagamento seguindo os parâmetros estabelecidos pela CLT “Consolidação das Leis Trabalhista”. Em suma o mesmo tem a função de realizar o calculo sobre a folha de pagamento a partir do cadastro dos funcionários, fazendo com que a operação se torne o mais simples possível para o administrador ou operador do sistema.

ABSTRACT

Given the demand for small or large companies that have the task to make the payment of its employees. Automation software to manage the payment of these employees makes simplify and facilitate the task. The software module is focused on the calculation of payroll according to the parameters established by CLT "Consolidation of Labor Laws". In short it has the function to perform the calculation on the payroll from the register of companies and an employee, making the operation becomes as simple as possible for the accountant or the system operator.

LISTA DE FIGURAS

SUMÁRIO

1. INTRODUÇÃO

O sistema será desenvolvido a partir do incentivo de familiares que trabalham na área de ciências contábeis, onde se encontra uma necessidade grande de software para gerenciamento da folha de pagamento.

Devido ao grande numero de empresas na qual um escritório contábil atende, existe a necessidade de um software para gerenciar toda a movimentação sobre a folha de pagamento dos funcionários de todas essas empresas.

Com base nessa necessidade, alimenta se a importância de um software seguro, completo e de fácil utilização.

Com o sistema Self Pay, o Administrador da empresa consegue realizar o pagamento de seus funcionários, através de um software implementado em seu próprio estabelecimento, tornando o processo mais ágil e fazendo com tenha todo controle sobre o pagamento de seus funcionários.

O desenvolvimento do software por sua vez é uma tarefa difícil, por envolver muitas regras de negocio desconhecidas e conter diferentes parâmetros de cálculos sobre suas regras de negocio.

O sistema será desenvolvido para uma empresa de vendas, que possui um numero considerável de funcionários. Assim será possível fazer a prévia do calculo da folha de pagamento de seus funcionários no próprio estabelecimento de vendas.

Sendo assim, será necessário o estudo sobre a área contábil para que seja possível fazer um software que atenda as necessidades diárias dos contadores.

1.1 OBJETIVOS

O sistema tem por objetivo fazer com que s tarefas para o calculo da folha se tornem mais fácil e prático, para que os mesmos possam atingir um numero maior de funcionários atendidos, fazendo com que o rendimento do trabalho seja maior.

O sistema também faz com que o proprietário tenha total controle sobre a folha de pagamento de seus funcionários.

Além disso, pretende-se também, assegurar a integridade e segurança das informações alimentadas no sistema, bem como, facilitar o gerenciamento na folha de pagamento de inúmeros funcionários ao mesmo tempo.

Para se realizar a folha de pagamento de funcionários de uma empresa é necessário a realização de diferentes cálculos sobre cada tipo de empresa ou cargo. Sendo assim, a informatização dessas operações garantirá que ocorram menos falhas, aumentando a agilidade nos procedimentos.

1.1 DESCRIÇÃO

Em um escritório contábil existem diversos módulos que englobam o software de automação. O software é focado no modulo sobre a folha de pagamento, cuja função é de realizar o cálculo da folha de pagamento, a partir do cadastro dos funcionários junto com os parâmetros de cálculos de acordo com a CLT - “Consolidação das leis trabalhista”. O sistema possui o cadastro de seus funcionários, cargos, departamentos, verbas. Utilizando interfaces de fácil utilização e manuseio.

1.4 JUSTIFICATIVA

A folha de pagamento de uma empresa com muitos funcionários torna a tarefa realizada manualmente algo demorado e com possíveis chances de erros humanos, principalmente com relação aos diferentes tipos de cálculos tais como: Previdência Social, IRRF, Salário Família, dentre outros. Além disso, as empresas são obrigadas nos termos do art. 225 do Decreto nº 3.048/99 a elaborar folha de pagamento mensal da remuneração paga, devida ou creditada a todos os segurados a seu serviço, de forma coletiva por estabelecimento, por obra de construção civil e por tomador de serviços, com a correspondente totalização e resumo geral.

Com um software para realizar o calculo da folha de pagamento dentro do próprio estabelecimento, o processo será realizado com mais rapidez e facilidade, e fazendo com que o proprietário tenha total controle sobre os salários a serem pagos.

Sendo assim a necessidade de um software para gerenciamento de folha de pagamento se torna não somente uma vantagem em termos de operacionais como se agrega também em termos legais.

2. TECNOLOGIAS UTILIZADAS

Neste capitulo é apresentado às tecnologias que foram utilizadas para o desenvolvimento do trabalho de conclusão de curso, desde linguagem de programação á ferramentas e aplicações utilizadas para programação.

2.1 ORIENTAÇÃO A OBJETOS

A orientação objeto é uma tecnologia que enxerga os sistemas como sendo coleção de objetos integrantes. Ela permite melhorar a reusabilidade extensibilidade dos softwares.

A tecnologia orientada a objeto é fundamentada no que, coletivamente chamamos de modelo de objeto, que englobam os princípios da abstração, hierarquização, encapsulamento, classificação, modularização, relacionamento, simultaneidade e persistência. Dento de PPO (programação Orientada Objeto) se encontra diversos conceitos.

2.1.2 CLASSE

Um dos conceitos é sobre a classe, que pode ser definida como uma estrutura de dados que combina estado (campos) e comportamentos (métodos e outros membros) suportam os mecanismos de herança e polimorfismo, que permitem uma classe “filho” estender e especializar uma classe “pai”.

A classe é declarada especificando os atributos e modificadores da classe, assim como o seu nome e também uma classe “pai” ou interface (caso elas existam).

Instancias da classe, objetos são criados a partir de um método construtor. Assim quando criada uma instancia a mesma aloca memória para o objeto criado, ”invoca” o método construtor para inicializar o objeto e retorna a referência para a instância.

O objeto alocado, logo que não mais referenciado, será limpo da memória pelo método destrutor.

Em uma classe temos:

- **Constantes:** Os valores constantes associados á classe.
- **Campos:** As variáveis da classe.
- **Métodos:** O que será executado pela classe.
- **Propriedades:** As ações para leitura e escrita de propriedades.
- **Indexadores:** As ações associadas com instâncias indexadas da classe.
- **Tipos:** Classes declaradas dentro de outra classe.
- **Operadores:** Os operadores de conversão e expressão suportados pela classe.

