

**UNIVERSIDADE PAULISTA:
CST EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**Rafael Antoniassi Vicechio
RA: n7756c6**

**Clodair Neves Queiroza
RA: g5677e2**

PIM – PROJETO INTEGRADOR MULTIDISCIPLINAR
Projeto Integrado Multidisciplinar

**SÃO JOSÉ DO RIO PRETO
2023**

**UNIVERSIDADE PAULISTA:
CST EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

Rafael Antoniassi Vicechio - n7756c6

Clodair Neves Queiroza - g5677e2

PIM – PROJETO INTEGRADOR MULTIDISCIPLINAR

Projeto Integrado Multidisciplinar

Projeto Integrador Multidisciplinar para
Obtenção do título de tecnólogo em
Análise e Desenvolvimento de Sistemas
Apresentado à Universidade Paulista.

**SÃO JOSÉ DO RIO PRETO
2023**

RESUMO

Este documento se trata de uma mostra completa referente ao processo de engenharia de requisitos, modelagem UML e criação do banco de dados de um sistema cujo a finalidade é produzir folhas de pagamento de maneira automática.

Tendo como finalidade expor esse extenso processo de maneira minuciosa, o manuscrito engloba o projeto desde suas primeiras definições, passando por diversos diagramas detalhando suas complexidades e terminando na modelagem e criação de um banco de dados.

Em resumo, observa-se nesta escritura uma rica documentação que visa gerar bases sólidas para a produção do sistema aqui descrito.

Palavras-chave: modelagem; documentação; banco de dados; engenharia de requisitos.

ABSTRACT

This document is a complete example of requirements engineering process, UML modeling and database creation of a system whose mission is to produce a payroll automatically.

Aiming to expose this extensive process in detail, the manuscript encompasses the project from its first definitions, going through several diagrams detailing its complexities and ending with the modeling and creation of a database.

In short, this writing contains a wealth of documentation that aims to generate solid foundations for the production of the system described here.

Keywords: modeling; documents; database; requirements engineering.

SUMÁRIO

1 INTRODUÇÃO.....	6
2 ESPECIFICAÇÃO DO PRODUTO.....	6
2.1 DEFINIÇÃO GERAL.....	6
2.2 FUNCIONALIDADE.....	6
3 REQUISITOS.....	7
3.1 REQUISITOS FUNCIONAIS.....	7
3.1.1 RF01: O Sistema deve permitir o cadastro de novos funcionários.....	7
3.1.2 RF02: O Sistema deve permitir a remoção de um funcionário, bem como a alteração de seus dados.....	7
3.1.3 RF03: O Sistema deve permitir que sejam visualizadas todas as folhas de pagamento de determinado funcionário.....	7
3.1.4 RF04: Informações de descontos de salário e faltas deverão ser inseridas no sistema por agente exterior.....	7
3.1.5 RF05: O sistema deverá levar em conta todos os fatores relevantes, como salário bruto, descontos, impostos e etc. no momento do calculo do salário líquido dos funcionários.....	8
3.1.6 RF06: As folhas de pagamento daquele mês deverão, por estandarte, serem geradas e disponibilizadas no sistema em uma data fixa.....	8
3.1.7 RF07: Deverá ser possível antecipar a geração de uma folha de pagamento antes da data prevista.....	8
3.2 REQUISITOS NÃO FUNCIONAIS.....	8
3.2.1 Deverá ser utilizada a linguagem de programação C# para produção do sistema.....	9
3.2.2 Deverá ser utilizado o banco relacional SQL para a criação do banco de dados.....	9
3.2.3 O Sistema deverá ser criado seguindo todos os paradigmas da programação orientada a objetos.....	9
3.3 REGRAS DE NEGÓCIO.....	9
3.3.1 RN01: Folhas de pagamento de funcionários removidos devem permanecer salvas no banco de dados.....	9
3.3.2 RN02: O sistema deverá usar como objeto de pesquisa, invariavelmente, um valor de CPF.....	10

