**UNIVERSIDADE PAULISTA**

**UNIDADE PAULISTA**

**Análise e Desenvolvimento de Sistemas**

ALINE COUTINHO TRINDADE

BIANCA SOUZA SANTOS

GUILHERME CARVALHO

GUILHERME DA COSTA SILVA

IGOR SOARES

MARIAH EDUARDA SILVA

**NORTEN**

**São Paulo, SP**

**2023**

ALINE COUTINHO TRINDADE

BIANCA SOUZA SANTOS

GUILHERME CARVALHO

GUILHERME DA COSTA SILVA

IGOR SOARES

MARIAH EDUARDA SILVA

**NORTEN**

Projeto Integrado Multidisciplinar apresentado à Universidade Paulista – UNIP, como parte dos requisitos necessários à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Professor.Tarcisio

**São Paulo, SP**

**2023**

ALINE COUTINHO TRINDADE

BIANCA SOUZA SANTOS

GUILHERME CARVALHO

GUILHERME DA COSTA SILVA

IGOR SOARES

MARIAH EDUARDA SILVA

**NORTEN**

Aprovada em \_\_\_/\_\_\_/\_\_\_\_

BANCA EXAMINADORA

Prof. Tarcisio

Titulação-Instituição

**RESUMO**

Este projeto tem como objetivo principal o desenvolvimento de um software abrangente para gerenciar o cadastro de funcionários, com foco na visualização e armazenamento de dados. A solução será fornecida através de uma interconexão de aplicativos desktop, web e mobile. As principais funcionalidades incluem autenticação de usuários (login e recuperação de senha), registro de funcionários, consulta de folhas de pagamento, consulta de férias e geração de folhas de pagamento.

O embasamento teórico abrangeu temas cruciais, tais como regulamentações sobre acesso a dados, configuração de redes, gestão financeira, sistemas automatizados, administração de recursos humanos, e implementação de bancos de dados e integração entre os sistemas. As linguagens de programação utilizadas para desenvolver o software incluem C#, Java, e tecnologias Android para a aplicação móvel.

Esse projeto visa melhorar a eficiência e a precisão do processo de folha de pagamento, bem como facilitar o gerenciamento de recursos humanos em uma variedade de plataformas, proporcionando maior segurança e confiabilidade no armazenamento e acesso aos dados dos funcionários.

**Palavras Chaves** – Gestão de Dados, Sistemas, Segurança, Folha de Pagamento, Implementação.

**ABSTRACT**

This project aims primarily at the development of a comprehensive software for managing employee records, with a focus on data visualization and storage. The solution will be provided through an interconnection of desktop, web, and mobile applications. Key functionalities include user authentication (login and password recovery), employee registration, payroll inquiry, vacation inquiry, and payroll generation.

The theoretical framework covered crucial topics such as regulations on data access, network configuration, financial management, automated systems, human resources administration, and implementation of databases and integration between systems. The programming languages used to develop the software include C#, Java, and Android technologies for the mobile application.

This project seeks to improve the efficiency and accuracy of the payroll process, as well as facilitate human resources management across various platforms, providing increased security and reliability in the storage and access to employee data.

**Keywords** – Data Management, Systems, Security, Payroll, Implementation.

Sumário

[1. INTRODUÇÃO 1](#_Toc152027413)

[2. Programação Orientada a Objetos 2](#_Toc152027414)

[2.1 Desenvolvimento Desktop 2](#_Toc152027415)

[2.1.1 Tela Login 2](#_Toc152027416)

[2.1.2 Tela Menu 3](#_Toc152027417)

[2.1.3 Tela Cadastro Funcionário 4](#_Toc152027418)

[2.1.4 Tela Consulta Folha de Pagamento 5](#_Toc152027419)

[2.1.5 Tela Cadastro de Folha de Pagamento 6](#_Toc152027420)

[2.1.6 Tela Consultar Funcionário 6](#_Toc152027421)

[2.1.7 Tela Consultar Férias 7](#_Toc152027422)

[2.2 Desenvolvimento WEB 8](#_Toc152027423)

[2.2.1 Tela Login 8](#_Toc152027424)

[2.2.2 Tela de Menu 9](#_Toc152027425)

[2.2.3 Tela Minhas Informações 10](#_Toc152027426)

[2.2.4 Tela Folha de Pagamento 11](#_Toc152027427)

[2.2.5 Tela Histórico Salarial 11](#_Toc152027428)

[2.3 Desenvolvimento Mobile 12](#_Toc152027429)

[2.3.1 Tela Login 12](#_Toc152027430)

[2.3.2 Tela Menu 13](#_Toc152027431)

[2.3.3 Tela Folha de Pagamento 14](#_Toc152027432)

[3. Banco de Dados 16](#_Toc152027433)

[3.1 Modelo Lógico 16](#_Toc152027434)

[3.2 Entidade – Relacionamento Banco de Dados 19](#_Toc152027435)

[4. Requisitos 20](#_Toc152027436)

[4.1 Requisitos Funcionais 20](#_Toc152027437)

[4.1.1 Tela de Login 20](#_Toc152027438)

[4.1.2 Cadastrar Funcionários 20](#_Toc152027439)

[4.1.3 Cadastro de Funcionários de RH 21](#_Toc152027440)

[4.1.4 Consultar Funcionários 21](#_Toc152027441)

[4.1.5 Cadastro e Consulta de Folha de Pagamento 21](#_Toc152027442)

[4.1.6 Consultar Férias 21](#_Toc152027443)

[4.1.7 Tela Menu 22](#_Toc152027444)

[4.2 Requisitos de Desempenho 23](#_Toc152027445)

[4.3 Ciclo de vida do software 24](#_Toc152027446)

[4.4 Regras de Negócio 25](#_Toc152027447)

[4.4.1 Benefícios e Descontos 25](#_Toc152027448)

[5. Projeto Interface com o Usuário 27](#_Toc152027449)

[6. Diagramas e Análise do Sistema 29](#_Toc152027450)

[6.1 Diagrama Caso de Uso 30](#_Toc152027451)

[6.2 Diagrama de Classe 31](#_Toc152027452)

[6.3 Diagrama de Atividade 32](#_Toc152027453)

[7. Caso de Uso e Planilha de Teste 33](#_Toc152027454)

[7.1 Casos de sistemas WEB e Mobile 33](#_Toc152027455)

[7.2 Casos do sistema Destkop 35](#_Toc152027456)

[7.3 Planilha de Teste 38](#_Toc152027457)

[8. LGPD 40](#_Toc152027458)

[8.1 Princípios Fundamentais da LGPD Aplicados ao Projeto 40](#_Toc152027459)

[Transparência e Consentimento 40](#_Toc152027460)

[Finalidade Específica e Necessidade 40](#_Toc152027461)

[Segurança e Prevenção 40](#_Toc152027462)

[Consentimento Informado 41](#_Toc152027463)

[Acesso e Retificação de Dados 41](#_Toc152027464)

[Interfaces Web, Desktop e Mobile 41](#_Toc152027465)

[Treinamento e Conscientização 41](#_Toc152027466)

[9. Códigos e Sintaxe 43](#_Toc152027467)

[9.1 Sintaxe Desktop 43](#_Toc152027468)

[9.1.1 Cadastro de Folha de Pagamento 43](#_Toc152027469)

[9.1.2 Cadastro de Funcionários 49](#_Toc152027470)

[9.1.3 Consultar Férias 55](#_Toc152027471)

[9.1.4 Consultar Funcionários 59](#_Toc152027472)

[9.1.5 Consulta Folha de Pagamento 63](#_Toc152027473)

[9.1.6 Tela de Login 69](#_Toc152027474)

[9.1.7 Tela Menu 72](#_Toc152027475)

[9.2 Scripts SQL 75](#_Toc152027476)

[9.3 Sintaxe Web 78](#_Toc152027477)

[9.3.1 Folha de Pagamento 78](#_Toc152027478)

[9.3.2 Login 83](#_Toc152027479)

[9.4 Sintaxe Mobile 87](#_Toc152027480)

[10. Considerações Finais 92](#_Toc152027481)

[REFERÊNCIAS 94](#_Toc152027482)

**Lista de Figuras**

[Figura 1 - Tela de Login 2](#_Toc152027483)

[Figura 2 - Tela Menu 3](#_Toc152027484)

[Figura 3 - Tela de Cadastro de Funcionário 4](#_Toc152027485)

[Figura 4 - Folha de Pagamento 5](#_Toc152027486)

[Figura 5 - Cadastro de Folha de Pagamento 6](#_Toc152027487)

[Figura 6 - Consultar Funcionário 6](#_Toc152027488)

[Figura 7 - Consultar férias 7](#_Toc152027489)

[Figura 8 - Tela Login Web 8](#_Toc152027490)

[Figura 9 - Tela de Menu Web 9](#_Toc152027491)

[Figura 10 - Informações Web 10](#_Toc152027492)

[Figura 11 - Tela Folha de Pagamento Web 11](#_Toc152027493)

[Figura 12 - Histórico Salaria Web 11](#_Toc152027494)

[Figura 13 - Tela de Login Mobile 12](#_Toc152027495)

[Figura 14 - Tela de Menu Mobile 13](#_Toc152027496)

[Figura 15 - Tela Folha de Pagamento Mobile 14](#_Toc152027497)

[Figura 16 - Tela Minhas Informações Mobile 15](#_Toc152027498)

[Figura 17 - Ciclo de Software - Internet 24](#_Toc152027499)

[Figura 18 - Diagrama Caso de Uso 30](#_Toc152027500)

[Figura 19 - Diagrama de Classe Norten 31](#_Toc152027501)

[Figura 20 - Diagrama de Atividade Norten 32](#_Toc152027502)

# INTRODUÇÃO

O desenvolvimento de softwares é um processo que envolve a exploração de ideias e caminhos para a criação de produtos finais, estimulando diálogos e novas maneiras de gerenciar a comunicação entre equipes e clientes. Esse processo busca concretizar conceitos abstratos em soluções concretas e normalizadas, que podem ser refinadas ao longo do tempo. Como dito por Jorge Henrique (2003) “[...], além de ser uma entidade de natureza mecânica, é uma entidade descritiva, complexamente hierarquizada, cognitivo-linguística e histórica, concebida através de esforços coletivos durante um considerável período de tempo.”

No cenário atual, o mercado de trabalho e as corporações estão cada vez mais recorrendo à tecnologia para simplificar, agilizar e tornar mais práticas as tarefas relacionadas à gestão de dados de seus funcionários. Exemplos disso incluem a automação e a migração para ambientes em nuvem de fichas cadastrais e folhas de pagamento. Essas abordagens representam uma das maneiras mais eficientes de gerenciar a segurança, filtrar informações e desenvolver novos métodos e processos para a produção e pesquisa de mercado.

O armazenamento de dados desempenha um papel fundamental, permitindo que os dados cumpram suas funções, como a identificação de indivíduos e a manipulação de informações.

Este projeto apresenta um sistema que atua como um portal de acesso às informações de Recursos Humanos (RH). O sistema inclui versões Web e Mobile, que permitem aos funcionários acessar suas informações, verificar suas folhas de pagamento e histórico salarial, além de registrar o ponto de entrada e saída no horário de trabalho. O desenvolvimento deste sistema visa aprimorar a eficiência e a acessibilidade das informações de RH em um mundo cada vez mais digital e móvel.

# 2. Programação Orientada a Objetos

O desenvolvimento de um sistema consiste em sua sintaxe e lógica, neste projeto foi utilizado a linguagem C#, Java, Android e SQL. Através do planejamento de visuais e lógicas propostas, foram separados os protótipos e estes foram segmentados desta forma:

## 2.1 Desenvolvimento Desktop

O sistema contém 4 telas principais sendo elas com as funções de Login, Cadastro, Listar Cadastros, Folha de Pagamento.