- **Construtores:** As ações necessárias para inicializar as instâncias da classe ou ela mesma.
- **Destrutores:** As ações executadas antes da instância da classe serem removidos.
- **Eventos:** As notificações que podem ser geradas pela classe.

2.1.3 Objeto

Existe também o conceito sobre objeto, que representa uma instância de uma classe.

O objeto tem a Identidade, ou seja, todo objeto é único e pode ser distinguido de outros objetos.

O objeto possui o estado, determinado pelos dados contidos em si e também comportamentos, definidos pelos métodos e operações que o objeto disponibiliza.

O objeto representa algo do mundo real, por exemplo, um cliente, funcionário, imóvel e etc.

Para que o objeto seja instanciado utilizamos os construtores.
Um construtor sem parâmetro é um construtor padrão, caso não seja definido na classe o construtor, então é gerado automaticamente o construtor padrão assim como seus campos são gerados com valores padrões.
O construtor é declarado do mesmo modo que um método, exceto que não tem tipo de retorno e deve ter o mesmo nome da classe.

2.1.4 Modificadores de acesso

Outro conceito importante citado anteriormente é sobre os modificadores de acesso, que controlam a visibilidade e acesso de membros e métodos da classe.

Temos:

- **Public:** Acesso ilimitado.
- **Protected:** Acesso limitado ao this (isto) e classes derivadas (filhas).

- **Internal:** Acesso limitado ao assembly.
- **Protected Internal:** acesso limitado ao assembly e classes derivadas.
- **Private:** Acesso limitado à classe.

Esses são os parâmetros utilizados para a visibilidade dos membros e métodos das classes.

2.1.5 Métodos e Atributos

Em “POO” temos os métodos, onde é definido o que será executado pela classe.
Existem os métodos estáticos, que são acessados pela classe. Existem os métodos de instância (não estáticos) que são acessados através da instância da classe.

Em um método pode se ter uma lista de parâmetros, sendo as referências ou valores de variáveis que são passadas para o método, na qual podem ou não possuir um tipo de retorno. A assinatura de um método deve ser única na classe, que é formada por nome, numero, modificadores e tipos de parâmetros. Existem:

- **Métodos - sobrecarga:** Permite que métodos de uma mesma classe possuam o mesmo nome porem com assinaturas diferentes, quando sobrecarregados, os métodos são localizados de acordo com sua lista de parâmetro.
- **Métodos - Parâmetros:** São utilizados para passar valores ou variáveis por referência para os métodos**,** assim os parâmetros obtêm os valores dos argumentos que são passados para os métodos em sua execução.

Atributos

Os objetos do mundo real possuem propriedades que possuem valores. Estes valores definem o estado do objeto.

As propriedade recebem o nome de atributos em “POO.”

Podem ser definidos os atributos dos objetos como sendo variáveis ou campos que armazenam os diferentes valores que as características dos objetos contem.
O estado de um objeto é o conjunto de valores de seus atributos em um determinado instante.

O comportamento de um objeto é como ele age e reage em termos de suas mudanças de estado e troca de mensagens com outros objetos. Os atributos de um objeto somente mudam de valor através de estímulos externos ou internos. A única forma de modificar os atributos dos objetos é disparando eventos que provocam a transição desses estados no objeto. Entre outros conceitos.

2.2 Banco de dados “BD”

A importância da informação para a tomada de decisões nas organizações tem impulsionado o desenvolvimento dos sistemas de processamento de informações. Sendo assim temos alguns conceitos que partem dessa evolução.

- **“BD” Banco de dados:** Consiste em uma coleção ou conjunto de dados armazenados manualmente ou automatizado, que por sua vez são inter-relacionados, e que na maioria das vezes contém informações sobre um empreendimento particular.

- **“SGBD” Sistema de gerenciamento de banco de dados:** É um software com recursos específicos para facilitar a manipulação das informações de um “BD” e o desenvolvimento de programas aplicativos.

- **"SBD" Sistema de Banco de Dados:** Consiste em um sistema de manutenção de registros por computador envolvendo quatro componentes principais: dados, hardware, software e usuários. Um SBD possui como objetivos isolar o usuário de detalhes mais internos do BD, uma abstração de dados e prover independência de dados às aplicações sendo a estrutura física de armazenamento e estratégia de acesso.

2.2.1 Abstração de dados

O sistema de banco de dados "SBD" deve prover uma visão abstrata de dados para os usuários, isolando, desta forma, detalhes mais internos do "BD". A abstração se dá em três níveis:

Nível físico: Chamado também de “Esquema interno”, é o nível mais baixo de abstração. Descreve como os dados estão realmente armazenados, englobando estruturas complexas de baixo nível.

Nível conceitual ou lógico: Conhecido também como “Esquema Conceitual”, descreve quais os dados estão armazenados e seus relacionamentos. Neste nível, o BD é descrito através de estruturas relativamente simples, que podem envolver estruturas complexas no nível físico.

Nível de visões do usuário: É o nível externo, descrevendo partes do BD que serão visualizadas pelos usuários de acordo com suas necessidades. Uma visão é um subconjunto de dados do BD, sem que exista a necessidade de estarem armazenados no BD.

Esta arquitetura de três níveis, além de prover a abstração de dados, provê também a independência lógica e física dos dados. Uma independência lógica possui a capacidade de mudar o esquema conceitual sem a necessidade de modificar programas da aplicação e esquemas externos, enquanto que a física tem a capacidade de mudar o esquema interno sem a necessidade de alterar os esquemas conceitual e externo. O acréscimo de uma informação num

esquema conceitual é um exemplo de independência lógica e a reorganização física dos arquivos e a criação de estruturas de acesso adicionais.

2.2.2 Modelos lógicos de Dados

É o conjunto de ferramentas conceituais para a descrição dos dados, dos relacionamentos entre os mesmos e das restrições de consistência e integridade.
Em outras palavras, é o conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados, ou seja, tipos de dados, relacionamentos e restrições que devem ser mantidas sobre os dados.

