

UNIP – UNIVERSIDADE PAULISTA

PROJETO INTEGRADO MULTIDISCIPLINAR – PIM IV
CURSO SUPERIOR DE TECNOLOGIA DE ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

**LUIZ CARLOS DE ARAÚJO MACHADO
GABRIEL NATAN MAGALHÃES BENTO DE MEDEIROS
KAIO HENRIQUE ALVES DA SILVA
JOÃO OTAVIO SANTOS CUNHA**

RHS:
Sistema de Folha de Pagamento

**SANTOS SP
2023**

LUIZ CARLOS DE ARAÚJO MACHADO RA: G4223J6
GABRIEL NATAN MAGALHÃES BENTO DE MEDEIROS RA: N861355
KAIO HENRIQUE ALVES DA SILVA RA: G540ED4
JOÃO OTAVIO SANTOS CUNHA RA: F348CC0

RHS:

Sistema de Folha de Pagamento

Projeto de integração multidisciplinar
PIM apresentado como exigência
para a conclusão de semestre, junto à
Universidade Paulista UNIP.

Orientador: Prof. Thiago Quintão.

SANTOS SP
2023

CIP - Catalogação na Publicação

RHS: Sistema de Folha de Pagamento / Luiz Machado...[et al.]. - 2023.
100 f. : il. color + LGPD e Manual .

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto
de Ciência Exatas e Tecnologia da Universidade Paulista, Santos, 2023.
Área de Concentração: Tecnologia da informação.
Orientador: Prof. MBA Thiago Quintão.

1. Recursos Humanos. 2. Departamento Pessoal. 3. Desenvolvimento.
4. Sistema. 5. Folha de Pagamento. I. Machado, Luiz. II. Quintão, Thiago
(orientador).

LUIZ CARLOS DE ARAÚJO MACHADO RA: G4223J6
GABRIEL NATAN MAGALHÃES BENTO DE MEDEIROS RA: N861355
KAIO HENRIQUE ALVES DA SILVA RA: G540ED4
JOÃO OTAVIO SANTOS CUNHA RA: F348CC0

RHS:

Sistema de Folha de Pagamento

Projeto de integração multidisciplinar
PIM apresentada como exigência
para a conclusão de semestre, junto à
Universidade Paulista UNIP.

Orientador: Prof. Thiago Quintão.

Aprovado em:

BANCA EXAMINADORA

_____/_____

Prof. Nome do Professor
Universidade Paulista – UNIP

_____/_____

Prof. Nome do Professor
Universidade Paulista – UNIP

_____/_____

Prof. Nome do Professor
Universidade Paulista – UNIP

RESUMO

A empresa RHS (Recursos Humanos Solution) especializada em Recursos Humanos (RH) e Departamento Pessoal (DP) oferece serviços de subcontratação de processos administrativos para empresas de pequeno porte. Apesar de possuir um ecossistema de software abrangente que inclui cálculo de folha de pagamento, recibos de férias, pontos de atendimento, exames médicos, contratos de trabalho, entre outros, os programas não estão integrados e não compartilham um banco de dados comum. Isso resulta em atrasos e erros na transferência de dados entre os diferentes programas. Para resolver esse problema, a RHS decidiu contratar a Devcoast (Grupo PIM) para desenvolver um novo sistema que atenda às necessidades da empresa começando com a folha de pagamento. Apresentaremos neste trabalho a documentação, o desenvolvimento e o sistema de folha de pagamento que conta com 3 aplicações distintas (Mobile, Web, Desktop), que se comunicam de forma direta com o banco de dados unificado na nuvem.

Palavras-chave: Recursos Humanos. Departamento Pessoal. Desenvolvimento. Sistema. Folha de Pagamento.

ABSTRACT

The company RHS (Human Resources Solution) specialized in Human Resources (HR) and Personnel Department (DP) offers administrative process subcontracting services for small companies. Despite having a comprehensive software ecosystem that includes payroll calculation, vacation receipts, service points, medical exams, employment contracts, among others, the programs are not integrated and do not share a common database. This results in delays and errors in transferring data between different programs. To solve this problem, RHS decided to hire Devcoast (PIM Group) to develop a new system that meets the company's needs starting with payroll. In this work, we will present all the documentation and development of the time sheet system, which has 3 different applications (Mobile, Web, Desktop), which communicate directly with the unified database in the cloud.

Keywords: Human Resources. Personal department. Development. System. Payroll.

LISTA DE ILUSTRAÇÕES

Figura 1: Logo Devcoast	10
Figura 2: Diagrama de caso de uso do sistema de cadastro e login.	25
Figura 3: Diagrama de caso de uso do sistema folha de pagamento.....	26
Figura 4: Diagrama de atividades de login.	27
Figura 5: Diagrama de atividades de cadastro.	28
Figura 6: Diagrama de atividades da folha de pagamento recorte 1.	29
Figura 7: Diagrama de atividades da folha de pagamento recorte 2.	30
Figura 8: Diagrama de atividades da folha de pagamento completo.....	31
Figura 9: Diagrama de classes parte 1/3.....	31
Figura 10: Diagrama de classes parte 2/3.....	32
Figura 11: Diagrama de classes parte 3/3.....	33
Figura 12: Diagrama de classes completo.	34
Figura 13: Arquitetura da integração dos sistemas.	35
Figura 14: MER conceitual do banco de dados.....	36
Figura 15: MER do banco de dados.	37
Figura 16: SGBD pgAdmin 4.....	38
Figura 17: Tela inicial ElephantSQL.	39
Figura 18: Script Banco de dados: criação da tabela conta bancária.....	39
Figura 19: Script Banco de dados: criação da tabela empresa.	39
Figura 20: Script Banco de dados: criação da tabela folha de pagamento.	40
Figura 21: Script Banco de dados: criação da tabela registro de ponto.	40
Figura 22: Script Banco de dados: criação da tabela funcionário.....	41
Figura 23: Script Banco de dados: criação da tabela salário família.	41
Figura 24: Script Banco de dados: criação da tabela progressiva inss e irrf.	42
Figura 25: Script Banco de dados: criação da tabela usuário.	42
Figura 26: Script Banco de dados: criação da tabela dependentes.	42
Figura 27: Classe BancoDados desenvolvida em C#.	43
Figura 28: Classe BancoDados desenvolvida em Dart, parte 1/2.	43
Figura 29: Classe BancoDados desenvolvida em Dart, parte 2/2.	44
Figura 30: Scripts Banco de dados executados em um servidor.js, parte 1/3.....	44
Figura 31: Scripts Banco de dados executados em um servidor.js, parte 2/3.....	45
Figura 32: Scripts Banco de dados executados em um servidor.js, parte 3/3.....	45