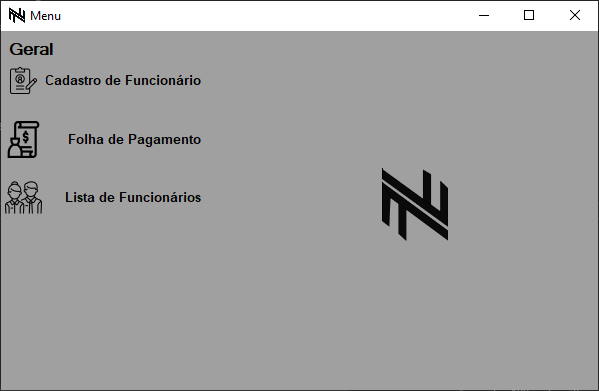
### 2.1.1 Tela Login

Figura 1 - Tela de Login



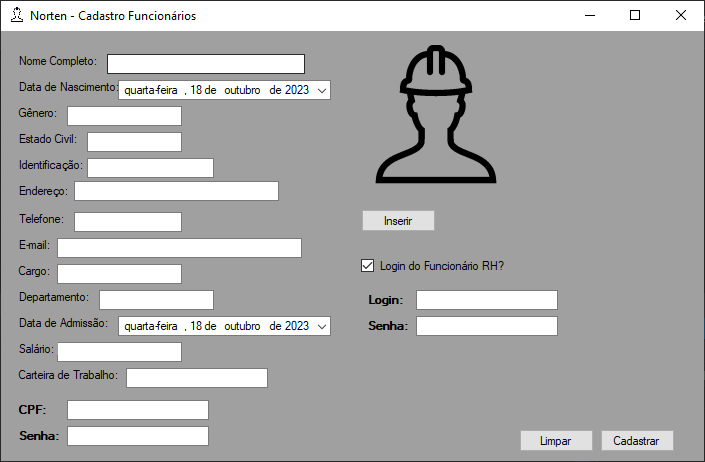
### 2.1.2 Tela Menu

Figura 2 - Tela Menu



### 2.1.3 Tela Cadastro Funcionário

Figura 3 - Tela de Cadastro de Funcionário

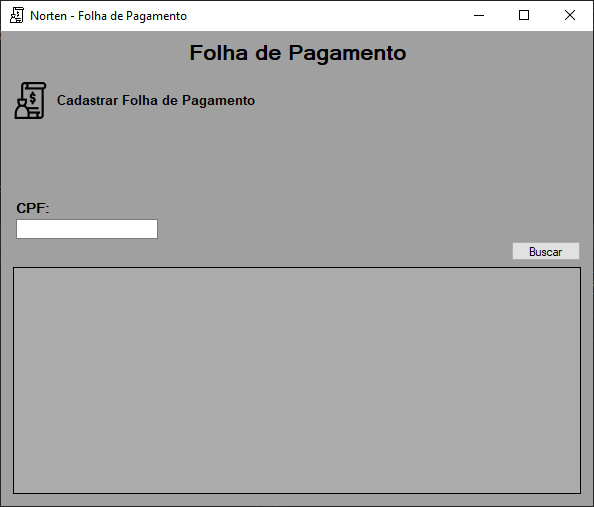


A tela de cadastro funcional apresentada na imagem contém os seguintes campos:

* ***Nome Completo:*** Este campo é usado para coletar o nome completo do funcionário.
* ***Data de Nascimento:*** Este campo é usado para coletar a data de nascimento do funcionário.
* ***Gênero:*** Este campo é usado para coletar o gênero do funcionário.
* ***Estado Civil:*** Este campo é usado para coletar o estado civil do funcionário.
* ***Identificação:*** Este campo é usado para coletar o número de identificação do funcionário.
* ***Endereço***: Este campo é usado para coletar o endereço do funcionário.
* ***Telefone:*** Este campo é usado para coletar o telefone do funcionário.
* ***Login do Funcionário RH?:*** Este campo é usado para indicar se o funcionário é um funcionário do departamento de Recursos Humanos.
* ***E-mail:*** Este campo é usado para coletar o endereço de e-mail do funcionário.
* ***Cargo:*** Este campo é usado para coletar o cargo do funcionário.
* ***Departamento:*** Este campo é usado para coletar o departamento do funcionário.
* ***Data de Admissão:*** Este campo é usado para coletar a data de admissão do funcionário.
* ***Salário:*** Este campo é usado para coletar o salário do funcionário.
* ***Carteira de Trabalho:*** Este campo é usado para coletar o número da carteira de trabalho do funcionário.
* ***CPF:*** Este campo é usado para coletar o CPF do funcionário.
* ***Senha:*** Este campo é usado para coletar a senha do funcionário.
* ***Foto:*** Este campo é usado para coletar os dados da foto em gravá-la em dados no sistema.

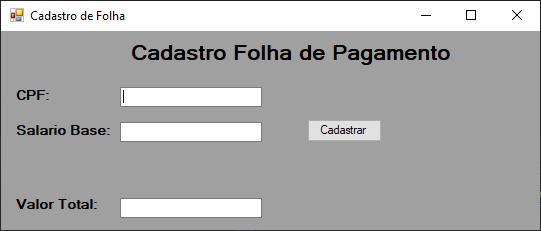
### 2.1.4 Tela Consulta Folha de Pagamento

Figura 4 - Folha de Pagamento



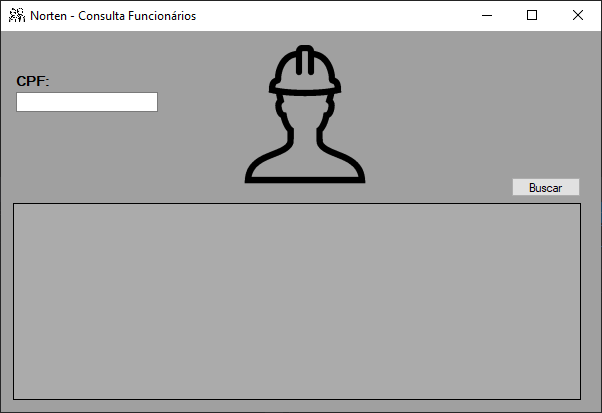
### 2.1.5 Tela Cadastro de Folha de Pagamento

Figura 5 - Cadastro de Folha de Pagamento



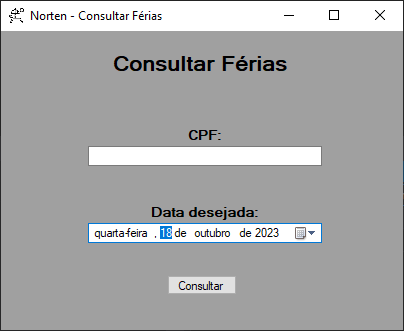
### 2.1.6 Tela Consultar Funcionário

Figura 6 - Consultar Funcionário



### 2.1.7 Tela Consultar Férias

Figura 7 - Consultar férias

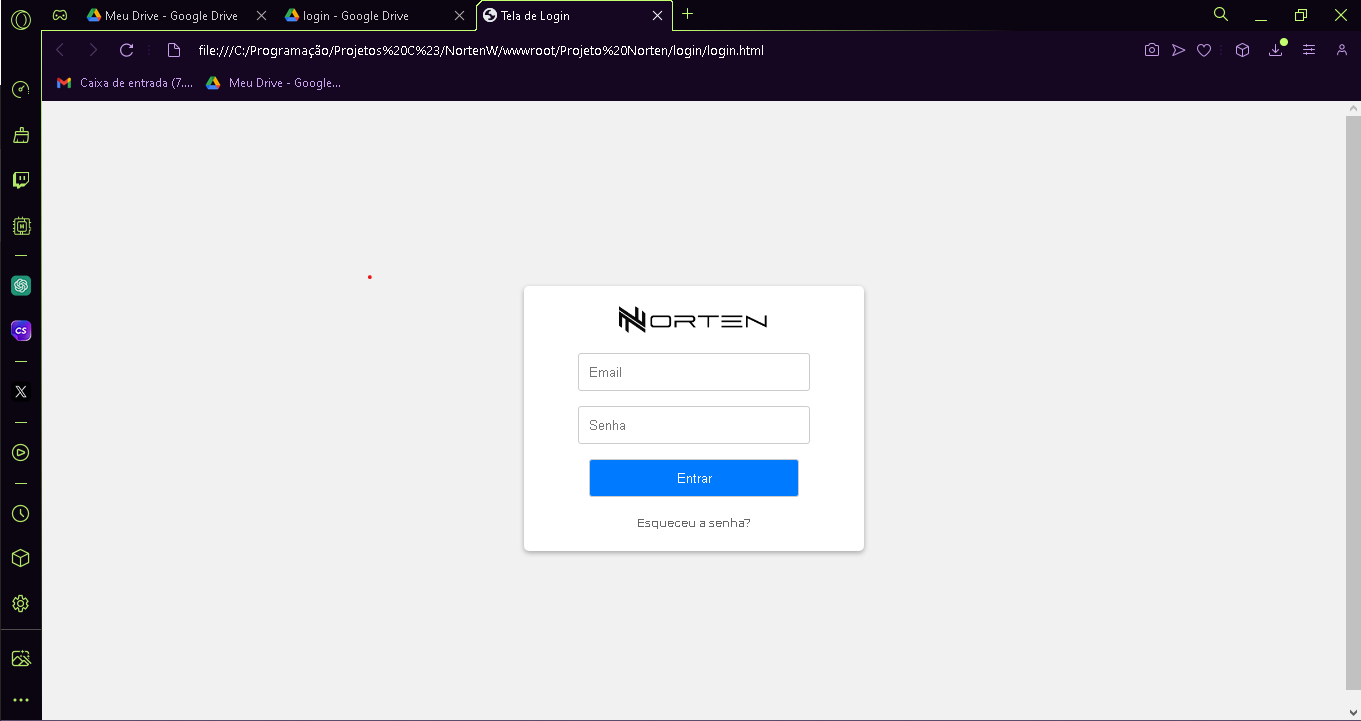


## 2.2 Desenvolvimento WEB

A aplicação via web, consiste na ideia e planejamento de ser uma das interfaces a qual o ***colaborador*** irá visualizar suas informações, sendo capaz de visualizar suas informações, sua folha de pagamento e seu histórico salarial.