Os modelos dividem-se em dois grupos: baseados em objetos e baseados em registros. No modelo orientado a objetos, o código executável é parte integrante do modelo de dados enquanto que no orientado a registros, o banco é estruturado em registros de formato fixo, onde cada registro tem sua coleção de atributos. Além disto, o modelo baseado em registro possuem linguagens para expressar consultas e atualizações no BD. Alguns modelos:

- **Modelo de Redes:** Inicialmente apresentado no relatório do grupo de trabalho CODASYL (Conference on Data Systems Language), em 1971, sendo chamado de modelo DBTG (Data Base Task Group). Foi revisado em 1978 e 1981. Apresenta como características os dados organizados em vários tipos de registros e tipos de sets (estabelecem os relacionamentos) e um problema de estratégia de acesso aos registros “filhos”.

- **Modelo Hierárquico:** Da mesma forma que o modelo de redes, no modelo hierárquico os dados são organizados em vários tipos de registros e o relacionamento dos registros “pai” e “filho” são explícitos. Apresenta como vantagens a simplicidade de processamento e o uso natural para organizações hierárquicas de dados.

- **Modelo Relacional:** Introduzido por E.F. Codd, em 1970, possui como características: base de dados visualizada como um conjunto de tabelas, cada uma representando uma relação, relacionamentos representados por valores de dados, simetria nas consultas.
  É uma desvantagem do modelo relacional a representação não natural de objetos complexos.

2.2.3 DDL e DML

O esquema de um banco de dados é a estrutura gera do BD. Sua estrutura não muda com frequência e há um esquema para cada nível de abstração e um sub-esquema a cada visão do usuário. Um esquema pode ser representado de uma forma textual ou gráfica. Normalmente, utiliza-se o Diagrama de Entidade Relacionamento Para a representação gráfica do esquema de um BD.

Os modelos de dados baseados em registros possuem linguagens para expressar consultar e atualizações no BD, chamadas e Linguagens de SGBD. Algumas delas atuam sobre o esquema do BD, outras sobre os dados armazenados.

DLL (Data Definition Language “Linguagem de Definição de Dados”)

Permite especificar o esquema do BD, através de um conjunto de definições de dados.
O esquema do BD fica armazenado numa área do SGBD chamada de Dicionário de Dados. Na maioria das vezes, uma DDL atua nos esquemas conceitual e interno.

DML (Data Manipulation Language “Linguagem de Manipulação de Dados”)

Em relação a BD, a manipulação dos dados dá-se pelas seguintes operações: A recuperação da informação armazenada .Está recuperação é no sentido de tornar a informação armazenada no banco disponível ao usuário ou programa aplicativo.
Consistem em outras operações, a inserção de novas informações, exclusões de informações e modificação de dados armazenados.

Uma DML permite ao usuário acessar ou manipular os dados, vendo-os da forma como são definidos no nível de abstração mais alto do modelo de dados utilizado.
No nível mais alto, no de visões de usuário, uma DML é: não-procedural, declarativa, orientada a conjuntos e dita linguagem de consulta (query language).
Em baixo-nível, no físico, uma DML é orientada a registros e considera uma sublinguagem de dados, embutida em uma linguagem hospedeira.

2.2.4 SQL “Structured Query Language”

A linguagem SQL é a linguagem de pesquisa declarativa padrão para banco de dados relacional. Muitas das características originais do SQL foram inspiradas na álgebra relacional.

A linguagem SQL surgiu em 1974 e foi desenvolvida nos laboratórios da IBM como interface para o Sistema Gerenciador de Banco de Dados Relacional (SGBDR) denominado SYSTEM R. Esse sistema foi criado com base em um artigo de 1970 escrito por Edgar F. Codd.

Alguns dos principais comandos SQL para manipulação de dados são: INSERT (inserção), SELECT (consulta), UPDATE (atualização), DELETE (exclusão). SQL possibilita ainda a criação de relações entre tabelas e o controle do acesso aos dados.

INSERT: O comando INSERT é usado para inserir um registro a uma tabela existente.

SELECT: O Select é o principal comando usado em SQL para realizar consultas a dados pertencentes a uma tabela.

UPDATE: É utilizado para mudar os valores de dados em uma ou mais linhas da tabela existente.

DELETE: O comando DELETE permite remover linhas existentes de uma tabela.

Junto aos comandos da linguagem são utilizadas as cláusulas, que por sua vez são condições de modificação utilizadas para definir os dados que deseja selecionar ou modificar em uma consulta.

O SQL possui também operadores relacionais e lógicos, que são usados para realizar comparações entre valores, em estruturas de controle.

2.3 FERRAMENTAS DE DESENVOLVIMENTO

Para o desenvolvimento do sistema será necessário a utilização de alguns programas, frameworks e Equipamento.

- Microsoft Visual Studio 2012: O Microsoft Visual Studio é um pacote de programas da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e J# (J Sharp) e para desenvolvimento na área web, usando a plataforma do ASP.NET.

- Astah Community: é um software para modelagem UML, desenvolvido na plataforma Java, o que garante sua portabilidade para qualquer plataforma que possui uma máquina virtual Java.

- Microsoft Visio Professional 2010: O Microsoft Visio é um aplicativo para criação de diagramas para o ambiente Windows, utilizado para gerar diagramas de diversos tipos, como organogramas, fluxogramas, modelagem de dados (usando UML ou outra notação gráfica qualquer), diagramas de redes, plantas baixas, cartazes, etc.

- DBDesigner: é um software de modelagem visual de banco de dados.

3. DIAGRAMAS

Neste capitulo será apresentado imagens dos diagramas utilizados na faze de planejamento e projetação do trabalho.

A figura a baixo retrata o diagrama de caso de uso geral, onde possui o ator e as funcionalidades que é o ator pode realizar no sistema.