Figura 33: Logo do app Registro de Ponto.....	50
Figura 34: Apresentação do app Registro de ponto.....	50
Figura 35: Logo do site Portal do Funcionário.....	51
Figura 36: Apresentação site Portal do Funcionário.....	51
Figura 37: Logo do app Folha de Pagamento.	52
Figura 38: Apresentação app Folha de pagamento.	52
Figura 39: Mind Map do projeto de interface da folha de pagamento.	53
Figura 40: Firebase console Authentication.	66
Figura 41: Função de criptografia em C#.	67
Figura 42: Lista de usuários com senha criptografadas.	67
Figura 43: API ElephantSQL.	68
Figura 44: Classe FolhaPagamento desenvolvido em C#, parte 1/7.....	69
Figura 45: Classe FolhaPagamento desenvolvido em C#, parte 2/7.....	70
Figura 46: Classe FolhaPagamento desenvolvido em C#, parte 3/7.....	71
Figura 47: Classe FolhaPagamento desenvolvido em C#, parte 4/7.....	72
Figura 48: Classe FolhaPagamento desenvolvido em C#, parte 5/7.....	73
Figura 49: Classe FolhaPagamento desenvolvido em C#, parte 6/7.....	74
Figura 50: Classe FolhaPagamento desenvolvido em C#, parte 7/7.....	75
Figura 51: Classe HomePage desenvolvido em Dart no Flutter, parte 1/11.....	76
Figura 52: Classe HomePage desenvolvido em Dart no Flutter, parte 2/11.....	77
Figura 53: Classe HomePage desenvolvido em Dart no Flutter, parte 3/11.....	77
Figura 54: Classe HomePage desenvolvido em Dart no Flutter, parte 4/11.....	78
Figura 55: Classe HomePage desenvolvido em Dart no Flutter, parte 5/11.....	78
Figura 56: Classe HomePage desenvolvido em Dart no Flutter, parte 6/11.....	79
Figura 57: Classe HomePage desenvolvido em Dart no Flutter, parte 7/11.....	79
Figura 58: Classe HomePage desenvolvido em Dart no Flutter, parte 8/11.....	80
Figura 59: Classe HomePage desenvolvido em Dart no Flutter, parte 9/11.....	80
Figura 60: Classe HomePage desenvolvido em Dart no Flutter, parte 10/11.....	81
Figura 61: Classe HomePage desenvolvido em Dart no Flutter, parte 11/11.....	81
Figura 62: Index.js do componente Home, desenvolvido em JavaScript, parte 1/2. .84	84
Figura 63: Index.js do componente Home, desenvolvido em JavaScript, parte 2/2. .84	84

LISTA DE TABELAS

Tabela 1: Tabela de contribuição previdenciária – INSS.....	18
Tabela 2: Tabela imposto sobre a renda das pessoas físicas.....	18
Tabela 3: Requisitos de Usuários.....	21
Tabela 4: Requisitos de Sistema - Funcionais	22
Tabela 5: Requisitos de Sistema - Não Funcionais.....	23
Tabela 6: Entidade: Funcionário.....	46
Tabela 7: Entidade: Empresa.....	46
Tabela 8: Entidade: ValoresFolha	47
Tabela 9: Entidade: ContaBancaria.....	47
Tabela 10: Entidade: Empresa.....	47
Tabela 11: Entidade: FolhaPagamento.....	48
Tabela 12: Entidade: TabelaProgressiva.	48
Tabela 13: Entidade: Usuario.....	49
Tabela 14: Tabela da complexidade dos campos.	53
Tabela 15: Requisito de interface RF001.	54
Tabela 16: Requisito de interface RF002.	55
Tabela 17: Requisitos de interface RF003, RF004 e RF005.....	56
Tabela 18: Requisito de interface RF006.	58
Tabela 19: Requisito de interface RF007.	61
Tabela 20: Requisito de interface RF008.	62
Tabela 21: Requisito de interface RF009.	63
Tabela 22: Requisito de interface RF010.	64
Tabela 23: Requisito de interface RF011.	65

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVO GERAL	11
2.1	Objetivos específicos.....	11
3	Referencial teórico	12
3.1	Recursos Humanos	12
3.1.1	Gestão de Recursos Humanos	12
3.1.2	Departamento Pessoal	13
3.1.3	Entrevista com o profissional da área de RH	13
3.1.4	Folha de Pagamento	15
3.1.4.1	Principais proventos da folha de pagamento	16
3.1.4.2	Principais descontos da folha de pagamento.....	17
3.1.4.3	Tabelas Contribuição previdenciária (INSS) e IRRF	18
4	DESENVOLVIMENTO	19
4.1	RHS (Recursos Humanos Solution)	19
4.1.1	RHS: atividades da folha de pagamento	19
4.2	Sistema de folha de pagamento	20
4.2.1	Requisitos do sistema	20
4.2.1.1	Requisitos funcionais e não funcionais	21
4.2.1.2	Diagramas de caso de uso.....	25
4.2.1.3	Diagramas de atividades	27
4.2.1.4	Diagrama de classes.....	31
4.3	Arquitetura do sistema.....	35
4.4	Banco de dados.....	36
4.4.1	Modelo Entidade-Relacionamento.....	36
4.4.2	PostgreSQL.....	37
4.4.3	Scripts do Banco de dados.....	39
4.4.4	Classes Banco de dados	42
4.4.5	Dicionário de dados.....	46
4.5	Design (UX)	49
4.5.1	Interface Gráfica.....	53
4.6	Lei Geral de Proteção de Dados (LGPD)	66