### 2.2.1 Tela Login

Figura 8 - Tela Login Web



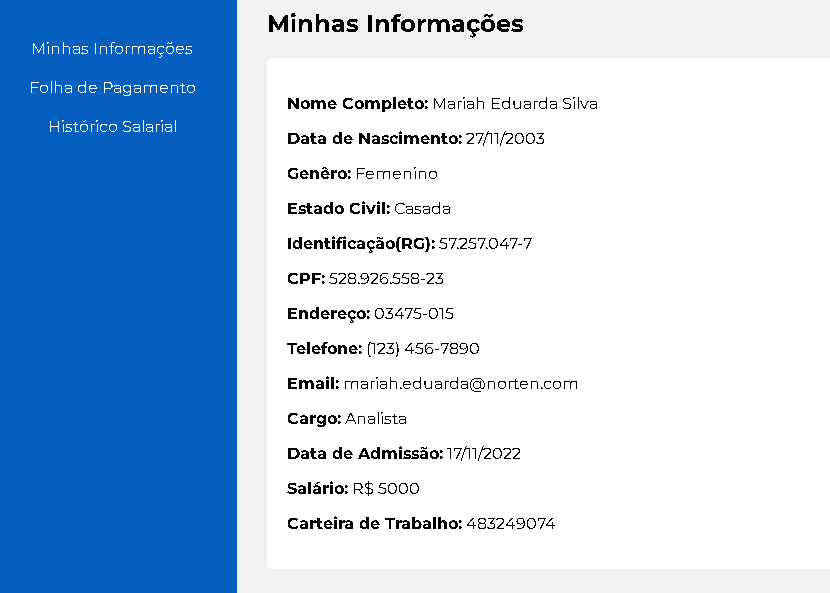
### 2.2.2 Tela de Menu

Figura 9 - Tela de Menu Web



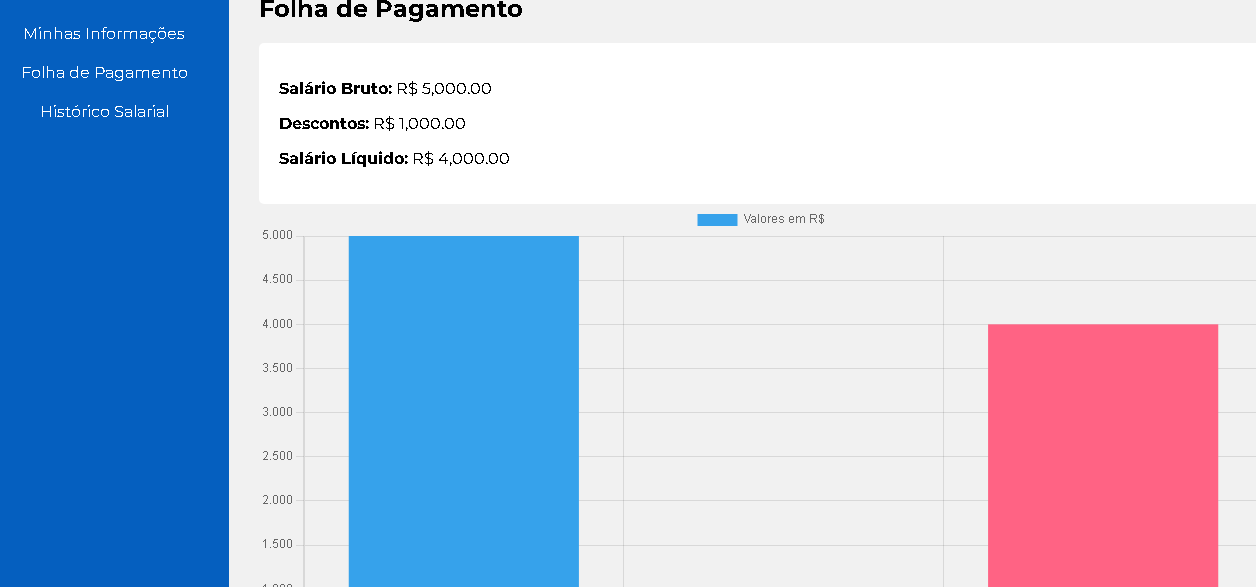
### 2.2.3 Tela Minhas Informações

Figura 10 - Informações Web



### 2.2.4 Tela Folha de Pagamento

Figura 11 - Tela Folha de Pagamento Web



### 2.2.5 Tela Histórico Salarial

Figura 12 - Histórico Salaria Web

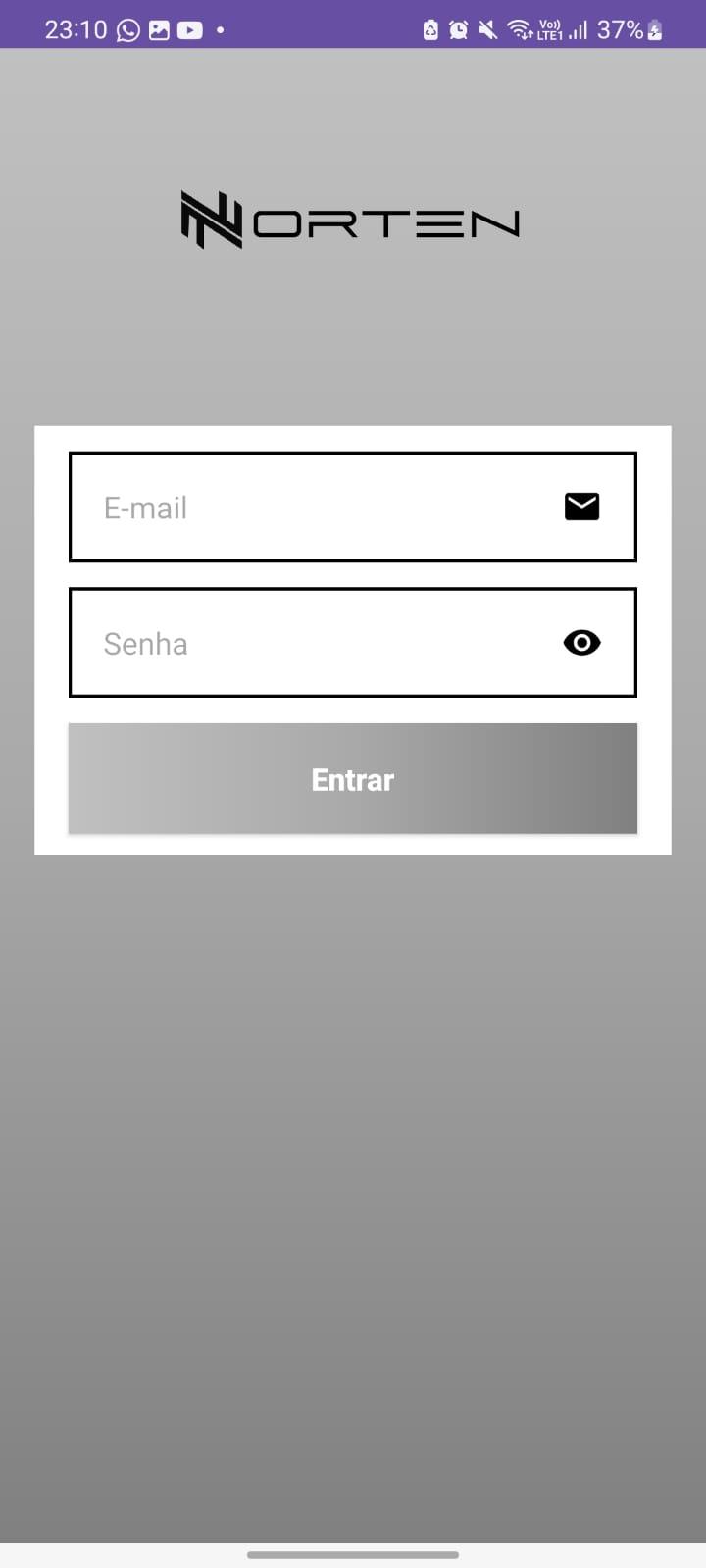


## 2.3 Desenvolvimento Mobile

A aplicação via mobile (smartphone), consiste na ideia e planejamento de ser uma das interfaces a qual o funcionário irá visualizar suas informações, sendo capaz de visualizar suas informações, sua folha de pagamento e seu histórico salarial.

### 2.3.1 Tela Login

Figura 13 - Tela de Login Mobile



### 2.3.2 Tela Menu

Figura 14 - Tela de Menu Mobile



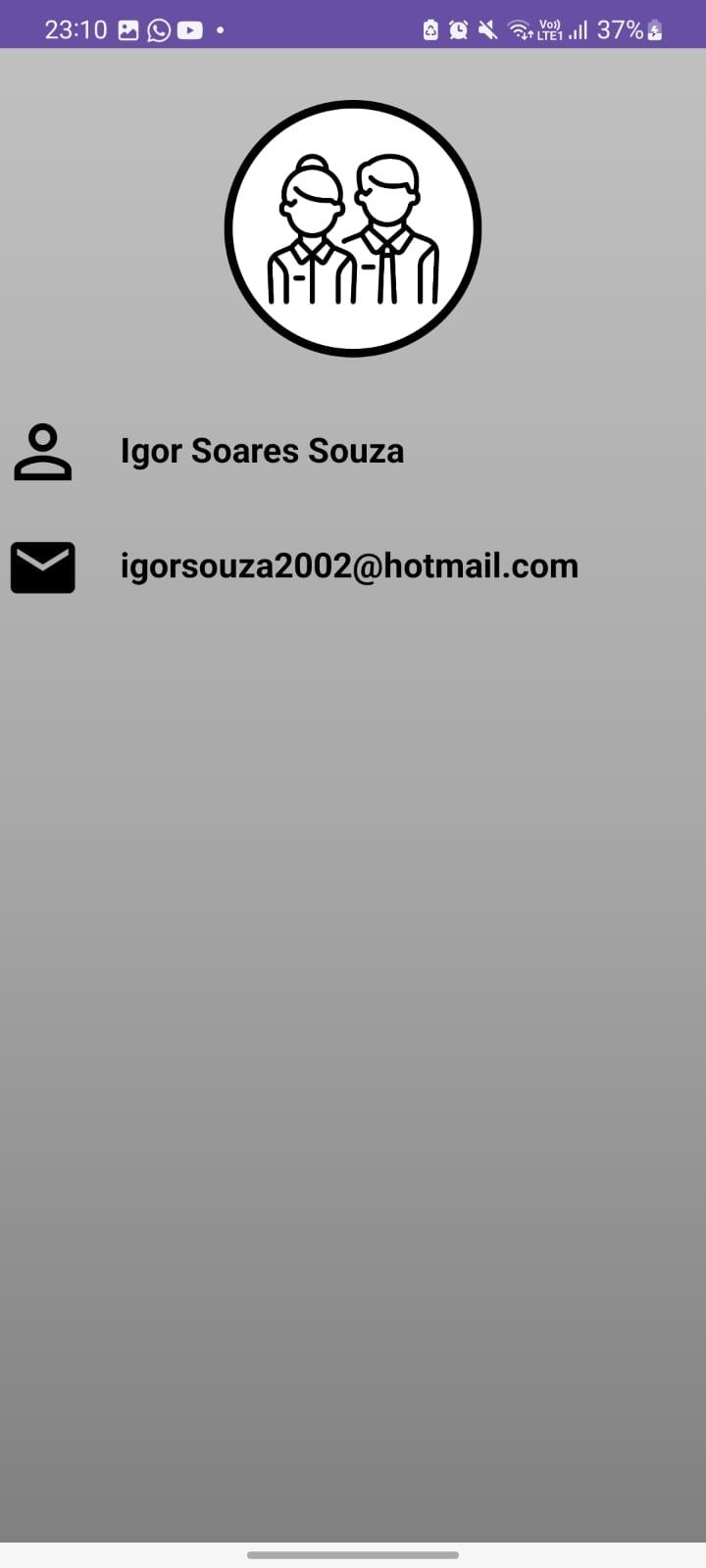
### 2.3.3 Tela Folha de Pagamento

Figura 15 - Tela Folha de Pagamento Mobile



* + 1. **Tela Minhas Informações**

Figura 16 - Tela Minhas Informações Mobile

****

# 3. Banco de Dados

Segundo DATE (2004, p. 10), “Um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa”.

A metodologia utilizada neste projeto, consistiu em utilizarmos um modelo lógico que atendia a estrutura do sistema, uma base de dados capaz de armazenar informações de funcionários, informações de login e registrasse folhas de pagamentos e o histórico relacionado a elas.

## 3.1 Modelo Lógico

O modelo lógico do banco de dados "Norten" é representado pelas seguintes tabelas e seus relacionamentos:

**Tabela Func\_Usuario\_RH:**

**Colunas:**

**idLogin** (Chave Primária, Inteiro, Auto-incremento)

**RH\_Login** (Varchar, Não nulo)

**RH\_senha** (Varchar, Não nulo)

**Tabela Funcionarios:**

**Colunas:**

**ID\_Func** (Chave Primária, Inteiro, Auto-incremento)

**Func\_CPF** (Char, não nulo, Único)

**Func\_NomeCompleto** (NVarchar)

**Func\_DataNascimento** (Data)

**Func\_Genero** (NVarchar)

**Func\_EstadoCivil** (NVarchar)

**Func\_Identificacao** (NVarchar)

**Func\_Endereco** (NVarchar)

**Func\_TelefoneCelular** (NVarchar)

**Func\_Email** (NVarchar)

**Func\_Cargo** (NVarchar)

**Func\_Departamento** (NVarchar)

**Func\_DataAdmissao** (Data)

**Func\_Salario** (Decimal)

**Func\_CarteiraTrabalho** (NVarchar)

**Func\_Foto** (Varbinary)

**Func\_Senha** (Varchar, Não nulo)

**idLogin** (Chave Estrangeira referenciando Func\_Usuario\_RH.idLogin)

**Tabela FolhaDePagamento**:

**Colunas**:

**ID** (Chave Primária, Inteiro, Auto-incremento)

**Conta** (NVarchar)

**SalarioBase** (Decimal)

**AdicionalNoturno** (Calculada a partir de SalarioBase)

**AdiantamentoQuinzenal** (Decimal, Padrão 0.00)

**PlanoDeSaude** (Decimal, Padrão 50.00)

**INSS** (Decimal, Padrão 0.00)

**ValeRefeicao** (Decimal, Padrão 1200.00)

**ValeTransporte** (Decimal, Padrão 1000.00)

**FGTS** (Calculada a partir de SalarioBase)

**DecimoTerceiro** (Decimal, Padrão 0.00)

**Ferias** (Decimal, Padrão 0.00)

**ID\_Func** (Chave Estrangeira referenciando Funcionarios.ID\_Func)

**SalarioBase** (Chave Estrangeira referenciando Funcionarios.Func\_Salario)

**Relacionamentos**:

**Func\_Usuario\_RH.idLogin** é chave primária, utilizada como chave estrangeira Funcionarios.idLogin.

**FolhaDePagamento.ID\_Func** é uma chave estrangeira referenciando Funcionarios.ID\_Func.

**FolhaDePagamento.SalarioBase** é uma chave estrangeira referenciando Funcionarios.Func\_Salario.

## 3.2 Entidade – Relacionamento Banco de Dados

Uma tabela ER de banco de dados é um diagrama que representa os dados e relacionamentos de um banco de dados. É uma ferramenta importante para o design de bancos de dados, pois ajuda a visualizar e entender como os dados estão organizados.