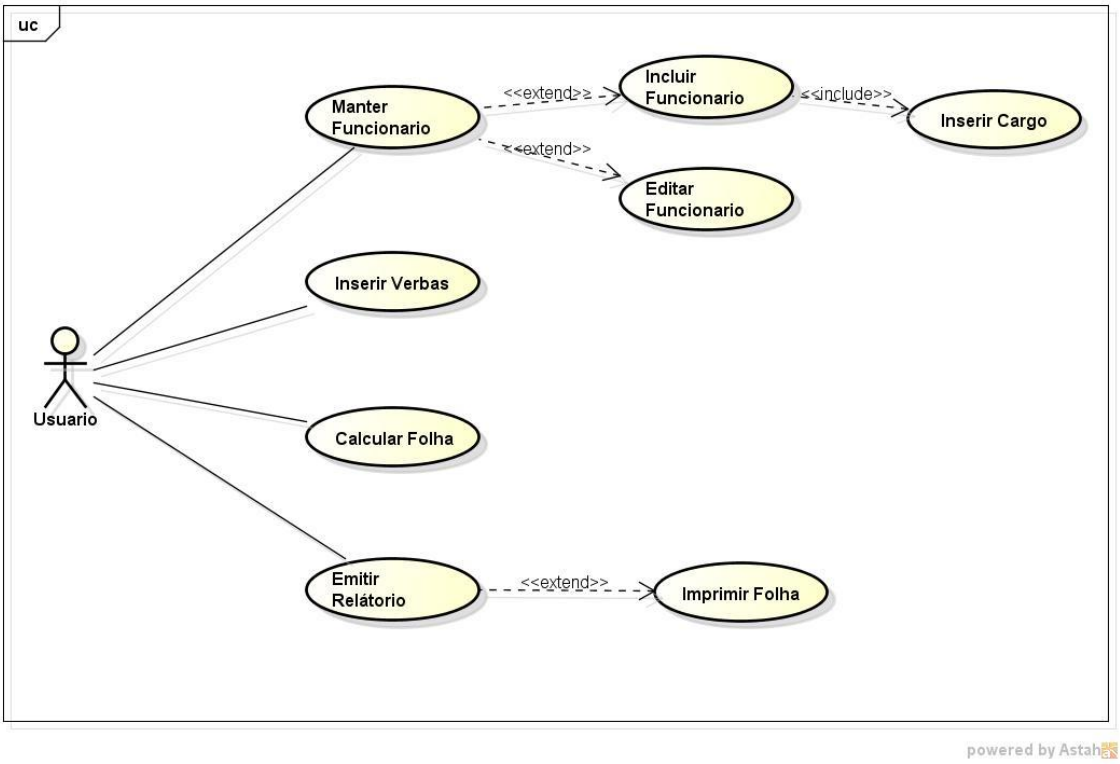

Figura 1 – Caso de Uso geral

A figura abaixo reporta a funcionalidade de inserção de cargo realizada pelo ator Usuário.

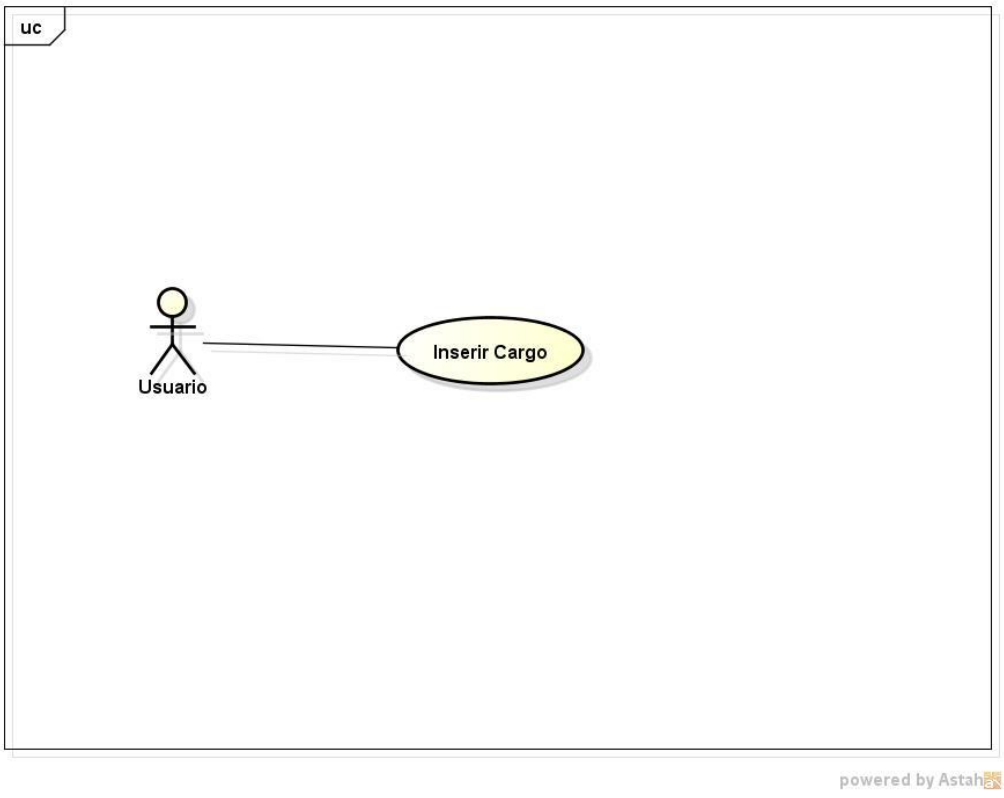

Figura 2 – Caso de uso Inserir cargo

A figura abaixo reporta a funcionalidade de inserção de verbas realizada pelo ator Usuário.