4.7 Código Fonte	68
5 CONCLUSÃO	85
REFERÊNCIAS.....	86
ANEXO A - TERMO DE CONSENTIMENTO PARA TRATAMENTO DE DADOS PESSOAIS LEI GERAL DE PROTEÇÃO DE DADOS PESSOAIS – LGPD	88
ANEXO B – MANUAL DO USUÁRIO.....	91

1 INTRODUÇÃO

A empresa RHS (Recursos Humanos Solution) especializada nas atividades de Recursos humanos (RH) e Departamento Pessoal (DP) atende muitas empresas de pequeno porte, através da subcontratação de diversos processos administrativos de RH e DP. Essa empresa já possui um ecossistema de software que fornece uma gama de serviços para o espaço de RH e PD, como softwares que calculam folha de pagamento, geram recibos de férias, pontos de atendimento, exames médicos, contratos de trabalho etc. Entretanto, esses diferentes programas não são integrados, não possuem um banco de dados comum, o que causa atrasos e erros durante a transferência de dados de um programa para outro.

Visando esse problema a RHS decidiu recorrer a Devcoast (Grupo PIM), para desenvolver um sistema que atende as necessidades da empresa.

Figura 1: Logo Devcoast



Fonte: Próprios autores, 2023.

Após várias Reuniões e entrevistas, foi decidido que os softwares atuais da empresa serão gradativamente substituídos pelo novo sistema, começando pelo cálculo e emissão de relatórios da folha de pagamento que está disponível nas plataformas de desktop (apenas para clientes para serviços internos confidenciais), plataformas da web e dispositivos móveis (para funcionários dos clientes).

2 OBJETIVO GERAL

Desenvolver um sistema de folha de pagamento eficiente e intuitivo que automatize e simplifique o processo de cálculo, garantindo a conformidade legal e a precisão dos dados.

2.1 Objetivos específicos

- Desenvolver e aplicar os conhecimentos adquiridos em sala de aula;
- Desenvolver algoritmos e os respectivos fluxogramas;
- Explorar e utilizar ferramentas computacionais para desenvolvimento e elaboração de diagramas de fluxo de software;
- Uso da matemática em programas de computador;
- Debagar o código passo a passo e investigar o funcionamento dele.

3 REFERENCIAL TEÓRICO

3.1 Recursos Humanos

Segundo (Editorial do Indeed, 2023) o departamento de Recursos Humanos é responsável por cuidar do ativo mais valioso da empresa: seus funcionários. Sua função é garantir a satisfação e o engajamento dos colaboradores, além de fornecer os recursos e materiais necessários para que desempenhem suas atividades de forma adequada. Para (Editorial do Indeed, 2023) o RH desempenha um papel fundamental na gestão das pessoas dentro da organização, estabelecendo uma relação direta com os funcionários e promovendo a cultura específica da empresa.

3.1.1 Gestão de Recursos Humanos

De acordo com (Marques, 2016). A gestão de recursos humanos tem os seguintes objetivos:

- Planejar e dimensionar, de maneira quantitativa e qualitativa, as necessidades da organização, com o objetivo de atender as especificidades do negócio e alcançar os objetivos estratégicos;
- Desempenhar o papel de designer organizacional, criando uma estrutura moderna e flexível que permita que a organização alcance seus objetivos se adaptando de forma ágil às demandas de um mercado competitivo;
- Realizar a seleção e recrutamento de profissionais qualificados para suprir as necessidades da organização;
- Desenvolver o potencial dos colaboradores por meio de programas de capacitação e treinamentos;
- Reconhecer a importância do trabalho e o desempenho individual de cada pessoa na organização, garantindo uma remuneração justa.

3.1.2 Departamento Pessoal

Segundo (Guimarães, 2023), o Departamento Pessoal é uma subdivisão do setor de Recursos Humanos que tem a responsabilidade de administrar os funcionários. Suas atividades e procedimentos incluem a contratação e demissão de colaboradores, o controle de férias, a gestão da folha de pagamento, a administração de benefícios, entre outras tarefas relacionadas.

Para (Senior, 2022), o objetivo principal do Departamento Pessoal é assegurar que a organização esteja em conformidade com as leis trabalhistas. Em essência, o DP funciona como um repositório de informações da empresa, sendo procurado por gestores e colaboradores para obter informações sobre:

- Processos de admissão e demissão em andamento;
- Períodos estabelecidos para licenças;
- Agendamento de férias;
- Registros de controle de ponto dos funcionários;
- Informações sobre alterações salariais.

Além disso o autor aponta que o DP também é responsável por garantir a conformidade da empresa com a legislação trabalhista, incluindo o cumprimento das obrigações mensais de pagamento de FGTS (Fundo de Garantia do Tempo de Serviço) e INSS (Instituto Nacional do Seguro Social) dos colaboradores.

3.1.3 Entrevista com o profissional da área de RH

A equipe Devcoast realizou uma entrevista com um profissional da área de RH, foram aplicadas 5 perguntas importantes com o foco de levantar algumas informações importantes sobre RH, DP e folha de pagamento.

1 – Qual a importância do RH e do departamento pessoal nas empresas?

De maneira geral, o DP é responsável pelas questões burocráticas referente aos colaboradores, leis trabalhistas e algumas questões fiscais.

Já o RH é mais estratégico, e cuida da capacitação e treinamento de colaboradores, atração e retenção de talentos e demais questões pertinentes às relações humanas.

2 – Qual a diferença entre RH e departamento pessoal?

A principal diferença do RH para o Departamento Pessoal é que, enquanto uma lida com atividades focadas nas relações humanas, o outro atende à burocracia sempre presente e necessária dentro de qualquer organização.

3 – Quais os principais problemas enfrentados pelo RH quando se trata do departamento pessoal?

Existem situações que todo o profissional de RH vai ter que enfrentar pelo menos uma vez na vida: problemas com a burocracia do setor; mudanças legislativas; conflitos entre colegas; líderes complicados; Muitas vezes as informações chegam invertidas até o RH

4 - Quais são as principais rotinas do departamento pessoal?