4 DIAGRAMAS UML.....	10
4.1 DIAGRAMA DE CASO DE USO.....	10
4.2 MODELAGEM DE PROCESSOS DE NEGÓCIO.....	11
4.3 DIAGRAMA DE SEQUÊNCIA.....	11
4.4 MODELO DE CLASSES DE DOMÍNIO.....	12
5 BANCO DE DADOS.....	12
5.1 DIAGRAMA ENTIDADE RELACIONAMENTO.....	12
5.2 MODELO LÓGICO.....	14
5.3 SCRIPTS.....	15
5.3.1 Criação do banco.....	15
5.3.2 Criação da tabela para funcionários.....	15
5.3.3 Criação da tabela para dependentes.....	15
5.3.4 Criação da tabela para horas trabalhadas.....	16
5.3.5 Criação da tabela para faltas.....	16
5.3.6 Criação de tabela para benefícios.....	16
5.3.7 Criação de tabela para descontos.....	17
5.3.8 Criação de tabela para impostos.....	17
5.3.9 Criação de tabela para folha de pagamento.....	17

1 INTRODUÇÃO

Este documento contém o escopo completo, engenharia de requisitos, diagramas UML, diagrama entidade relacionamento, modelo lógico e script de um banco de dados referentes ao processo de criação de um sistema de geração de folhas de pagamento.

2 ESPECIFICAÇÃO DO PRODUTO

2.1 DEFINIÇÃO GERAL

O sistema a ser especificado neste documento prestará o serviço de automatizar a geração de folhas de pagamento, otimizando e aperfeiçoando o processo, para que o mesmo atenda os requerimentos de qualidade e possa coexistir e se comunicar bem com outros sistemas.

2.2 FUNCIONALIDADE

Ao abrir o software, o usuário será apresentado com uma tela contendo um campo de pesquisa e um botão.

O botão representará o recurso de adicionar um novo funcionário ao banco de dados. Ao selecionar essa opção, o usuário irá ser direcionado a uma nova tela com um formulário que, por sua vez, conterá campos referentes a todas as informações relevantes para o cadastro.

O campo de pesquisa poderá ser utilizado inserindo um valor de CPF. Caso a pesquisa seja bem-sucedida, o usuário será levado a uma tela onde cards representando todas as folhas de pagamento referentes àquele CPF e geradas até aquele momento aparecerão, sendo possível clicar em qualquer um deles para visualizar os detalhes.

Nesta tela, também, estarão três botões que, em sequência, possibilitarão a visualização dos dados cadastrais daquele funcionário (e a alteração dos mesmos, por meio de outro botão nesta etapa), remoção daquele funcionário do banco de dados e a geração de sua folha de pagamento atual, mesmo que precocemente.

Em caso de exclusão, as folhas de pagamento da pessoa removida ainda ficarão salvas no banco de dados, possibilitando consultas posteriores.

Informações referentes a faltas ou descontos do salário deverão ser inseridas no banco por meio de sistema exterior integrado.

3 REQUISITOS

3.1 REQUISITOS FUNCIONAIS

3.1.1 RF01: O Sistema deve permitir o cadastro de novos funcionários.

Prioridade: Essencial.

Deve ser possível a inserção de novos funcionários no banco de dados, sem nenhuma limitação além da capacidade de armazenamento do equipamento utilizado.

3.1.2 RF02: O Sistema deve permitir a remoção de um funcionário, bem como a alteração de seus dados.

Prioridade: Essencial.

De maneira intuitiva, deve ser possível a remoção de um funcionário bem como a possibilidade de reenviar seu formulário com a finalidade de alterar e/ou atualizar seus dados.

3.1.3 RF03: O Sistema deve permitir que sejam visualizadas todas as folhas de pagamento de determinado funcionário.

Prioridade: Essencial.

Cada unidade de folha deve ter suas informações contida dentro de um card, colapsadas até que o usuário decida clicar para exibir os detalhes.

3.1.4 RF04: Informações de descontos de salário e faltas deverão ser inseridas no sistema por agente exterior.

Prioridade: Essencial.

Um sistema, exterior e integrado com este, terá o propósito de inserir no banco de dados iterações de faltas e descontos de salário sob demanda.

3.1.5 RF05: O sistema deverá levar em conta todos os fatores relevantes, como salário bruto, descontos, impostos e etc. no momento do cálculo do salário líquido dos funcionários.

Prioridade: Essencial.

Para realizar um cálculo correto e preciso, o algoritmo responsável pela definição do valor líquido de um salário deverá consagrar todas as variáveis relevantes.

3.1.6 RF06: As folhas de pagamento daquele mês deverão, por estandarte, serem geradas e disponibilizadas no sistema em uma data fixa.

Prioridade: Essencial.