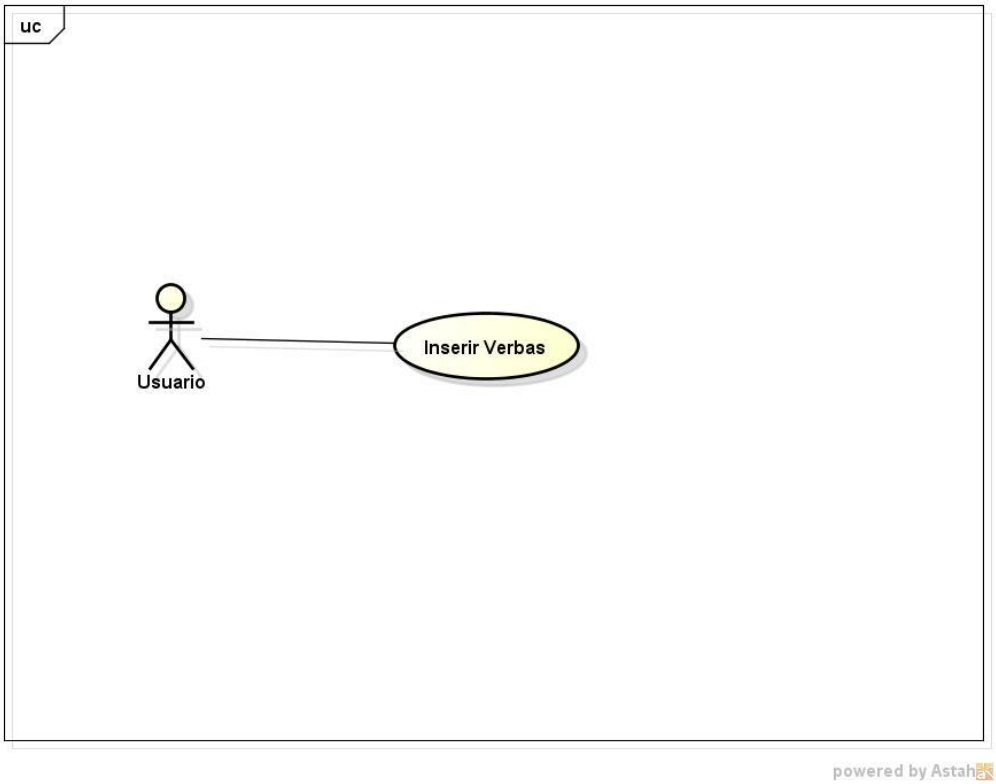

Figura 3 – Caso de Uso Inserir Verba

A figura abaixo reporta a funcionalidade de Emitir Relatório realizada pelo ator Usuário.

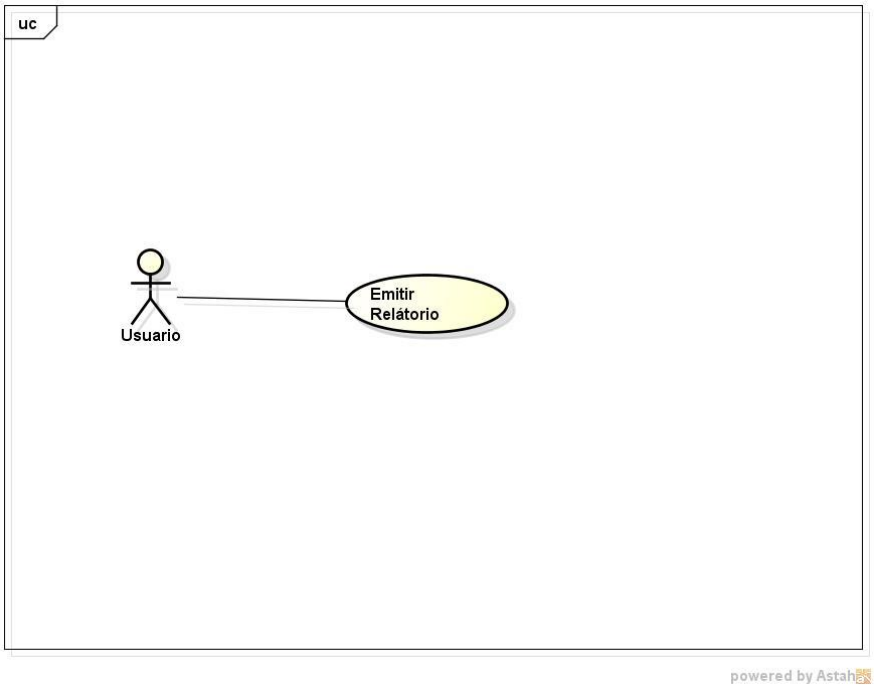

Figura 4 – Caso de Uso Emitir Relatório

A figura abaixo Mostra o digrama de classes onde é apresentado algumas das principais classes do sistema com seus relacionamentos.

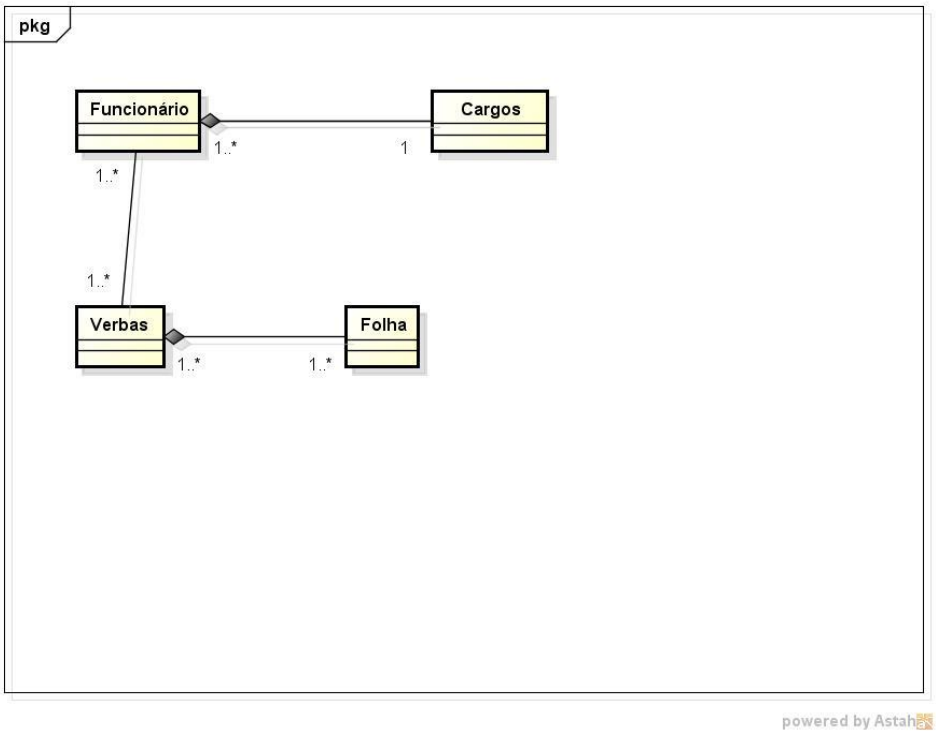

Figura 5 – Diagrama de Classes

3.1 NARRATIVA DO CASO DE USO

Abaixo é apresentada a narrativa de Alguns casos de uso do software.