Participar dos processos de admissão, demissões e desligamento, fazer a gestão da folha de pagamento, gerar encargos mensais e anuais, garantir o controle de horários, apoiar os setores no planejamento de férias, manter o compliance trabalhista etc.

5 – Como é a rotina de folha de pagamento?

A folha de pagamento é um documento que deve ser emitido mensalmente pelos empregadores, no qual estão descritas todas as informações referentes aos valores pagos e descontados dos trabalhadores.

Ou seja, a folha de pagamento é responsável por compilar todas as informações relacionadas à jornada do empregado durante o mês ou determinado período. Portanto, para que a apresentação dos cálculos seja a mais efetiva possível, é preciso que a equipe do Departamento Pessoal da empresa detenha certos conhecimentos técnicos sobre as leis trabalhistas e contabilidade.

As informações levantadas nesta entrevista foram essenciais para conduzir a pesquisa do referencial teórico e o desenvolvimento do sistema de folha de pagamento.

3.1.4 Folha de Pagamento

Segundo (Equipe TOTVS, 2023), a folha de pagamento é um registro que documenta todas as informações relacionadas à remuneração dos colaboradores, apresentando o total dos valores pagos pela empresa. Segundo o autor esses registros geralmente incluem os pagamentos recebidos pelo trabalhador durante um período específico, bem como informações adicionais relevantes, como deduções fiscais, benefícios e outras deduções. O autor ressalta que além de registrar e comprovar o pagamento desses valores, o documento também serve como um histórico de trabalho, que auxilia na gestão da empresa e na vida pessoal dos colaboradores, pois eles podem utilizá-lo em procedimentos como aposentadoria e financiamentos.

A (Equipe TOTVS, 2023) ressalta que a emissão mensal desse documento é obrigatória de acordo com as determinações do Decreto 3048/1999, aplicando-se a todas as empresas.

(Morgana Perini, 2023) destaca que a folha de pagamento deve conter os seguintes requisitos:

- Dados do empregador, incluindo informações como nome, razão social, CNPJ e endereço;
- Dados do empregado, como nome, CPF, cargo e função exercida;
- Descontos realizados, tais como INSS, contribuição sindical, FGTS, vales e outros descontos pertinentes;
- Número de dias trabalhados no período abrangido pela folha de pagamento;
- Valores correspondentes a horas extras e adiantamentos, quando aplicáveis;
- Valor bruto do salário, representando o montante total antes dos descontos;
- Valor líquido do salário, que indica o valor final a ser recebido pelo empregado após as deduções;

- Agrupamento dos segurados por categoria, abrangendo segurados empregados, trabalhadores avulsos, contribuintes individuais e empregados contratados por prazo determinado;
- Destaque do nome das seguradas em gozo de salário-maternidade;
- Indicação do número de quotas de salário-família atribuídas a cada segurado empregado ou trabalhador avulso.

Segundo (Morgana Perini, 2023), esses requisitos são necessários para garantir a devida documentação e transparência dos pagamentos efetuados pela empresa aos seus colaboradores.

3.1.4.1 Principais proventos da folha de pagamento

(Morgana Perini, 2023) aponta quais são os principais proventos da folha de pagamento:

- Salário: Valor fixo pago ao empregado pelo trabalho realizado;
- Horas extras: Valor adicional pago ao colaborador por horas trabalhadas além da jornada estabelecida;
- Descanso Semanal Remunerado (DSR): Pagamento pelo dia de descanso semanal do trabalhador;
- Adicional Noturno: Valor adicional pago ao trabalhador que realiza suas atividades durante a noite;
- Adicional de Insalubridade: Valor pago ao colaborador que trabalha em condições insalubres;
- Adicional de Periculosidade: Valor pago ao colaborador que exerce atividades de risco;
- Salário-família: Benefício previdenciário concedido ao colaborador com filhos menores de 14 anos ou com dependentes inválidos;
- Diárias para viagens: Valores pagos ao colaborador para cobrir despesas em viagens a trabalho;
- Adicional por tempo de serviço: Benefício concedido pela empresa em acordo ou convenção coletiva, baseado no tempo de serviço do colaborador;

- Auxílio-Creche/Babá: Benefício oferecido por empresas com mais de 30 colaboradoras com filhos de 0 a 6 meses, que podem optar por um espaço físico ou receber auxílio financeiro para cuidados infantis.

Segundo (Morgana Perini, 2023), esses proventos compõem a remuneração total do colaborador e são essenciais para seu sustento e bem-estar. Segundo as autoras cabe às empresas garantir o correto cálculo e pagamento desses valores, em conformidade com as leis trabalhistas e acordos estabelecidos.

3.1.4.2 Principais descontos da folha de pagamento

(Furtado, 2022), destaca em lista os principais descontos da folha de pagamento:

- INSS: Imposto obrigatório que cobre benefícios previdenciários, como licença-maternidade e auxílio-doença;
- FGTS: Fundo de Garantia por Tempo de Serviço, depositado pelo empregador para proteger o trabalhador em demissões sem justa causa;
- Imposto de Renda (IRRF): Desconto determinado pelo governo com base na remuneração do colaborador;
- Adiantamento Salarial: Valor adiantado ao funcionário que é descontado posteriormente;
- Vale-transporte: Benefício obrigatório que pode gerar descontos de até 6% do salário base, caso o valor entregue seja maior;
- Vale-refeição: Benefício oferecido por algumas empresas, podendo ser descontado até 20% do valor;
- Atrasos e Faltas: Descontos aplicados em casos de ausências não justificadas pelo colaborador.

Segundo (Furtado, 2022) é importante lembrar que o total de descontos não deve ultrapassar 70% do salário do empregado. De acordo com (Furtado, 2022), cada desconto tem sua base legal e deve ser aplicado de acordo com as regras estabelecidas pelas legislações pertinentes. Segundo o autor o gestor deve

consultar as leis e regulamentações adequadas para garantir a correta aplicação dos descontos na folha de pagamento.