Para se adequar à prática de pagamento em uma data prevista e constante, o sistema deverá entregar todas folhas de pagamento atuais no quinto dia útil de todo mês.

3.1.7 RF07: Deverá ser possível antecipar a geração de uma folha de pagamento antes da data prevista.

Prioridade: Essencial

Será fornecida a ferramenta para que uma folha de pagamento seja gerada antes do quinto dia útil. A mesma conterá valores condizentes à data em que foi precocemente emitida.

3.2 REQUISITOS NÃO FUNCIONAIS

3.2.1 RNF01: Deverá ser utilizada a linguagem de programação C# para produção do sistema.

Prioridade: Essencial

Sendo orientada a objetos, a linguagem C# apresenta todas as funcionalidades necessárias para a produção de um sistema deste tipo.

3.2.2 RNF02: Deverá ser utilizado o banco relacional SQL para a criação do banco de dados.

Prioridade: Essencial

Levando em conta a alta variabilidade dos dados em questão, o SQL é ideal por conta, entre outros motivos, de sua qualidade relacional.

3.2.3 RNF03: O Sistema deverá ser criado seguindo todos os paradigmas da programação orientada a objetos.

Prioridade: Essencial

Para garantir a qualidade e manutenibilidade do produto, o mesmo deverá ser estruturado no paradigma O.O.P.

3.3 REGRAS DE NEGÓCIO

3.3.1 RN01: Folhas de pagamento de funcionários removidos devem permanecer salvas no banco de dados.

Prioridade: Importante.

Sendo um dado importante, as folhas de pagamento geradas para determinado CPF deverão permanecer salvas no banco de dados mesmo após sua exclusão.

3.3.2 RN02: O sistema deverá usar como objeto de pesquisa, invariavelmente, um valor de CPF.

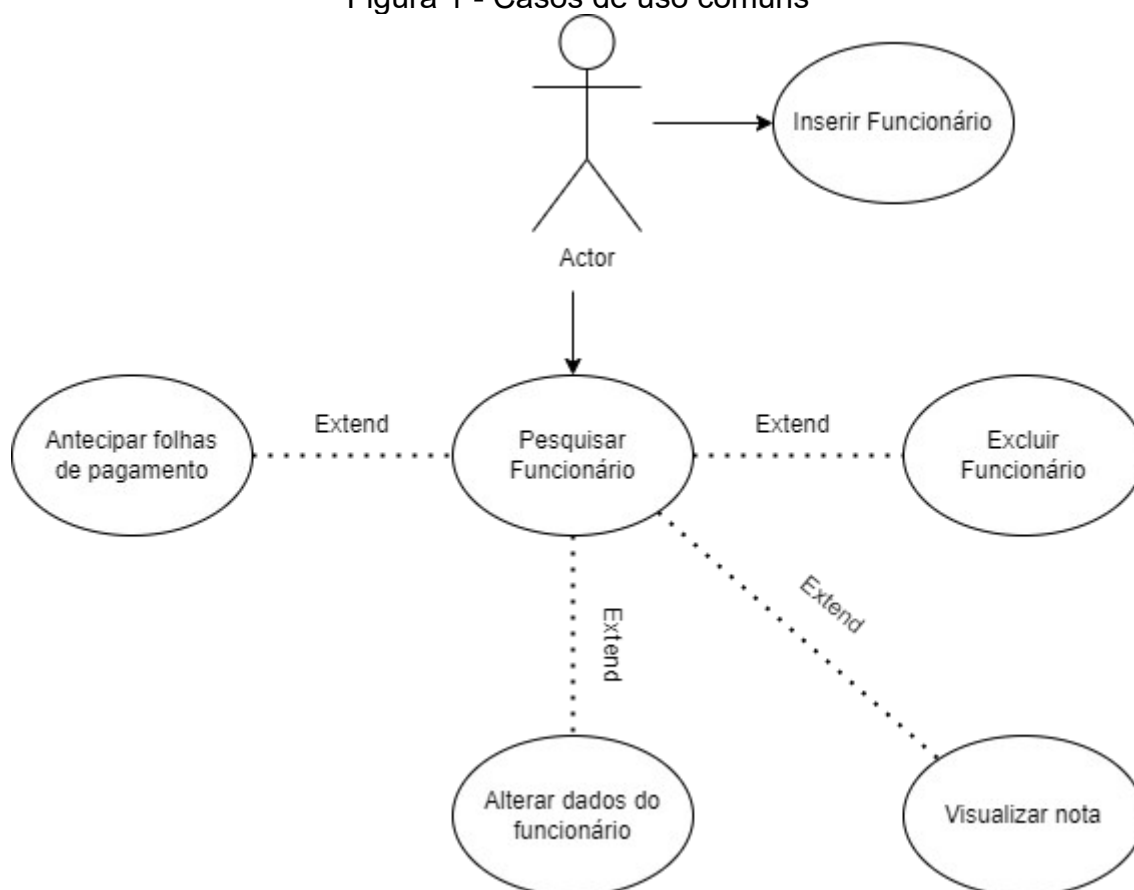
Prioridade: Desejável.