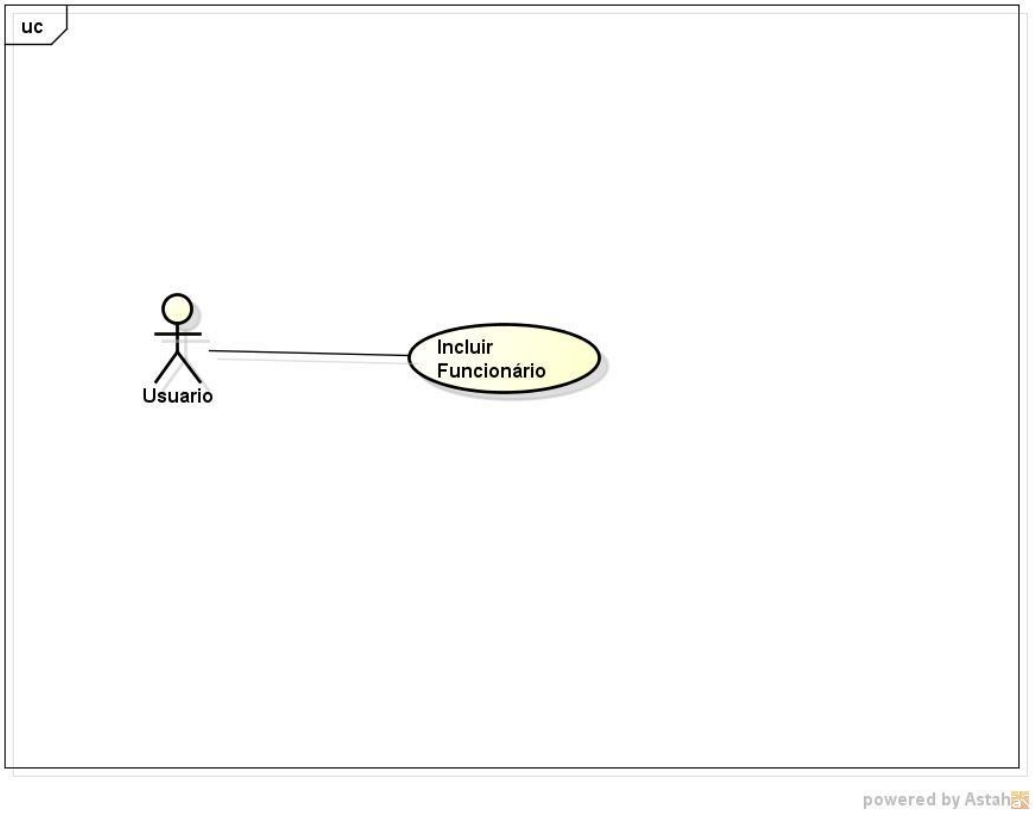

Figura 6 – Caso de Uso Incluir Funcionário

Caso de uso: Incluir Funcionário

1. Finalidade/Objetivo

- Permite que o contador possa realizar a inclusão sobre o cadastro do funcionário associado a uma empresa.

2. Atores

- Contador

3. Pré-Condições

- O Operador deve ter realizar o login no sistema.
- O operador deve acessar a opção sobre cadastro de funcionário.

4. Evento Inicial

O operador acessa na tela inicial do sistema e selecionar a opção de cadastro de Funcionários.

5. Fluxo Principal

a) O sistema Apresenta a tela de cadastro, onde solicita os dados necessários para o cadastro do Funcionário.
b) O operador preenche os campos solicitados para o cadastro do Funcionário e seleciona para salvar o registro.
c) O sistema realiza a verificação sobre os dados informados em seguida realiza a inclusão do registro.
d) o caso de uso é encerrado.

6. Fluxos Alternativos

A1. Cancela a Operação

a) O operador cancela a operação, após ter ou não informado algum dado.
b) O sistema Retorna ao passo 5.a do Fluxo Principal.

7. Fluxos de Exceção

E1. Erro na validação de dados

a) O sistema realiza a verificação sobre os dados informados e os mesmos possuem inconsistências
b) O sistema retorna a mensagem sobre o campo incorreto e retorna ao passo 5.a do Fluxo Principal.

E2. Duplicidade de Registro

a) O sistema verifica que já possui outro registro armazenado referente ao que está sendo cadastrado.
b) O sistema reporta a duplicidade e o código do registro, em seguida o caso de uso é encerrado.

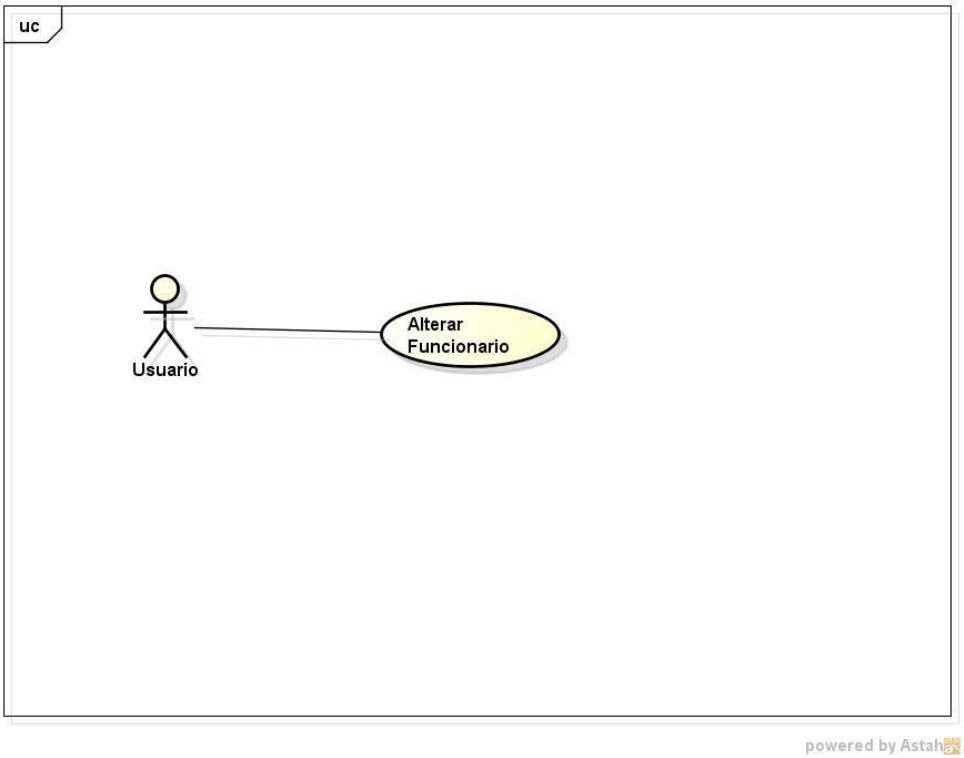

Figura 7 – Caso de Uso Alterar Funcionário

Caso de uso: Alterar Funcionário

1. Finalidade/Objetivo

- Permite que o contador possa realizar a Alteração sobre o cadastro do funcionário.