3.1.4.3 Tabelas Contribuição previdenciária (INSS) e IRRF

Tabelas de contribuição atualizadas segundo a portaria Interministerial MPS/MF Nº 26, de 10 de janeiro de 2023 e o Decreto Nº 9.580, de 22 de novembro de 2018.

Tabela 1: Tabela de contribuição previdenciária – INSS.

TABELA DE CONTRIBUIÇÃO PREVIDENCIÁRIA - INSS	
Faixa Salarial	Alíquota Progressiva
Taxa de desconto até R\$ 1.302,00	7,5%
Taxa de desconto entre R\$ 1.302,01 e R\$ 2.571,29	9%
Taxa de desconto entre R\$ 2.571,30 e R\$ 3.856,94	12%
Taxa de desconto entre R\$ 3.856,95 e R\$ 7.507,49	14%

Fonte: Portaria Interministerial MPS/MF nº 26, de 10 de janeiro de 2023.

Tabela 2: Tabela imposto sobre a renda das pessoas físicas.

IMPOSTO SOBRE A RENDA DAS PESSOAS FÍSICAS (vigente desde abril de 2015)		
Base de cálculo	Alíquota	Parcela a deduzir
Até R\$ 1.903,98	-	-
De R\$ 1.903,99 até R\$ 2.826,65	7,5 %	R\$ 142,80
De R\$ 2.826,66 até R\$ 3.751,05	15%	R\$ 354,80
De R\$ 3.751,06 até R\$ 4.664,68	22,5 %	R\$ 636,13
Acima de R\$ 4.664,68	27,5 %	R\$ 869,36

Fonte: Decreto nº 9.580/2018.

Valor da dedução mensal por dependente: R\$ 189,59.

4 DESENVOLVIMENTO

4.1 RHS (Recursos Humanos Solution)

Empresa fundada em 2016 especializada em gestão dos Recursos Humanos e de Departamento Pessoal, que atende empresas de pequeno porte, oferecendo soluções modernas, ágeis e personalizadas buscando a satisfação dos clientes, fornecendo serviços de qualidade focados em atender às diversas necessidades de seus clientes. A RHS prioriza a gestão de pessoas e investe no desenvolvimento, treinamento e motivação de talentos, reconhecendo a importância das pessoas para o sucesso dos negócios. A RHS age de forma ética e transparente, cumpre as regras e regulamentos aplicáveis e garante a integridade e confidencialidade das informações de clientes e funcionários. A RHS incentiva a inovação e a melhoria contínua de processos e serviços, buscando constantemente novas formas de agregar valor aos seus clientes e melhorar a eficiência e eficácia da empresa.

4.1.1 RHS: atividades da folha de pagamento

Atividades relacionadas a folha de pagamento levantadas em entrevistas com a RHS:

- Realizar a coleta de dados dos funcionários, como informações pessoais, salário base, cargos, horas trabalhadas, benefícios e descontos;
- Efetuar os cálculos dos salários dos funcionários, levando em conta as horas trabalhadas, salário base, adicional de horas extras, descontos legais (INSS, IRRF) e benefícios (vale-transporte, vale-refeição etc.);
- Verificar faltas, atrasos e demais situações que podem impactar a remuneração;
- Verificar e aplicar as alíquotas previdenciárias, obedecendo ao teto pré-estabelecido pelo INSS;
- Realizar cálculo de rescisões e demais verbas trabalhistas a serem pagas em caso de rescisão contratual;
- Gerar holerites, com as informações detalhadas do valor bruto e líquido a ser pago aos funcionários;

- Acompanhar as normas e legislações trabalhistas e previdenciárias que possam impactar a folha de pagamento;
- Prestar atendimento aos funcionários, esclarecendo dúvidas e fornecendo informações sobre a remuneração e descontos.

4.2 Sistema de folha de pagamento

4.2.1 Requisitos do sistema

O sistema deve ser capaz de controlar o acesso, permitindo que os usuários tenham um perfil e login próprio previamente cadastrados pelo Gerente, cada perfil deverá possuir as seguintes informações: nome completo, cargo, login e senha. Cada cadastro possuirá acesso apenas para informações importantes para sua função.

O sistema deve ser capaz de realizar o cadastro de empresas clientes e tratar a folha de pagamento de seus funcionários de acordo com suas particularidades.

O sistema deve possuir uma área para a realização de cadastro de funcionário, possibilitando a entrada de informações pessoais, salariais, benefícios e descontos e permitir atualizações e edições frequentes.

O sistema deve possuir uma rotina de cálculo da folha de pagamento, que deve considerar todas as informações do cadastro do funcionário, como salário base, horas trabalhadas, benefícios e descontos previstos em lei (INSS, IRRF, entre outros), além de faltas e atrasos. O cálculo deve estar de acordo com as normas previdenciárias e trabalhistas.

O sistema deve ser capaz de realizar o controle de autorização para a emissão da folha de pagamento de cada funcionário.

O sistema deve permitir que o Gerente e usuários revisem as informações antes da geração da folha de pagamento.

O sistema deve permitir a emissão dos holerites dos funcionários em PDF, contendo as informações de remuneração e desconto detalhadas, correspondentes àquela folha de pagamento.

O sistema deve possuir funcionalidades de segurança dos dados, garantindo que as informações dos funcionários estejam preservadas e a salvo em caso de incidentes ou perda de dados.

4.2.1.1 Requisitos funcionais e não funcionais

Tabela 3: Requisitos de Usuários.