Para fins de prevenção de problemas e garantia de qualidade, o sistema deverá utilizar como objeto de pesquisa apenas a chave primária dos funcionários, seu CPF.

4 DIAGRAMAS UML

4.1 DIAGRAMA DE CASO DE USO

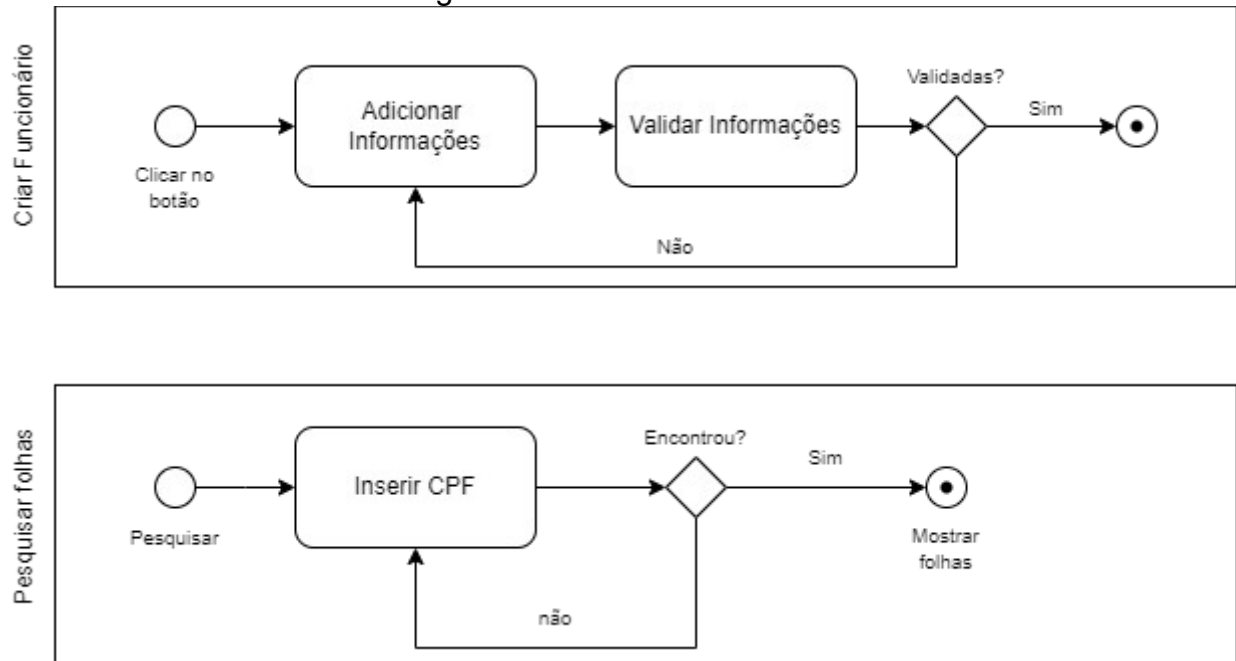
Figura 1 - Casos de uso comuns



Fonte a: app.diagrams.net, 2023

4.2 MODELAGEM DE PROCESSOS DE NEGÓCIO

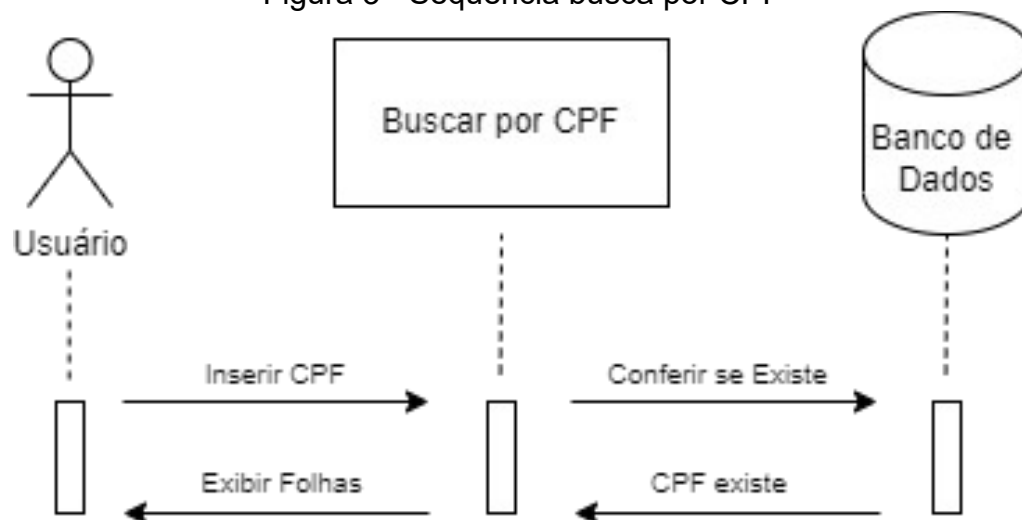
Figura 2 - Processos comuns



Fonte b: app.diagrams.net, 2023

4.3 DIAGRAMA DE SEQUÊNCIA

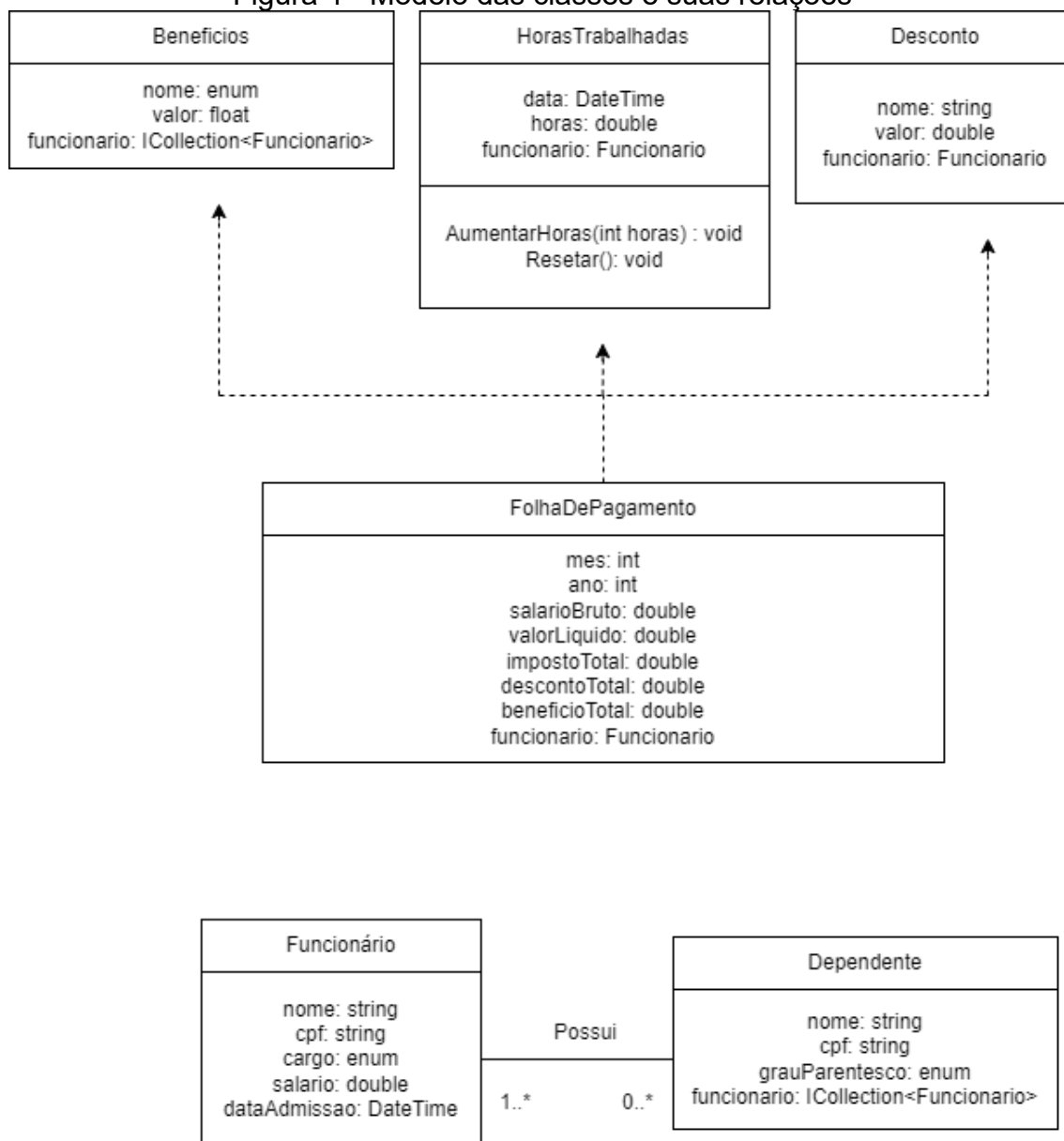
Figura 3 - Sequência busca por CPF



Fonte c: app.diagrams.net, 2023

4.4 MODELO DE CLASSES DE DOMÍNIO

Figura 4 - Modelo das classes e suas relações

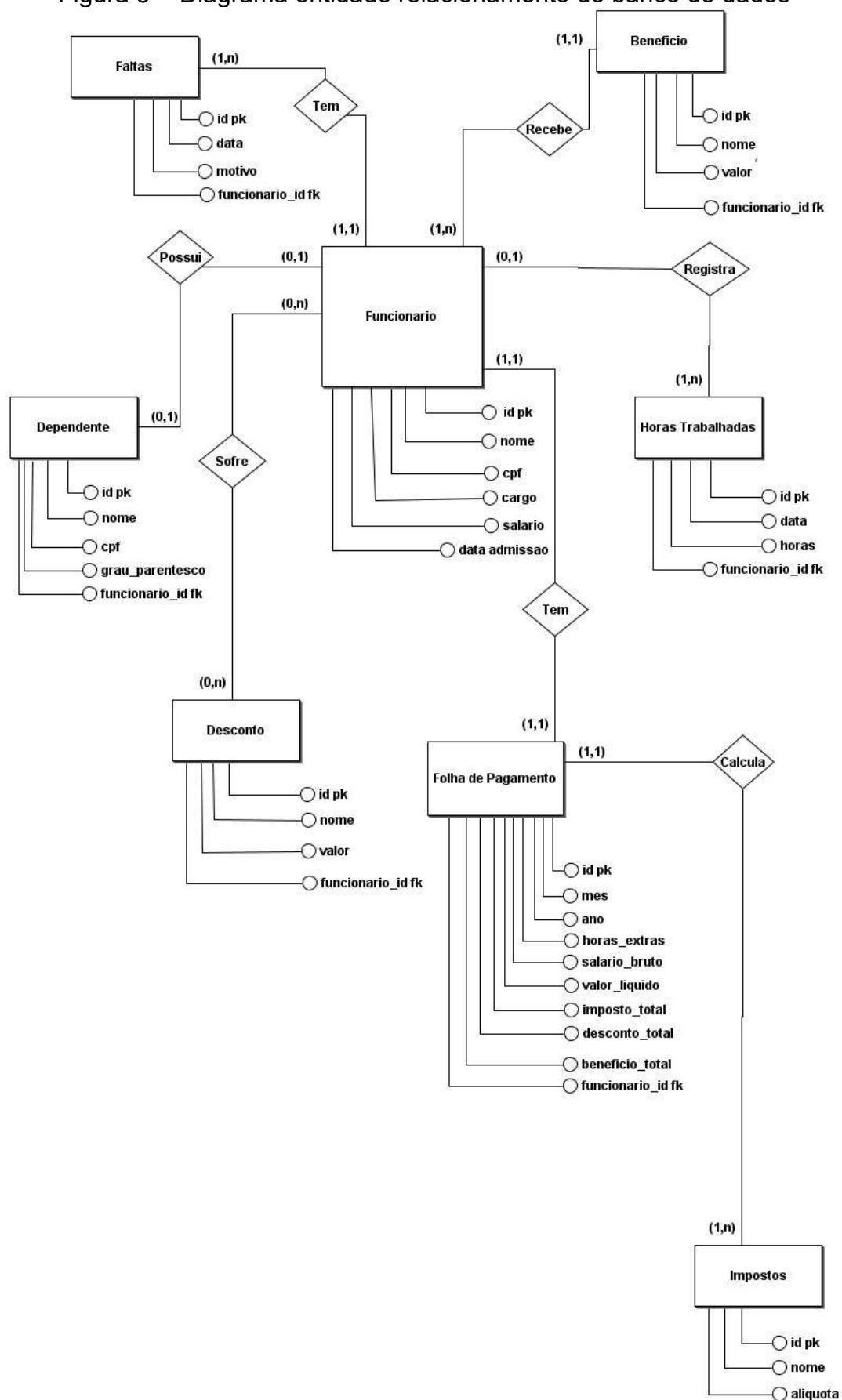


Fonte d: app.diagrams.net, 2023

5 BANCO DE DADOS

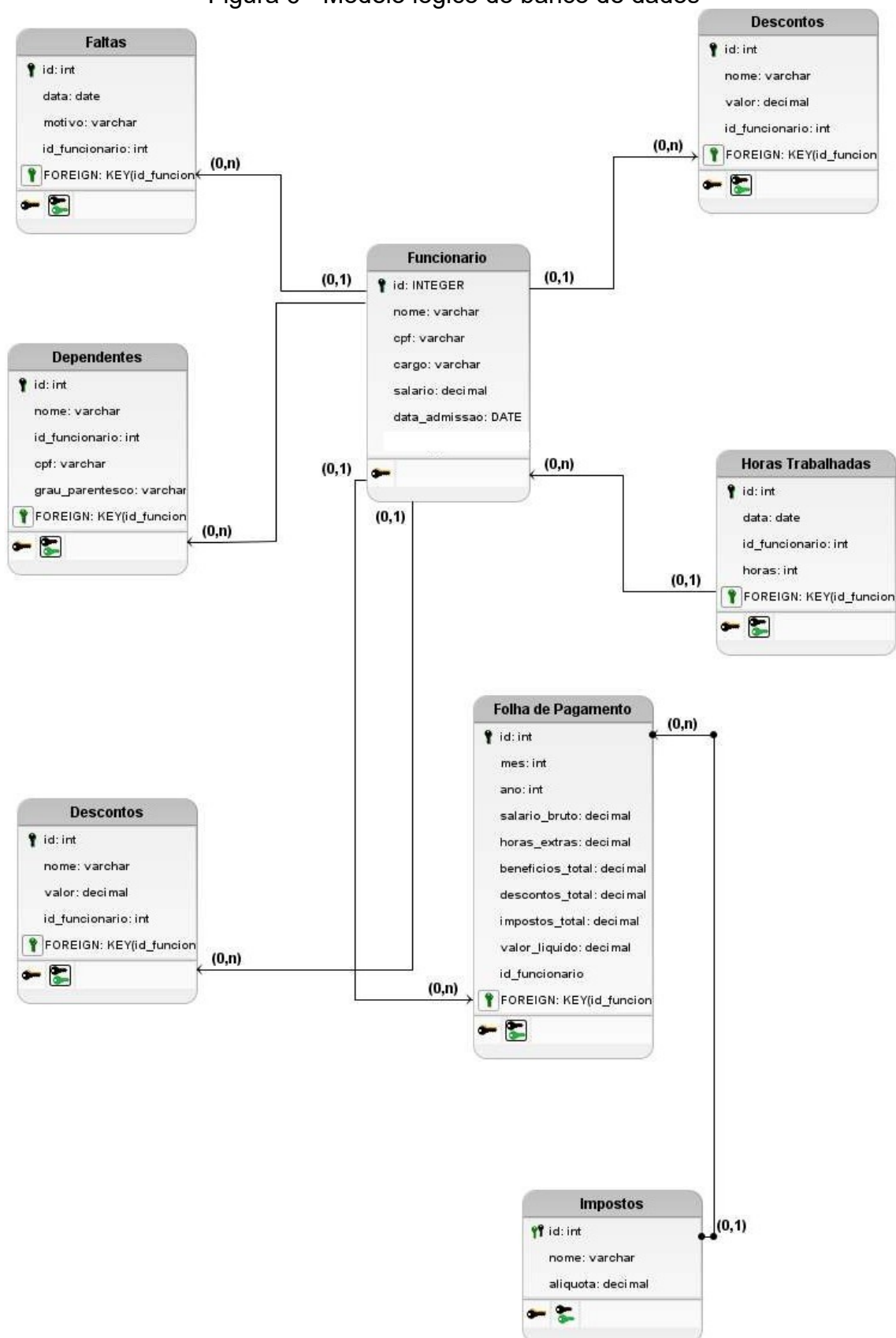
5.1 DIAGRAMA ENTIDADE RELACIONAMENTO