2. Atores

- Contador

3. Pré-Condições

- O Operador deve ter realizar o login no sistema.
- O operador deve acessar o cadastro do Funcionário.

4. Evento Inicial

- O operador Realize a busca e acessa o cadastro do Funcionário.

5. Fluxo Principal

a) Após a busca, o sistema apresenta a tela de cadastro, onde o operador seleciona a opção para realizar a alteração.
b) O operador realiza a alteração no cadastro e seleciona a opção para salvar o registro.
c) O sistema realiza a verificação sobre os dados alterados em seguida salve a alteração no registro.
d) o caso de uso é encerrado.

6. Fluxos Alternativos

A1. Cancela a Operação

a) O operador cancela a operação, após ter ou não informado algum dado.
b) O sistema Retorna ao passo 5.a do Fluxo Principal, sem nenhum alteração no cadastro.

7. Fluxos de Exceção

E1. Erro na validação de dados

a) O sistema realiza a verificação sobre os dados Alterados e os mesmos possuem inconsistências
b) O sistema retorna a mensagem sobre o campo incorreto e retorna ao passo 5.a do Fluxo Principal.

E2. Duplicidade de Registro

a) O sistema verifica que já possui outro registro armazenado referente ao que está sendo alterado.
b) O sistema reporta a duplicidade e o código do registro, em seguida o caso de uso é encerrado.

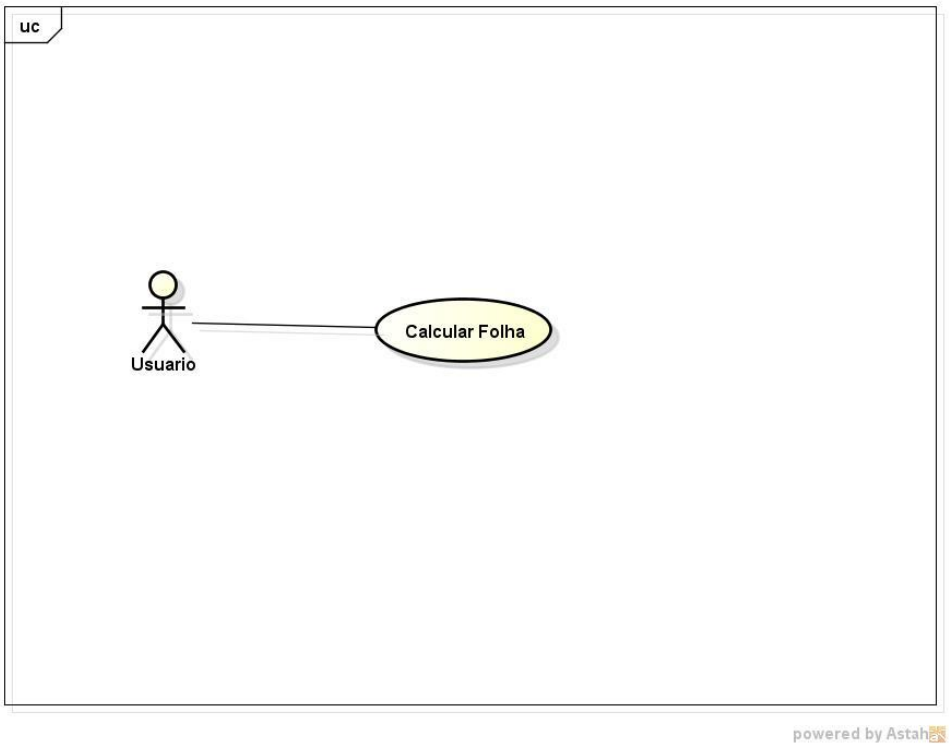

Figura 8 – Caso de Uso Calcular Folha

Caso de uso: Calcular Folha

1. Finalidade/Objetivo

- Permite que o contador possa realizar o calculo sobre a folha de pagamento do respectivo funcionário.

2. Atores

- Contador

3. Pré-Condições

- O Operador deve ter realizar o login no sistema.
- Deve conter o cadastro da empresa e funcionário.
- As verbas devem estar associadas ao cadastro do funcionário.

4. Evento Inicial

O operador acessa na tela inicial do sistema e selecionar a opção mestre de movimento.
Em seguida o operador selecionar a opção de calcular folha de pagamento.

5. Fluxo Principal

a) O sistema Apresenta a tela de mestre de movimento para importar os parâmetros atuais selecionar o tipo de folha e assim libera para realizar o calculo.
b) O Operador informa os parâmetros a respectiva empresa e selecionar para iniciar o cálculo
c) O sistema realiza a verificação sobre os dados informados em seguida realiza o calculo.
d) o caso de uso é encerrado.

6. Fluxos Alternativos

A1. Cancela a Operação

a) O operador cancela a operação, assim não é realizado o calculo da empresa.
b) O sistema Retorna ao passo 5.a do Fluxo Principal.

7. Fluxos de Exceção

E1. Erro na validação de dados

a) O sistema realiza a verificação sobre os dados informados e os mesmos possuem inconsistências
b) O sistema retorna a mensagem para que seja feito o procedimento novamente e retorna ao passo 5.a do Fluxo Principal.

4. CÁLCULOS IMPORTANTES

Esse capitulo contem o método teórico sobre cálculo da folha de pagamento aplicado no software desenvolvido.

Para realizar o cálculo sobre a folha mensal de pagamento com o acréscimo de horas extras, segue o seguinte modelo.

Com base na quantidade de salários mínimos que o funcionário recebe, é feito a multiplicação da quantidade de salários mínimos pelo valor do salário mínimo atual, tendo assim o valor do salário base.