Requisitos de Usuários	
Requisitos Funcionais	Requisitos Não Funcionais
[RF001] - O sistema deverá permitir seu acesso através de um sistema de login e senha.	[RNF001] - Facilidade de uso e navegação.
[RF002] - O sistema deverá permitir que o gerente realize seu auto cadastro.	[RNF002] - O sistema deverá ser capaz de executar seus processos de forma rápida.
[RF003] - O sistema deverá permitir que o gerente realize o cadastro de empresas clientes.	[RNF003] - O sistema deverá ser adaptável ao cliente.
[RF004] - O sistema deverá permitir que o gerente realize o cadastro de usuários no sistema.	[RNF004] - O sistema deverá suportar usuários simultâneos.
[RF005] - O sistema deverá permitir que os usuários cadastrem funcionários no sistema.	[RNF005] - O sistema deverá utilizar criptografia para cifrar e decifrar as senhas cadastradas.
[RF006] - O sistema deverá permitir que os usuários lancem informações dos funcionários pertinentes à folha de pagamento.	[RNF006] - A folha de pagamento deverá ser impressa em PDF.
[RF007] - O sistema deverá ser capaz de realizar os cálculos da folha de pagamento.	[RNF007] - Os cálculos da folha de pagamento deverão estar de acordo com as normas previdenciárias e trabalhistas.
[RF008] - O sistema deverá permitir revisar a folha antes da geração da folha de pagamento.	[RNF008] - O sistema deverá ser desenvolvido para as plataformas desktop, web e dispositivos móveis.
[RF009] - O sistema deverá permitir a emissão da folha de pagamento dos funcionários.	[RNF009] - O sistema deverá fazer o uso de um banco de dados.

Fonte: Próprios autores, 2023.

Tabela 4: Requisitos de Sistema - Funcionais.

Requisitos de Sistema - Funcionais	
[RF001]	<p>1 - O primeiro acesso ao sistema deverá ser realizado através de um "perfil padrão".</p> <p>2 - O login do gerente ou administrador deverá ser cadastrado.</p>
[RF002]	<p>1 - O perfil padrão deverá ser utilizado pelo gerente para a realização do seu auto cadastro.</p> <p>2 - O auto cadastro realizado pelo gerente conterá os campos a serem preenchidos em uma única aba "Cadastro de Usuário".</p> <p>3 - Na aba "Cadastro de Usuário" deverá ter os seguintes campos a serem preenchidos: nome, departamento, cargo, selecionar perfil e definir senha.</p> <p>4 - O gerente é o único perfil que possui total acesso ao sistema, não havendo restrições.</p>
[RF003]	1 - O cadastro do cliente deverá ser realizado apenas pelo perfil do Gerente e em uma única aba e conter os seguintes campos: razão social, CNPJ, endereço, cidade, estado, telefone, área de atuação, quantidade de funcionários, esta aba deverá ser chamada de "Cadastro de Cliente".
[RF004]	1 - O Gerente é o único perfil que poderá cadastrar ou excluir administradores e restringir os acessos de acordo com a sua função.
[RF005]	<p>1 - O cadastro de funcionário será realizado em uma única aba e os campos divididos em áreas, como informações pessoais, contratuais, e bancárias, esta aba deverá ser chamada de "Cadastro de Funcionários".</p> <p>2 - Na aba "Cadastro de Funcionários" e na área de informações pessoais deverá conter os seguintes campos: nome, data de nascimento, RG, CPF, endereço, cidade, estado, e-mail, telefone, cargo, data de admissão, status, salário base e quantidades de dependentes.</p> <p>3 - Na aba "Cadastro de Funcionários" e no setor informações bancárias deverão conter os seguintes campos: instituição bancária, número da agência, número da conta.</p> <p>4 - O sistema deverá permitir que as informações fornecidas para o cadastro de funcionário sejam editadas, atualizadas ou excluídas sempre que o usuário necessitar.</p>
[RF006]	1 - O sistema deverá possuir uma aba para lançamentos da folha de pagamento, chamado de "Lançamentos da Folha".

	2 - Na aba "Lançamentos da Folha" deverá ser preenchida com as informações recuperadas do cadastro do funcionário e da folha de ponto (emitido em outro sistema).
	3 - Na aba "Lançamentos da Folha" deverá conter os seguintes campos: selecionar o funcionário, adicionar valores de benefícios, valores de descontos, horas trabalhadas, faltas, selecionar o ano das tabelas de desconto do INSS e IR.
[RF007]	1 - O sistema deverá possuir uma aba chamada de "Dados da Folha", que irá conter as informações sobre as tabelas de desconto do INSS e IR 2 - O sistema deve fazer o uso dos dados preenchidos na aba "Lançamentos da Folha" para realizar o cálculo da folha de pagamento.
[RF008]	1 - O sistema deve permitir "Revisar Folha", contendo em lista as folhas de pagamento lançadas, identificadas pelo nome, código do funcionário e data de lançamento. 2 - Cada folha da lista de revisão irá conter um botão para a emissão da folha de pagamento.
[RF009]	1 – A folha de pagamento emitida deverá ser salva no banco de dados.

Fonte: Próprios autores, 2023.

Tabela 5: Requisitos de Sistema - Não Funcionais.