Figura 5 - Diagrama entidade relacionamento do banco de dados



5.2 MODELO LÓGICO

Figura 6 - Modelo lógico do banco de dados



5.3 SCRIPTS

5.3.1 Criação do banco

O primeiro passo no processo é criar o banco (definindo seu nome) e especificar o uso dele nas próximas operações:

```
Create database Pim3  
use Pim3
```

5.3.2 Criação da tabela para funcionários

```
create table tbFUNCIONARIO  
(  
    id int identity,  
    nome varchar(50)not null ,  
    cpf varchar (11)not null,  
    cargo varchar(20)not null,  
    salario decimal(9,2) ,  
    data_admissao date not null ,  
    primary key(id)  
)
```

5.3.3 Criação da tabela para dependentes

```
create table tbDEPENDENTE  
(  
    id int identity primary key,  
    nome varchar(50) not null,  
    id_funcionario int not null,  
    cpf varchar (11) not null,  
    grau_parentesco varchar (10) not null,  
    FOREIGN KEY(id_funcionario) REFERENCES tbFuncionario(id) )
```


5.3.4 Criação da tabela para horas trabalhadas

```
create table tbHORAISTRABALHADAS
(
    id int identity primary key,
    data date not null,
    id_funcionario int not null,
    horas int not null,
    FOREIGN KEY(id_funcionario) REFERENCES tbFuncionario(id)
)
```

5.3.5 Criação da tabela para faltas

```
create table tbFALTA
(
    id int primary key,
    data date not null,
    motivo varchar (30),
    id_funcionario int not null,
    FOREIGN KEY(id_funcionario) REFERENCES tbFuncionario(id)
)
```

5.3.6 Criação de tabela para benefícios

```
create table tbBENEFICIO
(
    id int primary key,
    nome varchar(50)not null,
    valor decimal (9,3)not null,
    id_funcionario int not null,