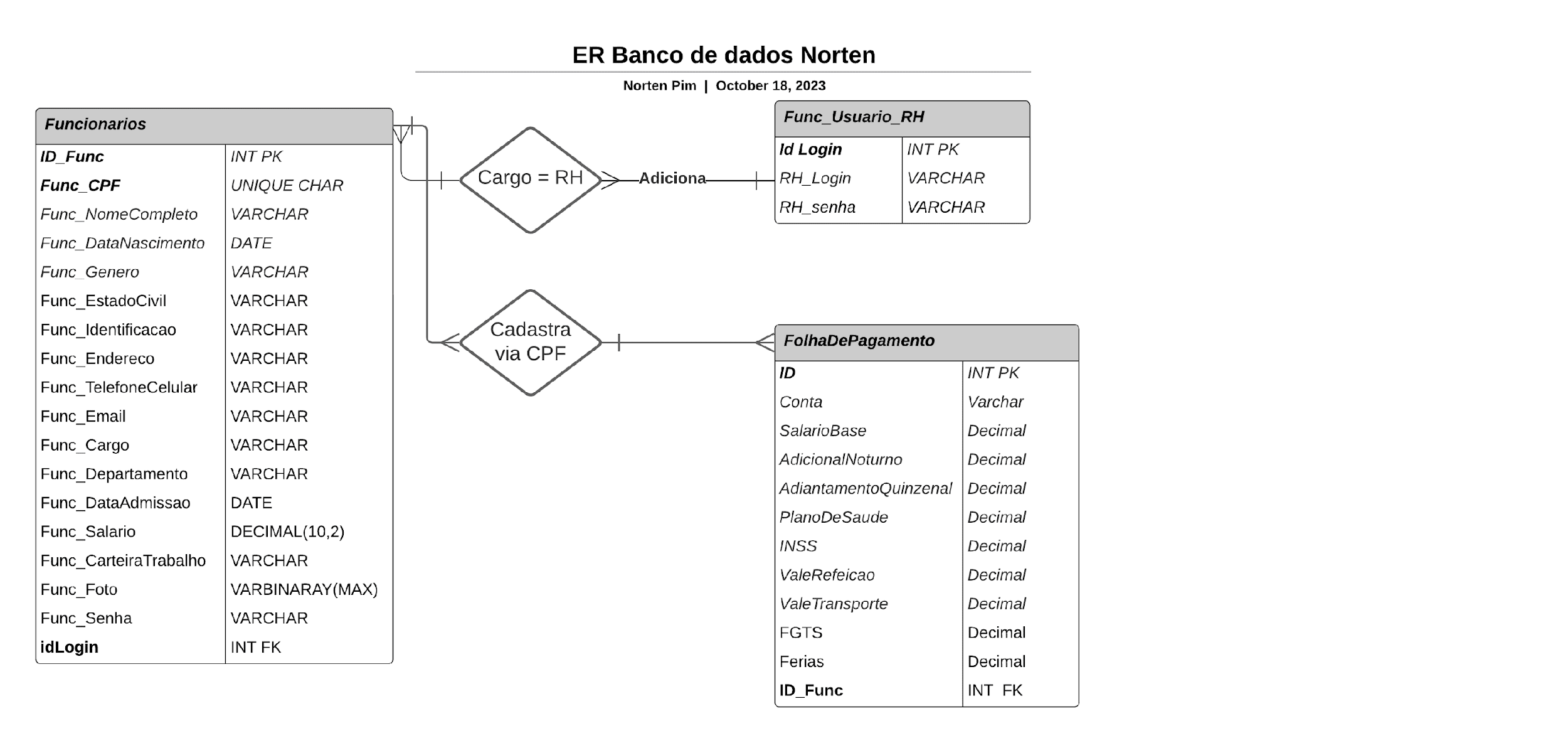
Uma tabela ER é composta por dois componentes principais:

Entidades: representam os objetos do mundo real que serão armazenados no banco de dados. Por exemplo, uma entidade pode representar um funcionário, um produto ou uma venda.

Relacionamentos: representam as relações entre as entidades. Por exemplo, um relacionamento pode representar a relação entre um funcionário e um produto, ou entre uma venda e um produto.

Os relacionamentos são representados por linhas que conectam as entidades. O tipo de relacionamento pode ser representado por um símbolo, como uma linha sólida para um relacionamento 1 para 1, uma linha com uma seta para um relacionamento 1 para muitos, ou uma linha com duas setas para um relacionamento muitos para muitos.

**Figura 11 -** ER Banco de dados Norten



# 4. Requisitos

A engenharia de software em função ao projeto planeja automatizar seus processos, aprofundar técnicas e práticas de desenvolvimento de softwares, focando áreas de lógica e algoritmos, acompanhando cada produção de um sistema e analisando seu ciclo de vida e requisitos do sistema.

## 4.1 Requisitos Funcionais

Em engenharia de software, um requisito funcional define uma função de um sistema de software ou seu componente. O requisito funcional representa o que o software faz, em termos de tarefas e serviços. Uma função é descrita como um conjunto de entradas, seu comportamento e as saídas.

### 4.1.1 Tela de Login

O sistema deve iniciar em uma Tela de Login com prévio usuário administrativo já previamente adicionado ao banco de dados sendo este o usuário: Gui e a senha: 123.

### 4.1.2 Cadastrar Funcionários

O sistema deverá cadastrar os dados de registro dos Funcionários e separadamente um usuário e senha para acesso do sistema web e mobile.

### 4.1.3 Cadastro de Funcionários de RH

O sistema deverá cadastrar os dados de registros de um funcionário de RH junto de seu acesso de login e senha criado em um painel no sistema de registro.

### 4.1.4 Consultar Funcionários

O sistema deverá mostrar na tela através da opção *CONSULTAR FUNCIONARIOS*. As informações serão mostradas de acordo com o CPF do funcionário.

### 4.1.5 Cadastro e Consulta de Folha de Pagamento

O sistema deverá gerar um registro no banco com as informações do holerite do funcionário selecionado e mostrar na tela a informação.

O sistema deverá ter dois acessos para essa folha de pagamento, a primeira tela para visualização e consulta e a segunda para cadastrar o CPF do Funcionário e seu salário atual.

### 4.1.6 Consultar Férias

O sistema deverá informar ao usuário se o funcionário poderá prosseguir ou não com as suas férias, sendo utilizados a entrada do CPF e da data solicitada pelo funcionário, dando uma margem de um ano desde sua data de admissão.

### 4.1.7 Tela Menu

O sistema deverá informar por meio de botões e imagens as telas que serão utilizadas as ações dele.

## 4.2 Requisitos de Desempenho

Um requisito de desempenho impõe condições aos requisitos funcionais. Por exemplo:

* Velocidade
* Eficiência
* Precisão
* Produtividade

**Sistema designado**

O sistema deverá ser implementado em sistemas Windows, Android, Web e no banco de dados SQL Server.

**Limite de Tempo de Resposta**

O sistema deve responder suas buscas de listagem com o limite de até 20 segundos.

## 4.3 Ciclo de vida do software

O ciclo de vida de um projeto é a divisão da Gestão do Projeto em fases pelas quais ele deve passar do início ao término. A cada período que corresponda a uma fase, o projeto pode sofrer incrementos e alterações significativas que ditarão o ritmo das atividades que devem ser desenvolvidas. Assim como descrito por José Macoratti (Internet) “A principal função do ciclo de vida do desenvolvimento de software é indicar as fases, atividades, entregas e responsabilidades de cada envolvido no processo de desenvolvimento de software.”

E tudo resultara da complexidade e do escopo do projeto, cada fase pode ser dividida em subunidades, de modo a conferir uma melhor organização ao trabalho desenvolvido.

Figura 17 - Ciclo de Software - Internet



Fonte: MACORRATI

## 4.4 Regras de Negócio

No mundo complexo do desenvolvimento de software, as "regras de negócio" desempenham um papel crucial. Elas são os princípios orientadores que definem as operações e comportamentos de um sistema de software específico. Em termos simples, são as diretrizes que determinam como um negócio deve operar e interagir com seus dados, processos e usuários por meio do software.

As regras de negócio encapsulam políticas, procedimentos e limitações que moldam a maneira como um software funciona. Elas são derivadas dos requisitos do negócio e são essenciais para traduzir as necessidades dos clientes em funcionalidades tangíveis.

### 4.4.1 Benefícios e Descontos

Assim determinados pela Lei brasileira e apresentado por Marcelo Furtado(Internet), os benefícios e descontos da folha de pagamento serão estes:

Validação de Salário Base: O salário base não pode ser menor do que um valor mínimo especificado pela empresa.

Validação de Valores Mínimos para Adicionais e Benefícios: Os valores para adicional noturno, adiantamento quinzenal, plano de saúde, INSS, vale refeição e vale transporte não podem ser negativos.

Validação de FGTS: O valor do FGTS deve ser calculado com base em uma porcentagem do salário base.

#### 4.4.1.1 Regras de Cálculo

Cálculo do Salário Líquido: O salário líquido pode ser calculado subtraindo os descontos (INSS, plano de saúde, adiantamento quinzenal, etc.) do salário base e adicionando os benefícios (adicional noturno, vale refeição, vale transporte, etc.).

Cálculo do FGTS: O FGTS pode ser calculado como uma porcentagem do salário base.

Processamento de Pagamento: A folha de pagamento deve ser processada mensalmente para cada funcionário, levando em consideração as horas trabalhadas, os adicionais e os benefícios.

Notificação de Saldo Insuficiente para Deduções: Se o salário base não cobrir as deduções (INSS, plano de saúde, etc.), a empresa deve ser notificada para tomar medidas corretivas.

Derivação do Décimo Terceiro Salário e Férias: O décimo terceiro salário e as férias podem ser calculados com base no salário base do funcionário e no tempo de serviço.

Independente de hierarquia de cargos, os funcionários receberam a mesma quantia de Vale-Transporte, Vale-Refeição e Vale-Alimentação

# 5. Projeto Interface com o Usuário

O projeto de interface com o usuário envolve uma abordagem multidisciplinar, combinando conhecimentos de design visual, psicologia cognitiva, ergonomia e interação humano-computador. Ao longo deste trabalho, iremos explorar essas diferentes áreas de conhecimento e como elas se aplicam na prática do projeto de interface.

Com o crescente uso de dispositivos móveis, aplicativos e sistemas web, a importância do projeto de interface com o usuário torna-se cada vez mais relevante. Uma interface bem projetada não só melhora a experiência do usuário, mas também pode aumentar a eficiência e a produtividade do sistema. Além disso, a interface pode desempenhar um papel crucial na adoção e aceitação de um sistema pelos usuários.

As cores e o design têm um impacto significativo na experiência do usuário ao interagir com um aplicativo. Aqui estão algumas maneiras pelas quais as cores e o design podem influenciar os usuários:

* **Atração visual e identidade de marca:** As cores desempenham um papel importante na criação de uma primeira impressão e na atração visual dos usuários. O uso de cores atraentes e harmoniosas pode despertar o interesse e chamar a atenção do usuário para o aplicativo. Além disso, as cores também podem ser usadas para transmitir a identidade da marca, criando uma associação visual entre o aplicativo e a marca.
* **Organização e estruturação:** Um bom design utiliza cores de forma eficaz para organizar e estruturar as informações no aplicativo. As cores podem ser usadas para diferenciar seções ou categorias diferentes, facilitando a navegação e a compreensão do conteúdo. Por exemplo, o uso de cores diferentes em botões ou ícones relacionados a ações específicas pode ajudar os usuários a identificar rapidamente as funcionalidades disponíveis.
* **Feedback visual:** As cores podem ser utilizadas para fornecer feedback visual aos usuários, indicando estados, status ou ações específicas. Por exemplo, ao pressionar um botão, a mudança de cor deste botão pode indicar que a ação foi reconhecida pelo aplicativo. Da mesma forma, o uso de cores diferentes para indicar erros ou sucesso em um processo pode ajudar os usuários a entenderem o resultado de suas ações de forma mais rápida e clara.
* **Legibilidade e acessibilidade:** O design de cores adequado é essencial para garantir a legibilidade do conteúdo e a acessibilidade para todos os usuários. O contraste entre o texto e o plano de fundo é fundamental para facilitar a leitura, especialmente para pessoas com deficiências visuais. Além disso, é importante considerar a escolha de cores que não causem fadiga visual e que sejam agradáveis ​​ao longo do tempo de uso do aplicativo.

# 6. Diagramas e Análise do Sistema

A análise de sistemas orientados a objetos é uma abordagem amplamente utilizada no desenvolvimento de software, que tem como base os conceitos da orientação a objetos. Essa metodologia oferece uma série de benefícios, como maior reutilização de código, modularidade, flexibilidade e facilidade de manutenção. Neste trabalho, iremos explorar a aplicação da análise de sistemas orientados a objetos em um cenário específico: o desenvolvimento de um sistema de folha de pagamento.