Em seguida é feito o calculo da hora extra, que é transformada em adicional para o calculo do salário liquido. O calculo é feito pelo valor do salário dividido pelo numero de horas que foram trabalhadas no mês, com resultado que será o valor da hora trabalhada é feito o adicional de 50% do mesmo, que multiplicado pelo numero de horas extras realizas se torna o valor de horas extras. Em seguida podemos realizar o calculo de INSS, sendo o valor da soma entre o salário base adicionado ao valor das horas extras, o resultado é multiplicado pela alíquota corresponde ao valor do salário de acordo com a tabela vigente do INSS, o resultado é valor de desconto do INSS.

Outro desconto sobre a folha é referente ao IRRF (Imposto de renda retido na fonte), que é calculado pelo Salário Base subtraído pelo INSS, subtraído também pelo valor da soma do numero de dependentes. Com o resultado é aplicado o desconto da multiplicação do salário base pela alíquota correspondente ao valor do salário de acordo com a tabela vigente do IRRF.

Em soma o valor do salário líquido mensal é o salário base subtraído pelo valor de calculo do INSS e subtraído pelo valor de calculo do IRRF.

Serão apresentados alguns trechos de código como exemplo da formula citada a cima.

Abaixo é apresentado um trecho do código fonte, da formula utilizada na implementação do software para realizar o calculo de hora extra.

```
decimal valor3 = 0;
                    int jornada = new
FuncionarioDAL().verificaJornada(Convert.ToInt32((lblcodfun.Text)));
                    foreach (DataGridViewRow row in dgverbas.Rows)
                    {
                        if (Convert.ToInt32(row.Cells[1].Value) == 3)
                        {
                            valor3 = Convert.ToDecimal(row.Cells[4].Value);
                        }
                    }

                    valorhora = valor3 / jornada;
                    horaextra = valorhora * Convert.ToDecimal(1.5) * quant;

                    txtValor.Text = Convert.ToDecimal(horaextra).ToString();

                    break;
```

Abaixo é apresentado um trecho do código fonte, da formula utilizada na implementação do software para realizar o calculo sobre o valor do salário bruto.

```
int numsal = new FuncionarioDAL().verificaSal(Convert.ToInt32((lblcodfun.Text)));

                    decimal salariomin = new SalarioMinDAL().verificasalmin();

                    salariobase = numsal * salariomin;

                    txtValor.Text = Convert.ToDecimal(salariobase).ToString();

                    break;

                   default:

                   Console.WriteLine("Erro");

                    break;
```

5. CRONOGRAMA

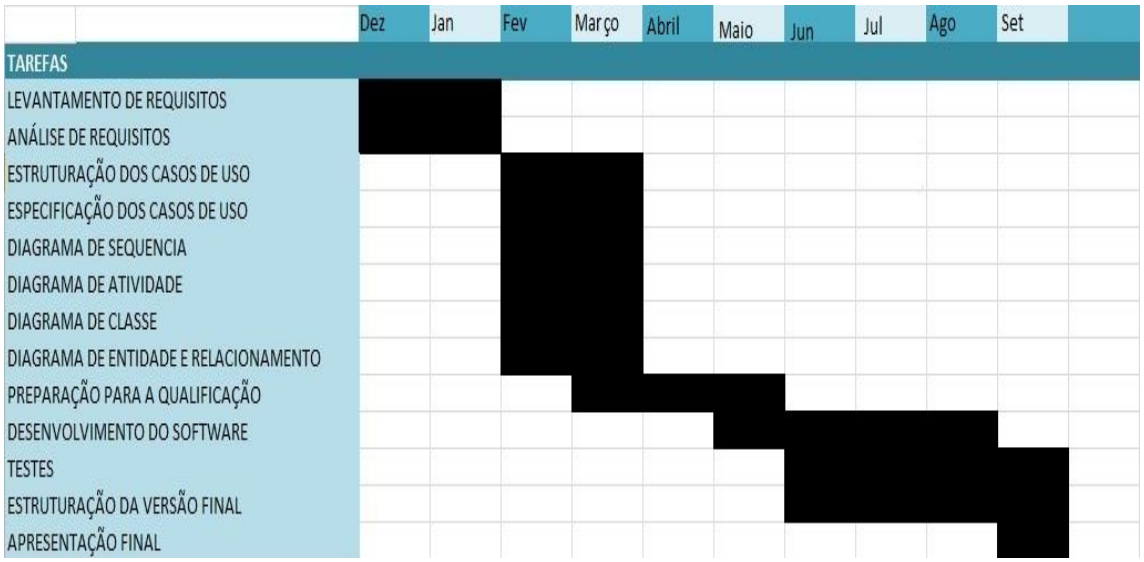

Figura 9 – Cronograma de Tarefas

6. CONCLUSÃO

Com a utilização do software para gerenciamento de folha de pagamento será possível que o Funcionário de uma loja de Vendas possa fazer em seu próprio estabelecimento á prévia do calculo sobre a folha de pagamento de seus funcionários, onde em uma pequena empresa pode ser implantado o sistema para que possa ser feita pela própria empresa o calculo da folha, com facilidade, praticidade e segurança sobre os dados e cálculos. Tornando o processo de pagamento mais ágil simples e acessível ao estabelecimento.

Devido a grande a demanda para realizar o calculo de seus inúmeros funcionários, torna se essencial a integração de um software de automação para gerenciar a folha de pagamento, tendo em vista melhorias sobre integridade das informações, menos falhas e um maior rendimento por se tornar algo mais fácil e rápido de realizar.

Além da obrigatoriedade de acordo com a lei art. 225 do Decreto nº 3.048/99 a elaborar a folha mensal de remuneração paga a seu serviço de forma coletiva a todo o estabelecimento.

Sendo assim a criação de um software para gerenciar a folha de pagamento se torna uma grande oportunidade.

7. REFERENCIAS

Deitel, H.M. **C# Como Programar**, Pearson Education.

Edwin Lima/Eugênio Reis. **C# e NET para desenvolvedores.**

José Antonio Ramalho. **Oracle 10G**, Cengage Learning.

Juruá Editora. **CLT Consolidação das Leis do Trabalho**.

Marco Aurélio de Souza. **Oracle Banco de Dados**, Ciência Moderna.

Moacir Quadros. **Gerencia de Projeto de Software** – Técnica e ferramentas.

O'Reilly. **Use a cabeça C#.**

O'Reilly. **Use a cabeça SQL.**