    FOREIGN KEY(id_funcionario) REFERENCES tbFuncionario(id)
)
```

5.3.7 Criação de tabela para descontos

```
create table tbDESCONTO
(
    id int primary key,
    nome varchar(50)not null,
    valor decimal (9,2)not null,
    id_funcionario int not null,
    FOREIGN KEY(id_funcionario) REFERENCES tbFuncionario(id)
)
```

5.3.8 Criação de tabela para impostos

```
create table tbIMPOSTO
(
    id int primary key,
    nome varchar(50) not null,
    aliquota decimal (9,2) not null
)
```

5.3.9 Criação de tabela para folha de pagamento

```
create table tbFOLHAPAGAMENTO
(
    id int primary key,
    mes int not null,
    ano int not null,
    salario_bruto decimal ,
    horas_extras decimal ,
    beneficios_total decimal ,
    descontos_total decimal ,
    impostos_total decimal ,
    valor_liquido decimal ,
    id_funcionario int not null,
    FOREIGN KEY(id_funcionario) REFERENCES tbFuncionario(id)
)
```

REFERÊNCIAS BIBLIOGRÁFICAS

1. Sistema de Folha de Pagamento: saiba tudo sobre o melhor aliado do RH. metadados.com.br, 2022. Disponível em: <<https://www.metadados.com.br/blog/sistema-de-folha-de-pagamento>>. Acesso em: 26 de maio de 2023.
2. UML – Basic Notations. Disponível em: <https://www.tutorialspoint.com/uml/uml_basic_notations.htm>. Acesso em: 26 de maio de 2023.
3. Ricaldi, Tiago. Os 9 princípios da Modelagem de Processos de Negócio. neomind.com.br, 2020. Disponível em: <file:///C:/Users/rafael.antoniassi/Desktop/university/PIM_2/PIM_2.pdf>. Acesso em: 26 de maio de 2023.
4. Diagrama de caso de uso UML: O que é, como fazer e exemplos. Disponível em: <<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>>. Acesso em: 26 de maio de 2023.