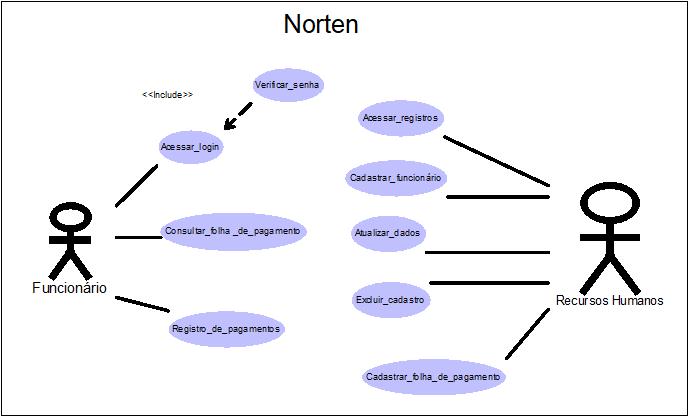
A folha de pagamento é uma função essencial em qualquer organização, responsável por calcular e registrar os salários dos funcionários, bem como as deduções legais e benefícios. O desenvolvimento de um sistema de folha de pagamento pode ser bastante complexo, envolvendo diversos cálculos e regras específicas de cada empresa.

Nesse contexto, a análise de sistemas orientados a objetos se torna uma abordagem valiosa, pois permite uma modelagem mais precisa e eficiente das entidades e regras de negócio envolvidas na folha de pagamento. Um dos principais artefatos utilizados nessa análise são os diagramas, que oferecem uma representação visual dos elementos do sistema e suas interações.

Os diagramas de análise de sistemas orientados a objetos permitem identificar as classes do sistema, seus atributos e métodos, assim como as relações entre elas. No caso específico da folha de pagamento, podemos representar, por exemplo, as classes Funcionário, Salário, Deduções, Benefícios, entre outras. Esses diagramas ajudam a compreender a estrutura do sistema e a definir a forma como os objetos se relacionam e interagem entre si.

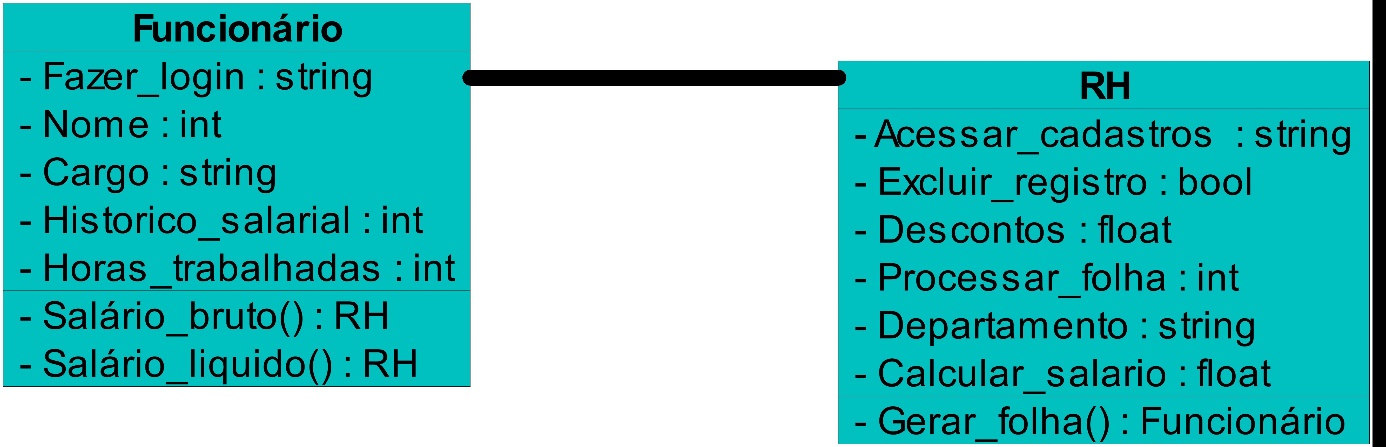
## 6.1 Diagrama Caso de Uso

Figura 18 - Diagrama Caso de Uso



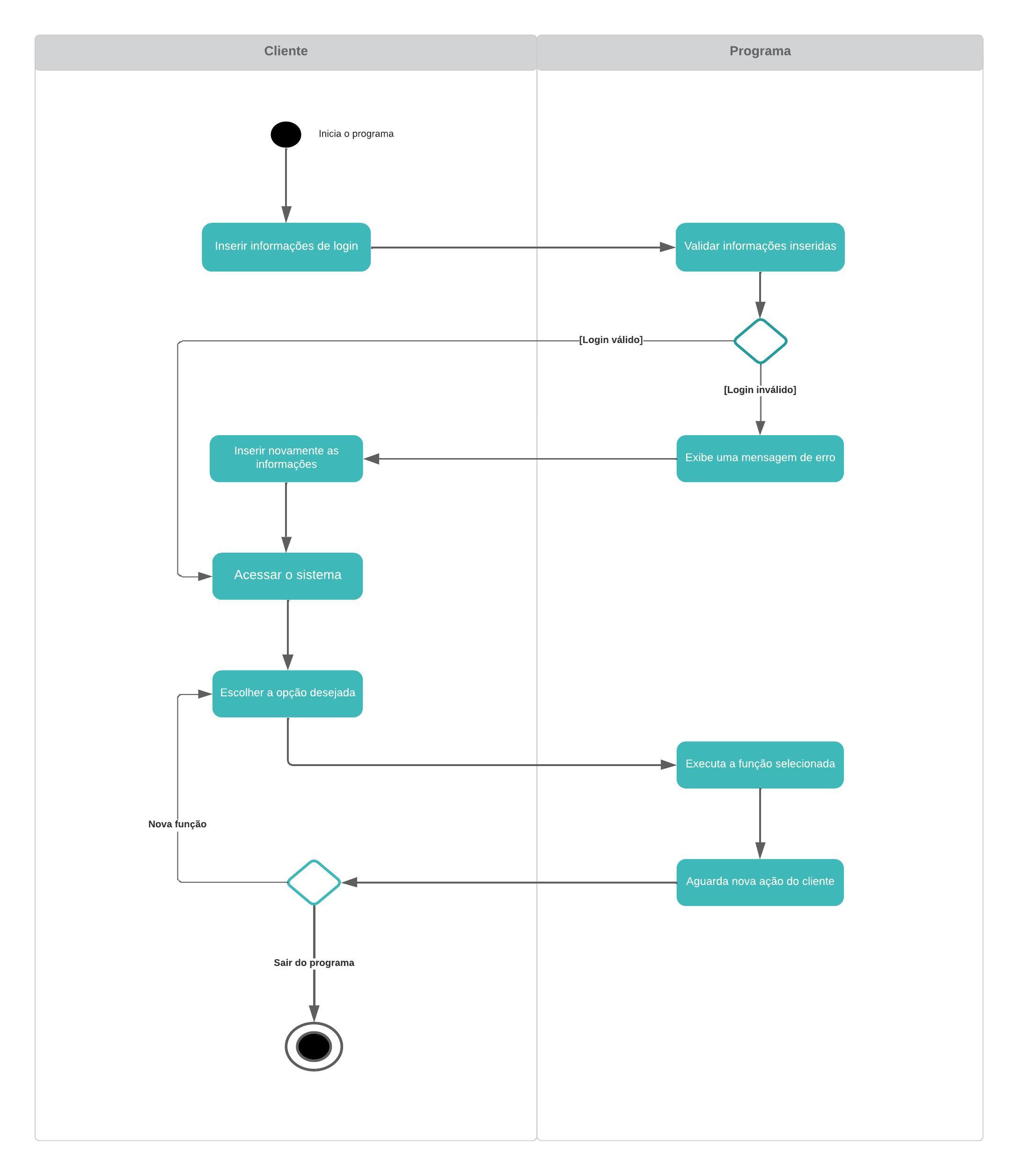
## 6.2 Diagrama de Classe

Figura 19 - Diagrama de Classe Norten



## 6.3 Diagrama de Atividade

Figura 20 - Diagrama de Atividade Norten



# 7. Caso de Uso e Planilha de Teste

Estes casos de usos descrevem as interações entre o usuário e o sistema no processo de cadastro de funcionários e gestão de folhas de pagamento. Ele abrange desde a entrada inicial de dados até a geração de folhas de pagamento e a manutenção de registros.

## 7.1 Casos de sistemas WEB e Mobile

|  |  |
| --- | --- |
| **Nome do caso de Uso** | **Acessar Login** |
| **Caso de Uso Geral** |  |
| **Ator Principal** | Funcionário |
| **Resumo** | Este caso de uso descreve o processo de um usuário realizando seu acesso ao sistema |
| **Pré-condição** | O funcionário precisa estar cadastrado no sistema |
| **Pós-condição** | Trocar a senha de primeiro acesso |
| Fluxo Principal |  |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Criação do registro |  |
|  | 2. Inserção no banco de dados |
| 3. Informa cpf e senha |  |
|  | 4. Consulta CPF e Senha |
|  | 5. Segue para o menu |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Nome do caso de Uso** | **Consultar Minhas Informações** |
| **Caso de Uso Geral** |  |
| **Ator Principal** | Funcionário |
| **Resumo** | Este caso de uso descreve o processo de um usuário acessando suas próprias informações |
| **Pré-condição** | O funcionário precisa estar cadastrado no sistema e ter acesso ao menu |
| **Pós-condição** |  |

|  |  |
| --- | --- |
| Fluxo Principal |  |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Selecionar a opção Minhas Informações |  |
|  | 2. Levar o sistema até a tela “Minhas Informações” |
|  | 3. Consultar no banco de dados os dados do usuário |
|  |  |

|  |  |
| --- | --- |
| **Nome do caso de Uso** | **Consultar Minha Folha de Pagamento** |
| **Caso de Uso Geral** |  |
| **Ator Principal** | Funcionário |
| **Resumo** | Este caso de uso descreve o processo de um usuário acessando sua folha de pagamento |
| **Pré-condição** | O funcionário precisa estar cadastrado no sistema e ter acesso ao menu |
| **Pós-condição** |  |
| Fluxo Principal |  |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Selecionar a opção Folha de pagamento |  |
|  | 2. Levar o sistema até a tela “Folha de pagamentos” |
|  | 3. Consultar no banco de dados os dados do usuário e mostrar na tela |
|  |  |

## 7.2 Casos do sistema Destkop

|  |  |
| --- | --- |
| **Nome do caso de Uso** | **Acessar Login** |
| **Caso de Uso Geral** |  |
| **Ator Principal** | RH |
| **Resumo** | Este caso de uso descreve o processo de um usuário realizando seu acesso ao sistema |
| **Pré-condição** | O RH precisa estar cadastrado no sistema |
| **Pós-condição** |  |
| Fluxo Principal |  |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Criação do registro |  |
|  | 2. Inserção no banco de dados |
| 3. Informa login e senha |  |
|  | 4. Consulta login e Senha |
|  | 5. Segue para o menu |

|  |  |
| --- | --- |
| **Nome do caso de Uso** | **Cadastrar Funcionário** |
| **Caso de Uso Geral** |  |
| **Ator Principal** | RH, Funcionário |
| **Resumo** | Este caso de uso descreve o processo de um cadastro de um funcionário |
| **Pré-condição** | O funcionário deve informar seus documentos |
| **Pós-condição** | O funcionário de RH deve escolher um login e senha. |
| Fluxo Principal |  |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Selecionara a opção de cadastro |  |
| 2. Irá inserir as informações do novo funcionário junto de sua foto |  |
|  | 3. Sistema deverá guardar as informações do cadastro no banco de dados |
|  | 4. Em caso de sucesso envie uma mensagem na tela. |
|  |  |

|  |  |
| --- | --- |
| **Nome do caso de Uso** | **Cadastrar Folha de pagamento** |
| **Caso de Uso Geral** |  |
| **Ator Principal** | RH |
| **Resumo** | Este caso de uso descreve o processo de um cadastro de uma folha de pagamento |
| **Pré-condição** | É necessário conhecimento do CPF e Salário Base do funcionário |
| **Pós-condição** | Para nova inserção é necessário estar atualizado do salário Base |
| Fluxo Principal |  |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Selecionara a opção de cadastro |  |
| 2. Irá inserir as informações do novo funcionário |  |
|  | 3. Sistema deverá guardar as informações do cadastro no banco de dados |
|  | 4. Em caso de sucesso envie uma mensagem na tela. |
|  |  |

|  |  |
| --- | --- |
| **Nome do caso de Uso** | **Consulta Férias** |
| **Caso de Uso Geral** |  |
| **Ator Principal** | RH |
| **Resumo** | Este caso de uso descreve o processo de uma consulta a férias de um funcionário |
| **Pré-condição** | É necessário conhecimento do CPF e data proposta do funcionário |
| **Pós-condição** | Estar a 1 ano desde sua data de admissão |
| Fluxo Principal |  |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Selecionara a opção de consulta |  |
| 2. Irá inserir as informações do funcionário |  |
|  | 3. Sistema irá verificar o banco de dados e a data de admissão |
|  | 4. Em caso de sucesso, sistema enviará mensagem para prosseguir com os documentos. |
|  |  |

## 7.3 Planilha de Teste

**Figura 12 -** Planilha de Teste Norten



# 8. LGPD

A Lei Geral de Proteção de Dados (LGPD, 14 de agosto de 2018) representa um marco jurídico no tratamento de dados pessoais no Brasil. Sua aplicação torna-se fundamental em projetos que envolvem informações sensíveis de indivíduos, como é o caso do sistema integrado de folha de pagamento, cadastro de funcionários e suas respectivas plataformas de acesso.