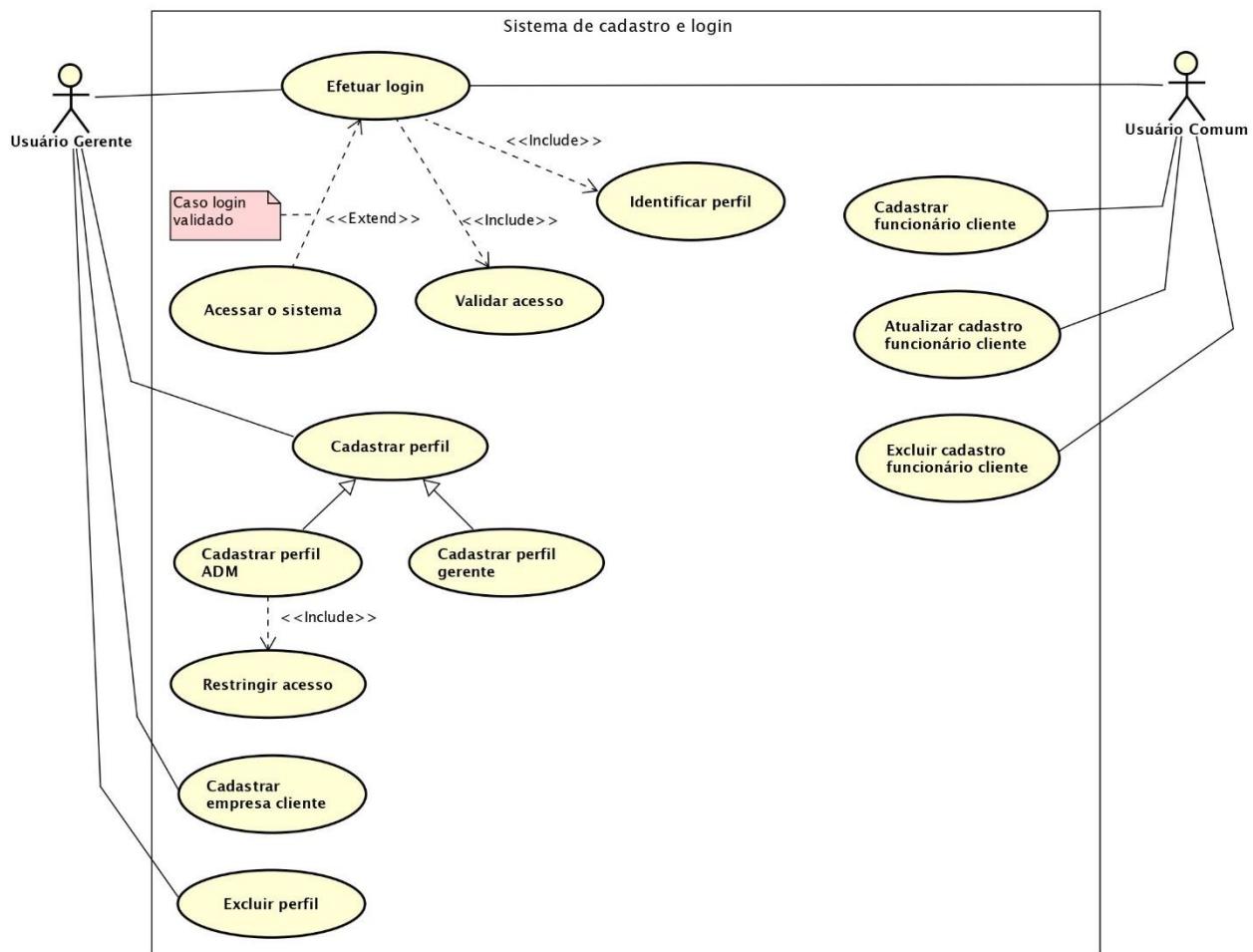
Requisitos de Sistema - Não Funcionais	
[RNF001]	1 - A interface do usuário deverá permitir que o usuário possa acessar as informações e realizar as tarefas necessárias de forma intuitiva e rápida, sem a necessidade de treinamentos complexos. 2 - A usabilidade do sistema deverá ser baseada na norma NBR ISO 9241-11.
[RNF002]	1 - O acesso ao sistema não deverá levar mais do que 15 segundos. 2 - O sistema deverá executar os cálculos com rapidez e em tempo hábil.
[RNF003]	1 - O sistema deverá ser submetido a testes de carga para determinar a capacidade mínima e máxima de folhas e pagamentos emitidas.

[RNF004]	<p>1 - O sistema deverá suportar no mínimo 2 usuários simultaneamente.</p> <p>2 - O Sistema deverá garantir o desempenho mesmo com usuários simultâneos.</p>
[RNF005]	<p>1- O método que será utilizado para cifrar e decifrar as senhas cadastradas será o hashing SHA256.</p>
[RNF006]	<p>1 - A folha de pagamento deverá ser exportada em arquivo PDF em uma pasta no sistema operacional.</p>
[RNF007]	<p>1 - Os cálculos progressivos do INSS deverão estar de acordo com a tabela de contribuição do ano de 2023.</p> <p>2 - Os cálculos do IR deverão estar de acordo com a tabela do Imposto de Renda 2023.</p> <p>3 - Os descontos dos funcionários deverão ser conforme as leis trabalhistas.</p>
[RNF008]	<p>1 - Usará a linguagem C# para o sistema com interface gráfica para desktop.</p> <p>2 - A aplicação Web será desenvolvida com o uso da tecnologia React com a linguagem JavaScript.</p> <p>4 - A aplicação Mobile será desenvolvida utilizando a framework Flutter, com linguagem de programação Dart.</p>
[RNF009]	<p>1 - O banco de dados deverá ser o PostgreSQL hospedado em uma nuvem.</p>

Fonte: Próprios autores, 2023.