## 8.1 Princípios Fundamentais da LGPD Aplicados ao Projeto

### Transparência e Consentimento

No contexto do projeto, a transparência é assegurada por meio de políticas claras de privacidade e de tratamento de dados, de fácil acesso a todos os usuários. Além disso, o consentimento explícito dos funcionários é obtido para o tratamento de suas informações, garantindo que estejam cientes e de acordo com o uso de seus dados.

### Finalidade Específica e Necessidade

A coleta e o tratamento de dados no sistema têm propósitos específicos, como a gestão da folha de pagamento e do cadastro de funcionários. A implementação da LGPD garante que apenas informações estritamente necessárias para essas finalidades sejam obtidas, evitando excessos na coleta de dados.

### Segurança e Prevenção

Medidas de segurança são adotadas para proteger os dados de acessos não autorizados ou de possíveis incidentes. Mecanismos como criptografia, controles de acesso e backups regulares são implementados para prevenir incidentes de segurança.

### Consentimento Informado

Ao cadastrar um novo funcionário, é essencial obter seu consentimento explícito para o tratamento de seus dados. Isso pode ser feito por meio de um termo de consentimento presente no processo de admissão.

### Acesso e Retificação de Dados

A LGPD garante aos funcionários o direito de acessar e corrigir suas informações pessoais. No projeto, mecanismos são implementados para que os funcionários possam exercer esses direitos de forma fácil e rápida.

### Interfaces Web, Desktop e Mobile

As plataformas de acesso são desenvolvidas levando em consideração os princípios da LGPD. Mecanismos de segurança, como autenticação e autorização, são implementados em todas as interfaces, garantindo a proteção dos dados independentemente do meio de acesso.

### Treinamento e Conscientização

A equipe de desenvolvimento e os usuários das plataformas são treinados e conscientizados sobre a importância da LGPD e suas responsabilidades na proteção dos dados.

A implementação da LGPD no projeto de folha de pagamento, cadastro de funcionários e plataformas multiplataforma é crucial para assegurar a privacidade e segurança dos dados dos funcionários. Ao seguir os princípios e diretrizes da LGPD, a organização demonstra seu compromisso com a proteção dos direitos dos indivíduos e a conformidade legal, fortalecendo a confiança dos funcionários e parceiros.

# 9. Códigos e Sintaxe

Na ciência da computação, o termo **sintaxe** refere-se às regras que regem a composição de textos com significado em uma linguagem formal, tal como uma linguagem de **programação**, isto é, os textos para os quais faz sentido definir a semântica ou significado, ou fornecer uma interpretação.

## 9.1 Sintaxe Desktop

### 9.1.1 Cadastro de Folha de Pagamento

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Norten

{

public partial class FrmCadFolhaPagamento : Form

{

SqlConnection conn;

public FrmCadFolhaPagamento()

{

InitializeComponent();

conn = new SqlConnection("Data Source=LAB-1\_;Initial Catalog=Norten;Integrated Security=True");

}

public class FolhaDePagamento

{

public int ID\_Func { get; set; }

public decimal SalarioBase { get; set; }

public decimal AdicionalNoturno { get; set; }

public decimal AdiantamentoQuinzenal { get; set; }

public decimal PlanoDeSaude { get; set; }

public decimal INSS { get; set; }

public decimal ValeRefeicao { get; set; }

public decimal ValeTransporte { get; set; }

public decimal FGTS { get; set; }

public decimal DecimoTerceiro { get; set; }

public decimal Ferias { get; set; }

public void CalcularFolhaDePagamento()

{

AdicionalNoturno = SalarioBase \* 0.2m;

AdiantamentoQuinzenal = 100.00m;

PlanoDeSaude = 50.00m;

INSS = 200.00m;

ValeRefeicao = 900.00m;

ValeTransporte = 400.00m;

FGTS = SalarioBase \* 0.08m;

DecimoTerceiro = SalarioBase / 12;

Ferias = SalarioBase \* 1.33m;

}

public decimal CalcularTotalFolhaDePagamento()

{

return SalarioBase + AdicionalNoturno + AdiantamentoQuinzenal + PlanoDeSaude - INSS - ValeRefeicao - ValeTransporte + FGTS + DecimoTerceiro + Ferias;

}

}

public int ObterIDFuncionarioPorCPF(string cpf)

{

int id = -1; // Define um valor padrão para o caso do CPF não ser encontrado.

using (SqlConnection conn = new SqlConnection("Data Source=localhost\\SQLEXPRESS01;Initial Catalog=Norten;Integrated Security=True"))

{

conn.Open();

string query = "SELECT ID\_Func FROM Funcionarios WHERE Func\_CPF = @CPF";

using (SqlCommand command = new SqlCommand(query, conn))

{

command.Parameters.AddWithValue("@CPF", cpf);

object result = command.ExecuteScalar();

if (result != null)

{

id = Convert.ToInt32(result);

}

}

}

return id;

}

public void btnCadFolhaPagamento\_Click(object sender, EventArgs e)

{

string cpf = txtboxFolha\_CPF.Text;

int idFuncionario = ObterIDFuncionarioPorCPF(cpf);

if (idFuncionario != -1)

{

decimal salarioBase = decimal.Parse(txtboxFolha\_Salario.Text);

// Calculando os valores das colunas calculadas

decimal adicionalNoturno = salarioBase \* 0.2m;

decimal fgts = salarioBase \* 0.8m;

decimal decimoTerceiro = salarioBase / 12;

decimal feriasconta = salarioBase / 3m;

decimal ferias = salarioBase + feriasconta;

try

{

conn.Open();

string sql = "INSERT INTO FolhaDePagamento (ID\_Func, Conta, SalarioBase, AdicionalNoturno, FGTS, DecimoTerceiro, Ferias) " +

"VALUES (@ID\_Func, @Conta, @SalarioBase, @AdicionalNoturno, @FGTS, @DecimoTerceiro, @Ferias)";

using (SqlCommand command = new SqlCommand(sql, conn))

{

command.Parameters.AddWithValue("@ID\_Func", idFuncionario);

command.Parameters.AddWithValue("@Conta", "Conta do Funcionário");

command.Parameters.AddWithValue("@SalarioBase", salarioBase);

command.Parameters.AddWithValue("@AdicionalNoturno", adicionalNoturno);

command.Parameters.AddWithValue("@FGTS", fgts);

command.Parameters.AddWithValue("@DecimoTerceiro", decimoTerceiro);

command.Parameters.AddWithValue("@Ferias", ferias);

command.ExecuteNonQuery();

MessageBox.Show("Folha de pagamento inserida com sucesso!", "Sucesso", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show("Ocorreu um erro: " + ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

else

{

MessageBox.Show("Funcionário não encontrado. Verifique o CPF inserido.", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void FrmCadFolhaPagamento\_FormClosed(object sender, FormClosedEventArgs e)

{

FrmFolhaPagamento FP = new FrmFolhaPagamento();

this.Hide();

FP.Show();

conn.Close();

}

}

}

### 9.1.2 Cadastro de Funcionários

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

namespace Norten

{

public partial class FrmCadFunc : Form

{

SqlConnection conn = new SqlConnection("Data Source=LAB-1\_;Initial Catalog=Norten;Integrated Security=True");

public FrmCadFunc()

{

InitializeComponent();

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked)

{

// Mostra o painel

panelFuncRH.Visible = true;

}

else

{

// Oculta o painel

panelFuncRH.Visible = false;

}

}

private void btnInserirImg\_Click(object sender, EventArgs e)

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "Arquivos de Imagem|\*.jpg;\*.jpeg;\*.png;\*.bmp";

if (dialog.ShowDialog() == DialogResult.OK)

{

string caminhoDaImagem = dialog.FileName;

picboxFunc\_Foto.Image = Image.FromFile(caminhoDaImagem);

}

}

private void btnCadFunc\_Click(object sender, EventArgs e)

{

conn.Open();

int? idLoginRH = null;

if (checkBox1.Checked)

{

string consultaUsuarioRH = "INSERT INTO Func\_Usuario\_RH (RH\_Login, RH\_Senha) VALUES (@LoginRH, @SenhaRH); SELECT SCOPE\_IDENTITY();";

SqlCommand comandoUsuarioRH = new SqlCommand(consultaUsuarioRH, conn);

comandoUsuarioRH.Parameters.AddWithValue("@LoginRH", txtboxFunc\_RH\_Login.Text);

comandoUsuarioRH.Parameters.AddWithValue("@SenhaRH", txtboxFunc\_RH\_Senha.Text);

idLoginRH = Convert.ToInt32(comandoUsuarioRH.ExecuteScalar());

}

string consultaFuncionario = "INSERT INTO Funcionarios (Func\_CPF, Func\_NomeCompleto, Func\_DataNascimento, Func\_Genero, Func\_EstadoCivil, Func\_Identificacao, Func\_Endereco, Func\_TelefoneCelular, Func\_Email, Func\_Cargo, Func\_Departamento, Func\_DataAdmissao, Func\_Salario, Func\_CarteiraTrabalho, Func\_Foto, Func\_Senha, idLogin) VALUES (@CPF, @Nome, @DataNascimento, @Genero, @EstadoCivil, @Identificacao, @Endereco, @Telefone, @Email, @Cargo, @Departamento, @DataAdmissao, @Salario, @CarteiraTrabalho, @Foto, @Senha, @idLogin)";

SqlCommand comando = new SqlCommand(consultaFuncionario, conn);

comando.Parameters.AddWithValue("@CPF", txtboxFunc\_CPF.Text);

comando.Parameters.AddWithValue("@Senha", txtboxFunc\_Senha.Text);

comando.Parameters.AddWithValue("@Nome", txtboxFunc\_Nome.Text);

comando.Parameters.AddWithValue("@DataNascimento", dateFunc\_DataNasc.Value);

comando.Parameters.AddWithValue("@Genero", txtboxFunc\_Gen.Text);

comando.Parameters.AddWithValue("@EstadoCivil", txtboxFunc\_EC.Text);

comando.Parameters.AddWithValue("@Identificacao", txtboxFunc\_Ident.Text);

comando.Parameters.AddWithValue("@Endereco", txtboxFunc\_End.Text);

comando.Parameters.AddWithValue("@Telefone", txtboxFunc\_Tel.Text);

comando.Parameters.AddWithValue("@Email", txtboxFunc\_Email.Text);

comando.Parameters.AddWithValue("@Cargo", txtboxFunc\_Cargo.Text);

comando.Parameters.AddWithValue("@Departamento", txtboxFunc\_Depart.Text);

comando.Parameters.AddWithValue("@DataAdmissao", dateFunc\_DataAdm.Value);

comando.Parameters.AddWithValue("@Salario", Convert.ToDecimal(txtboxFunc\_Sal.Text));

comando.Parameters.AddWithValue("@CarteiraTrabalho", txtboxFunc\_Carteira.Text);

if (picboxFunc\_Foto.Image != null)

{

MemoryStream ms = new MemoryStream();

picboxFunc\_Foto.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);

byte[] imagemBytes = ms.ToArray();

SqlParameter paramFoto = new SqlParameter("@Foto", SqlDbType.VarBinary, -1);

paramFoto.Value = imagemBytes;

comando.Parameters.Add(paramFoto);

}

else

{

comando.Parameters.AddWithValue("@Foto", DBNull.Value);

}

if (idLoginRH != null)

{

comando.Parameters.AddWithValue("@idLogin", idLoginRH);

}

else

{

comando.Parameters.AddWithValue("@idLogin", DBNull.Value);

}

comando.ExecuteNonQuery();

conn.Close();

LimparCampos(panelCadFunc);

MessageBox.Show("Cadastro concluído com sucesso!", "Cadastro Concluído", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void btnLimparCampo\_Click(object sender, EventArgs e)

{

LimparCampos(panelCadFunc);

}

public void LimparCampos(Panel panel)

{

foreach (TextBox textbox in panel.Controls.OfType<TextBox>())

{

textbox.Text = "";

}

}

}

}

### 9.1.3 Consultar Férias

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Data.SqlTypes;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Norten

{

public partial class FrmConsultaFerias : Form

{

SqlConnection conn = new SqlConnection("Data Source=LAB-1\_;Initial Catalog=Norten;Integrated Security=True");

public FrmConsultaFerias()

{

InitializeComponent();

}

private void FrmConsultaFerias\_FormClosed(object sender, FormClosedEventArgs e)

{

FrmMenu menuprincipal = new FrmMenu();

this.Hide();

menuprincipal.Show();

conn.Close();

}

private void btnConsultarFerias\_Click(object sender, EventArgs e)

{

string cpf = txtboxFerias\_CPF.Text;

DateTime dataFeriasDesejadas = datetimeFerias.Value; // Obtendo a data do DateTimePicker

try

{

conn.Open();

string sql = "SELECT Func\_CPF, Func\_DataAdmissao FROM Funcionarios WHERE Func\_CPF = @CPF";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@CPF", cpf);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

string cpfEncontrado = reader["Func\_CPF"].ToString();

DateTime dataAdmissao = Convert.ToDateTime(reader["Func\_DataAdmissao"]);

TimeSpan diferenca = dataFeriasDesejadas - dataAdmissao; // Calcula a diferença em dias

int anosDeAdmissao = (int)(diferenca.TotalDays / 365);

if (anosDeAdmissao >= 1)

{

MessageBox.Show("O funcionário com CPF " + cpfEncontrado + " é elegível para férias.", "Elegível", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("O funcionário com CPF " + cpfEncontrado + " não é elegível para férias.", "Não Elegível", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

else

{

MessageBox.Show("Funcionário não encontrado.", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show("Ocorreu um erro: " + ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

}

}

### 9.1.4 Consultar Funcionários

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Norten

{

public partial class FrmConsultaFunc : Form

{

SqlConnection conn = new SqlConnection("Data Source=LAB-1\_;Initial Catalog=Norten;Integrated Security=True");

public FrmConsultaFunc()

{

InitializeComponent();

}

private void FrmConsultaFunc\_FormClosed(object sender, FormClosedEventArgs e)

{

FrmMenu menuprincipal = new FrmMenu();

this.Hide();

menuprincipal.Show();

conn.Close();

}

private void btnBuscarFunc\_Click(object sender, EventArgs e)

{

string cpf = txtboxConsultaCad\_CPF.Text;

try

{

conn.Open();

string sql = "SELECT \* FROM Funcionarios WHERE Func\_CPF = @CPF";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@CPF", cpf);

SqlDataReader reader = cmd.ExecuteReader();

DataTable dataTable = new DataTable();

dataTable.Load(reader);

// Remove as colunas de login, senha e id

dataTable.Columns.Remove("idLogin");

dataTable.Columns.Remove("Func\_Senha");

dataGridViewConsultaFunc.DataSource = dataTable;

if (dataTable.Rows.Count > 0)

{

byte[] fotoBytes = (byte[])dataTable.Rows[0]["Func\_Foto"];

MemoryStream ms = new MemoryStream(fotoBytes);

picboxFunc\_Foto.Image = Image.FromStream(ms);

}

else

{

picboxFunc\_Foto.Image = null; // Limpa a imagem se nenhum registro for encontrado

}

}

catch (Exception ex)

{

MessageBox.Show("Ocorreu um erro: " + ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

}

}

### 9.1.5 Consulta Folha de Pagamento

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Norten

{

public partial class FrmFolhaPagamento : Form

{

SqlConnection conn = new SqlConnection("Data Source=LAB-1\_;Initial Catalog=Norten;Integrated Security=True");

public FrmFolhaPagamento()

{

InitializeComponent();

// Inicialize a conexão no construtor

}

public void ExibirFolhaPagamento(int idFuncionario)

{

string cpf = txtboxFolha\_CPF.Text;

try

{

conn.Open();

string sql = "SELECT ID, Conta, SalarioBase, AdicionalNoturno, AdiantamentoQuinzenal, PlanoDeSaude, " +

"INSS, ValeRefeicao, ValeTransporte, FGTS, DecimoTerceiro, Ferias FROM FolhaDePagamento " +

"WHERE ID\_Func IN (SELECT ID\_Func FROM Funcionarios WHERE Func\_CPF = @CPF)";

using (SqlCommand command = new SqlCommand(sql, conn))

{

command.Parameters.AddWithValue("@CPF", cpf);

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable dt = new DataTable();

adapter.Fill(dt);

dataGridViewFolhaPagamento.DataSource = dt;

if (dataGridViewFolhaPagamento.Columns.Contains("ID\_Func"))

{

dataGridViewFolhaPagamento.Columns["ID\_Func"].Visible = false;

}

if (dataGridViewFolhaPagamento.Columns.Contains("ID"))

{

dataGridViewFolhaPagamento.Columns["ID"].Visible = false;

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ocorreu um erro: " + ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

public int ObterIDFuncionarioPorCPF(string cpf)

{

try

{

conn.Open();

string query = "SELECT ID\_Func FROM Funcionarios WHERE Func\_CPF = @CPF";

using (SqlCommand command = new SqlCommand(query, conn))

{

command.Parameters.AddWithValue("@CPF", cpf);

object result = command.ExecuteScalar();

if (result != null)

{

return Convert.ToInt32(result);

}

else

{

return -1;

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ocorreu um erro: " + ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

return -1;

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

public void BtnBuscarFolha\_Click(object sender, EventArgs e)

{

string cpf = txtboxFolha\_CPF.Text;

int idFuncionario = ObterIDFuncionarioPorCPF(cpf);

if (idFuncionario != -1)

{

ExibirFolhaPagamento(idFuncionario);

}

else

{

MessageBox.Show("Funcionário não encontrado. Verifique o CPF inserido.", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void btnCadFolhaPagamento\_Click(object sender, EventArgs e)

{

// Implemente aqui o código para cadastrar a folha de pagamento

}

private void FrmFolhaPagamento\_FormClosed(object sender, FormClosedEventArgs e)

{

FrmMenu menuprincipal = new FrmMenu();

this.Hide();

menuprincipal.Show();

conn.Close();

}

private void btnFormCADFolhaPagamento\_Click(object sender, EventArgs e)

{

FrmCadFolhaPagamento CadFolha = new FrmCadFolhaPagamento();

this.Hide();

CadFolha.Show();

conn.Close();

}

}

}

### 9.1.6 Tela de Login

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Collections;

namespace Norten

{

public partial class FrmLogin : Form

{

//Referencia de Conexão

//Mudar Data Source e se possivel Initial Catalog

//Data Source = Servidor do Banco (Localhost / Servidor)

//Initial Catalog = Banco de Dados em si (Norten/Cadastro)

SqlConnection conn = new SqlConnection("Data Source=LAB-1\_;Initial Catalog=Norten;Integrated Security=True");

public FrmLogin()

{

InitializeComponent();

txtFunc\_RH\_User.Select();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btnEntrar\_Click(object sender, EventArgs e)

{

conn.Open();

//Iniciando processo de Select

string sqlSelect = "SELECT \* FROM Func\_Usuario\_RH WHERE RH\_Login ='" + txtFunc\_RH\_User.Text + "' AND RH\_senha ='" + txtSenha.Text + "' ";

SqlDataAdapter dp = new SqlDataAdapter(sqlSelect, conn);

DataTable dt = new DataTable();

dp.Fill(dt);

if (dt.Rows.Count == 1)

{

FrmMenu menuprincipal = new FrmMenu();

this.Hide();

menuprincipal.Show();

conn.Close();

}

else

{

MessageBox.Show("Usuario ou Senha Incorretos", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtFunc\_RH\_User.Text = "";

txtSenha.Text = "";

txtFunc\_RH\_User.Select();

conn.Close();

}

}

### 9.1.7 Tela Menu

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Norten

{

public partial class FrmMenu : Form

{

public FrmMenu()

{

InitializeComponent();

}

private void FrmMenu\_FormClosed(object sender, FormClosedEventArgs e)

{

FrmLogin login = new FrmLogin();

login.Show();

}

private void btnFormCadFunc\_Click(object sender, EventArgs e)

{

FrmCadFunc cadFunc = new FrmCadFunc();

this.Hide();

cadFunc.Show();

}

private void btnFormFolhaPagamento\_Click(object sender, EventArgs e)

{

FrmFolhaPagamento FP = new FrmFolhaPagamento();

this.Hide();

FP.Show();

}

private void btnConsultaFerias\_Click(object sender, EventArgs e)

{

FrmConsultaFerias CF = new FrmConsultaFerias();

this.Hide();

CF.Show();

}

private void btnFormListFunc\_Click(object sender, EventArgs e)

{

FrmConsultaFunc CFunc = new FrmConsultaFunc();

this.Hide();

CFunc.Show();

}

}

}

## 9.2 Scripts SQL

Create database Norten

use Norten

go

--CRIANDO TABELA DE Usuario

create table Func\_Usuario\_RH(

idLogin INT IDENTITY PRIMARY KEY,

RH\_Login VARCHAR(50) NOT NULL,

RH\_senha VARCHAR(20) NOT NULL

)

CREATE TABLE Funcionarios (

ID\_Func INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Func\_CPF CHAR(11) NOT NULL UNIQUE,

Func\_NomeCompleto NVARCHAR(100),

Func\_DataNascimento DATE,

Func\_Genero NVARCHAR(20),

Func\_EstadoCivil NVARCHAR(20),

Func\_Identificacao NVARCHAR(20),

Func\_Endereco NVARCHAR(255),

Func\_TelefoneCelular NVARCHAR(15),

Func\_Email NVARCHAR(100),

Func\_Cargo NVARCHAR(50),

Func\_Departamento NVARCHAR(50),

Func\_DataAdmissao DATE,

Func\_Salario DECIMAL(10, 2),

Func\_CarteiraTrabalho NVARCHAR(20),

Func\_Foto VARBINARY(MAX),

Func\_Senha VARCHAR(255) NOT NULL,

idLogin INT NULL

);

ALTER TABLE Funcionarios

ADD CONSTRAINT FK\_Funcionarios\_Usuario

FOREIGN KEY (idLogin) REFERENCES Func\_Usuario\_RH(idLogin);

-- Insere um novo registro na tabela Func\_Usuario\_RH

INSERT INTO Func\_Usuario\_RH (RH\_Login, RH\_senha)

VALUES ('Gui', '123');

-- Insere um novo registro na tabela Funcionarios

INSERT INTO Funcionarios (Func\_CPF, Func\_NomeCompleto, Func\_DataNascimento, Func\_Genero, Func\_EstadoCivil, Func\_Identificacao, Func\_Endereco, Func\_TelefoneCelular, Func\_Email, Func\_Cargo, Func\_Departamento, Func\_DataAdmissao, Func\_Salario, Func\_CarteiraTrabalho, Func\_Foto, Func\_Senha, idLogin)

VALUES ('51627074813', 'Guilherme Carvalho', '2002-04-03', 'M', 'C', 'RG 123456789-0', 'Rua da Paz, 123', '(11) 98765-4321', 'guilhermecarvalho381@email.com', 'Desenvolvedor de software', 'Tecnologia da informação', '2023-10-11', 5000.00, '12345678901', NULL, '123', 1);

SELECT \* FROM FolhaDePagamento

CREATE TABLE FolhaDePagamento (

ID INT IDENTITY(1,1) PRIMARY KEY,

Conta NVARCHAR(100),

SalarioBase DECIMAL(10, 2),

AdicionalNoturno DECIMAL(10, 2),

AdiantamentoQuinzenal DECIMAL(18,2) DEFAULT 0.00,

PlanoDeSaude DECIMAL(18,2) DEFAULT 50.00,

INSS DECIMAL(18,2) DEFAULT 0.00,

ValeRefeicao DECIMAL(18,2) DEFAULT 1200.00,

ValeTransporte DECIMAL(18,2) DEFAULT 1000.00,

FGTS DECIMAL(18,2),

DecimoTerceiro DECIMAL(18,2) DEFAULT 0.00,

Ferias DECIMAL(18,2) DEFAULT 0.00,

ID\_Func INT,

FOREIGN KEY (ID\_Func) REFERENCES Funcionarios(ID\_Func),

);

## 9.3 Sintaxe Web

### 9.3.1 Folha de Pagamento

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Portal de RH - Folha de Pagamento</title>

<link rel="stylesheet" href="style.css">

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<link

href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&amp;display=swap"

rel="stylesheet"

/>

<style>

body {

font-family: "Montserrat", sans-serif;

margin: 0;

padding: 0;

background-color: #f1f1f1;

}

.sidebar {

height: 100%;

width: 250px;

position: fixed;

top: 0;

left: 0;

background-color: #055fbf;

color: #fff;

padding: 20px 0;

}

.sidebar h1 {

text-align: center;

margin: 0;

}

.sidebar ul {

list-style: none;

padding: 0;

margin: 20px 0;

}

.sidebar li {

padding: 10px 0;

text-align: center;

}

.sidebar a {

text-decoration: none;

color: #fff;

display: block;

transition: 0.3s;

}

.sidebar a:hover {

background-color: #ffffff41;

}

.content {

margin-left: 260px;

padding: 20px;

}

h2 {

margin-top: 0;

}

.payroll {

background-color: #fff;

padding: 20px;

border-radius: 5px;

}

/\* Ajuste o estilo conforme desejado para tornar a página mais bonita e profissional \*/

</style>

</head>

<body>

<div class="sidebar"

<ul>

<li><a href="../info/index.html">Minhas Informações</a></li>

<li><a href="index.html">Folha de Pagamento</a></li>

<li><a href="../historico/index.html">Histórico Salarial</a></li>

</ul>

</div>

<div class="content">

<h2>Folha de Pagamento</h2>

<div class="payroll">

<p><strong>Salário Bruto:</strong> R$ 5,000.00</p>

<p><strong>Descontos:</strong> R$ 1,000.00</p>

<p><strong>Salário Líquido:</strong> R$ 4,000.00</p>

</div>

<div class="chart-container">

<canvas id="payroll-chart"></canvas>

</div>

</div>

<script>

var ctx = document.getElementById("payroll-chart").getContext("2d");

var payrollChart = new Chart(ctx, {

type: "bar",

data: {

labels: ["Salário Bruto", "Descontos", "Salário Líquido"],

datasets: [

{

label: "Valores em R$",

data: [5000, 1000, 4000],

backgroundColor: ["#36A2EB", "#FFCE56", "#FF6384"],

},

],

},

options: {

scales: {

y: {

beginAtZero: true,

},

},

},

});

</script>

</body>

</html>

### 9.3.2 Login

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Tela de Login</title>

<link rel="stylesheet" type="text/css" href="style.css" media="screen" />

<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&amp;display=swap"

rel="stylesheet"

/>

</head>

<body>

<div class="container">

<img src="../img/Logo.png" alt="Logo da Empresa" class="logo" />

<form>

<div class="form-group">

<input type="email" placeholder="Email" required />

</div>

<div class="form-group">

<input type="password" placeholder="Senha" required />

</div>

<div class="form-group">

<input type="submit" value="Entrar" />

</div>

<a href="esqueciSenha.html" class="forgot-password">Esqueceu a senha?</a>

</form>

</div>

</body>

</html>

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

using Microsoft.EntityFrameworkCore;

namespace NortenW.Pages

{

public class LoginModel : PageModel

{

private readonly ApplicationDbContext \_context;

public LoginModel(ApplicationDbContext context)

{

\_context = context;

}

[BindProperty]

public string Cpf { get; set; }

[BindProperty]

public string Senha { get; set; }

public IActionResult OnPost()

{

var funcionario = \_context.Funcionarios.FirstOrDefault(f => f.Func\_CPF == Cpf && f.Func\_Senha == Senha);

if (funcionario != null)

{

// Autentica��o bem-sucedida, redirecione para a p�gina protegida

return RedirectToPage("/Index");

}

else

{

ModelState.AddModelError(string.Empty, "Credenciais inv�lidas. Tente novamente.");

return Page();

}

}

}

public class Funcionario

{

public int ID\_Func { get; set; }

public string Func\_CPF { get; set; }

public string Func\_Senha { get; set; }

// Adicione outras propriedades conforme necess�rio

}

public class ApplicationDbContext : DbContext

{

public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)

: base(options)

{

}

public DbSet<Funcionario> Funcionarios { get; set; }

}

}

## 9.4 Sintaxe Mobile

package com.example.norten;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

bt\_entrar = findViewById(R.id.bt\_entrar);

bt\_entrar.setOnClickListener(view ->{

startActivity(new Intent(this, Principal.class));

});

text\_esqueceusenha = findViewById(R.id.text\_esqueceusenha);

text\_esqueceusenha.setOnClickListener(view ->{

startActivity(new Intent(this, Tela\_Esqueceu\_Senha.class));

});

}

public class Principal extends AppCompatActivity {

private View Minhas\_Informacoes;

private TextView Text\_Minhas\_Informacoes;

private View Folha\_Pagamento;

private TextView Text\_Folha\_Pagamento;

private View Historico\_Salarial;

private TextView Text\_Historico\_Salarial;

private Button Button\_Deslogar;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_principal);

Text\_Minhas\_Informacoes = findViewById(R.id.Text\_Minhas\_Informacoes);

Text\_Minhas\_Informacoes.setOnClickListener(view ->{

startActivity(new Intent( this, TelaMinhasInformacoes.class));

});

Minhas\_Informacoes = findViewById(R.id.Minhas\_Informacoes);

Minhas\_Informacoes.setOnClickListener(view ->{

startActivity(new Intent(this, TelaMinhasInformacoes.class));

});

Text\_Folha\_Pagamento = findViewById(R.id.Text\_Folha\_Pagamento);

Text\_Folha\_Pagamento.setOnClickListener(view ->{

startActivity(new Intent(this, Tela\_Folha\_Pagamento.class));

});

Folha\_Pagamento = findViewById(R.id.Folha\_Pagamento);

Folha\_Pagamento.setOnClickListener(view ->{

startActivity(new Intent( this, Tela\_Folha\_Pagamento.class));

});

Text\_Historico\_Salarial = findViewById(R.id.Text\_Historico\_Salarial);

Text\_Historico\_Salarial.setOnClickListener(view ->{

startActivity(new Intent(this, Tela\_Historico\_Salarial.class));

});

Historico\_Salarial = findViewById(R.id.Historico\_Salarial);

Historico\_Salarial.setOnClickListener(view ->{

startActivity(new Intent(this, Tela\_Historico\_Salarial.class));

});

Button\_Deslogar = findViewById(R.id.Button\_Deslogar);

Button\_Deslogar.setOnClickListener(view ->{

startActivity(new Intent(this, Login.class));

});

}

}

# 10. Considerações Finais

No decorrer deste trabalho, exploramos o tema das folhas de pagamentos automatizadas e seu impacto na eficiência e produtividade das empresas. Por meio da análise e desenvolvimento de sistemas, foi possível compreender os benefícios que essa automação traz para a gestão de recursos humanos.

Ao longo da pesquisa, observamos que a automação das folhas de pagamento apresenta diversas vantagens, tanto para os colaboradores quanto para as organizações. A redução de erros e a eliminação de tarefas manuais repetitivas são apenas algumas das melhorias proporcionadas por esse sistema. Além disso, a automatização permite a integração com outros setores, como o financeiro e o contábil, resultando em processos mais ágeis e confiáveis.

Destaca-se também a importância do desenvolvimento de sistemas adequados e personalizados às necessidades específicas de cada empresa. A análise detalhada dos requisitos e a implementação de boas práticas de programação são fundamentais para garantir a eficiência e a segurança do sistema de folha de pagamento automatizada.

É importante ressaltar que as folhas de pagamentos automatizadas não substituem o papel dos profissionais de recursos humanos. Pelo contrário, elas os liberam de atividades operacionais rotineiras, permitindo que se concentrem em atividades estratégicas, como a gestão de talentos, a análise de indicadores e a formulação de políticas de remuneração mais eficazes.

Por fim, o estudo realizado neste trabalho demonstrou que as folhas de pagamentos automatizadas são uma solução promissora e indispensável para as empresas modernas. Sua implementação proporciona benefícios tangíveis e intangíveis, contribuindo para a otimização dos processos organizacionais e para a melhoria do ambiente de trabalho.

É válido ressaltar que a área de análise e desenvolvimento de sistemas possui um papel fundamental na criação e aprimoramento dessas ferramentas. Os profissionais dessa área têm a responsabilidade de aplicar seus conhecimentos técnicos e habilidades para desenvolver sistemas robustos, intuitivos e seguros, que atendam às demandas do mercado e contribuam para o sucesso das organizações.

# REFERÊNCIAS

[CABRAL,](http://www.kerdna.com.br/) JORGE HENRIQUE. Qual a prática do desenvolvimento de software, **Ciência e Cultura**, vol.55, no.2abr. 2003

Disponível em: http://cienciaecultura.bvs.br/scielo.php?script=sci\_arttext&pid=S0009-67252003000200021. Acesso em: 20. maio. 2023

DATE, C. J.. **INTRODUÇÃO A SISTEMAS DE BANCOS DE DADOS**. 8. ed. Rio de Janeiro: Elsevier, 2003. Acesso em: 20. maio. 2023

[MACORATTI,](http://www.kerdna.com.br/) JOSÉ CARLOS. .NET – O ciclo da vida do desenvolvimento de Software, **Macoratti.net [Internet]**.

Disponível em: https://www.macoratti.net/17/09/net\_slcd1.htm. Acesso em: 21. maio. 2023

[FURTADO,](http://www.kerdna.com.br/) MARCELO. Quais são os descontros em folha de pagamento?, **Convenia [Internet]**.

Disponível em: https://blog.convenia.com.br/descontos-da-folha-de-pagamento/ Acesso em: 21. maio. 2023

TEMER, MICHAEL, et al (2018). Lei Geral de Proteção de Dados Pessoais (LGPD). **L13709 – Planalto [Internet]**.

Disponível em: https://www.planalto.gov.br/ccivil\_03/\_ato20152018/2018/lei/l13709.htm

Acesso em: 21. maio. 2023

**UNIVERSIDADE PAULISTA**

**UNIDADE AVENIDA PAULISTA**

**Análise e Desenvolvimento de Sistemas**

Aline Coutinho Trindade

Bianca Souza Santos

Guilherme Carvalho

Guilherme da Costa Silva

Igor Soares

Mariah Eduarda Silva

**Atividade de Extensão Universitária**

**PIM - NORTEN**

**São Paulo**

**2023**

**Relatório de Visita Técnica à Instituição Amigo Incomparável**

Data da Visita: 28/11/2023

**Introdução**

A visita técnica à Instituição Amigo Incomparável revelou importantes lacunas na gestão de dados de colaboradores e beneficiários. A organização, conhecida por seu compromisso em auxiliar comunidades por meio de shows beneficentes e distribuição de cestas básicas, carece de um sistema eficiente para cadastrar e gerenciar informações essenciais. Esta análise detalhada visa identificar as necessidades específicas da instituição e propor soluções que possam otimizar seus processos.

Durante a visita, observou-se que a Amigo Incomparável realiza atividades complexas de gestão de pessoal e beneficiários sem o suporte de um sistema dedicado. A falta de um sistema integrado dificulta a organização e atualização de dados, tornando o processo manual suscetível a erros e limitando a capacidade de análise de informações cruciais.

**Necessidades Identificadas**

**Cadastro de Colaboradores:**

A instituição carece de uma plataforma para registrar informações detalhadas sobre seus colaboradores, incluindo voluntários, artistas envolvidos nos shows beneficentes e membros da equipe administrativa.

**Registro de Beneficiários:**

A falta de um sistema centralizado para cadastrar e acompanhar beneficiários compromete a eficiência na distribuição de cestas básicas. A instituição precisa de uma solução que permita o registro, atualização e rastreamento das informações relevantes.

**Integração de Dados:**

A ausência de integração entre os dados dos colaboradores e beneficiários dificulta a coordenação eficaz das atividades da instituição. Um sistema integrado seria crucial para otimizar a comunicação interna e a alocação de recursos.

**Conclusão**

A visita técnica à Amigo Incomparável destaca a urgência de implementar soluções tecnológicas para melhorar a eficiência operacional e maximizar o impacto social da instituição. A proposta de um sistema personalizado visa preencher as lacunas identificadas, proporcionando à organização as ferramentas necessárias para gerenciar seus colaboradores, beneficiários e atividades beneficentes de forma mais eficaz. Este projeto não apenas fortalecerá a capacidade da Amigo Incomparável de ajudar o próximo, mas também contribuirá para uma gestão mais transparente e eficiente de seus recursos.

**Evidências:**