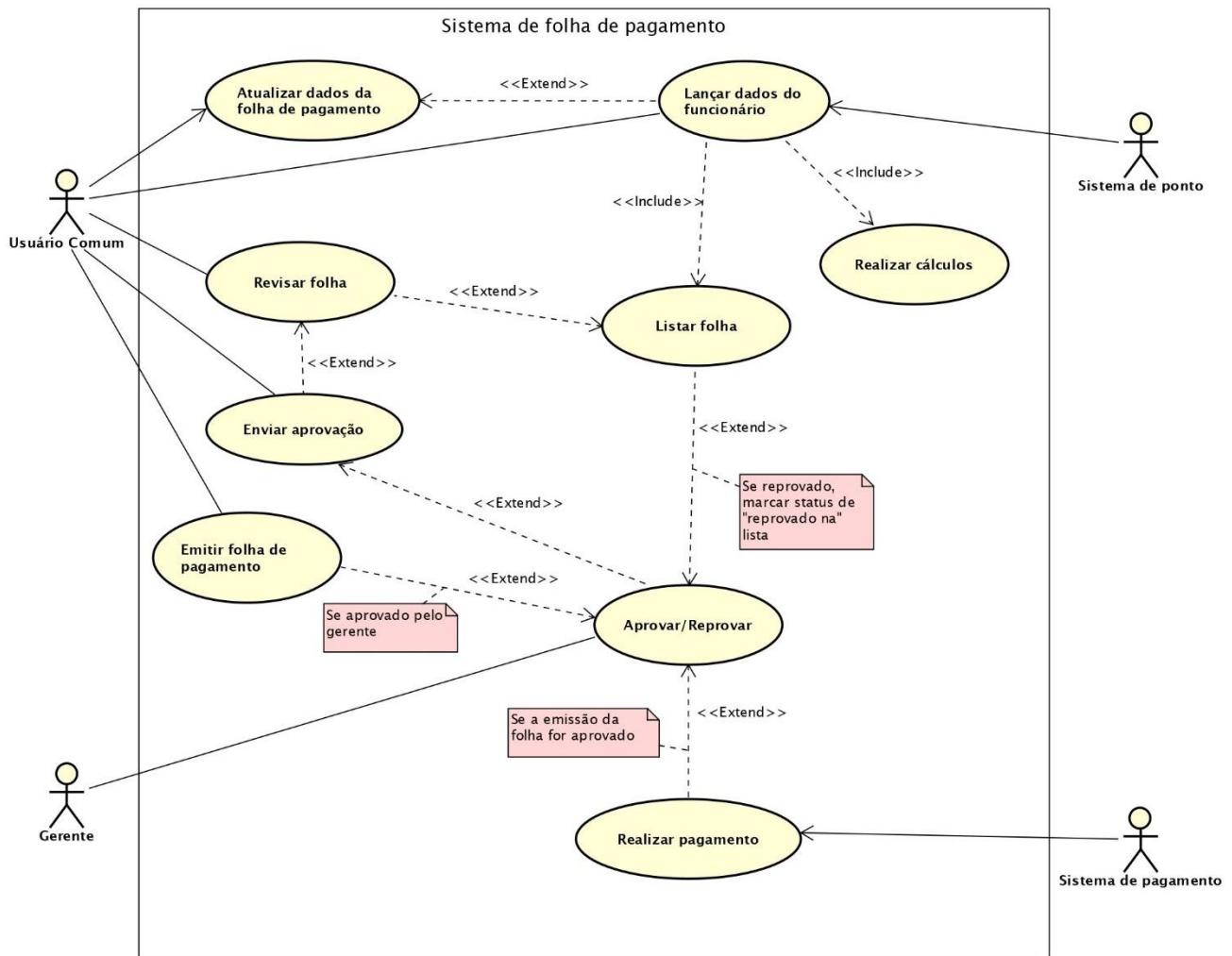
4.2.1.2 Diagramas de caso de uso

Figura 2: Diagrama de caso de uso do sistema de cadastro e login.



Fonte: Próprios autores, 2023.

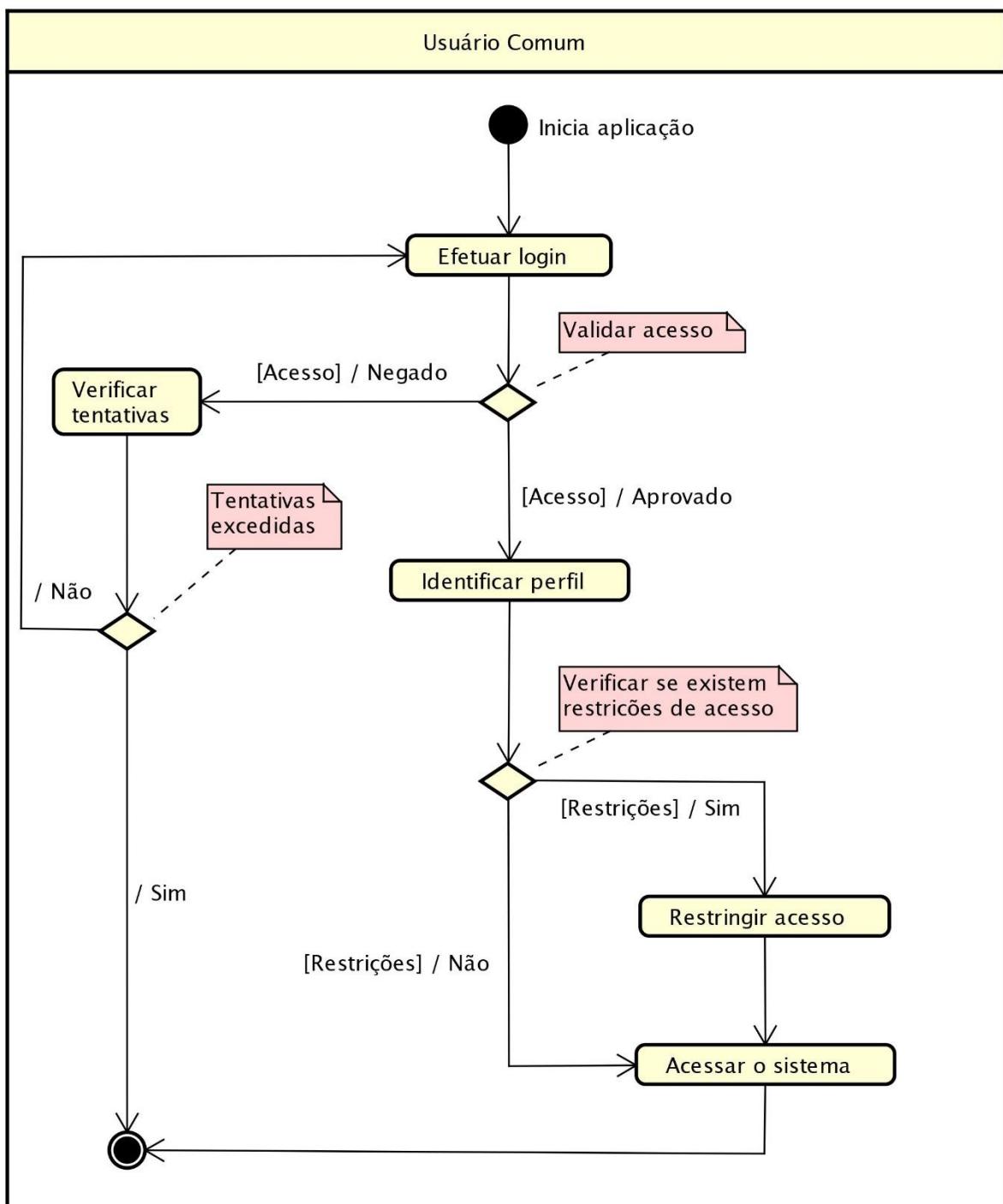
Figura 3: Diagrama de caso de uso do sistema folha de pagamento.



Fonte: Próprios autores, 2023.

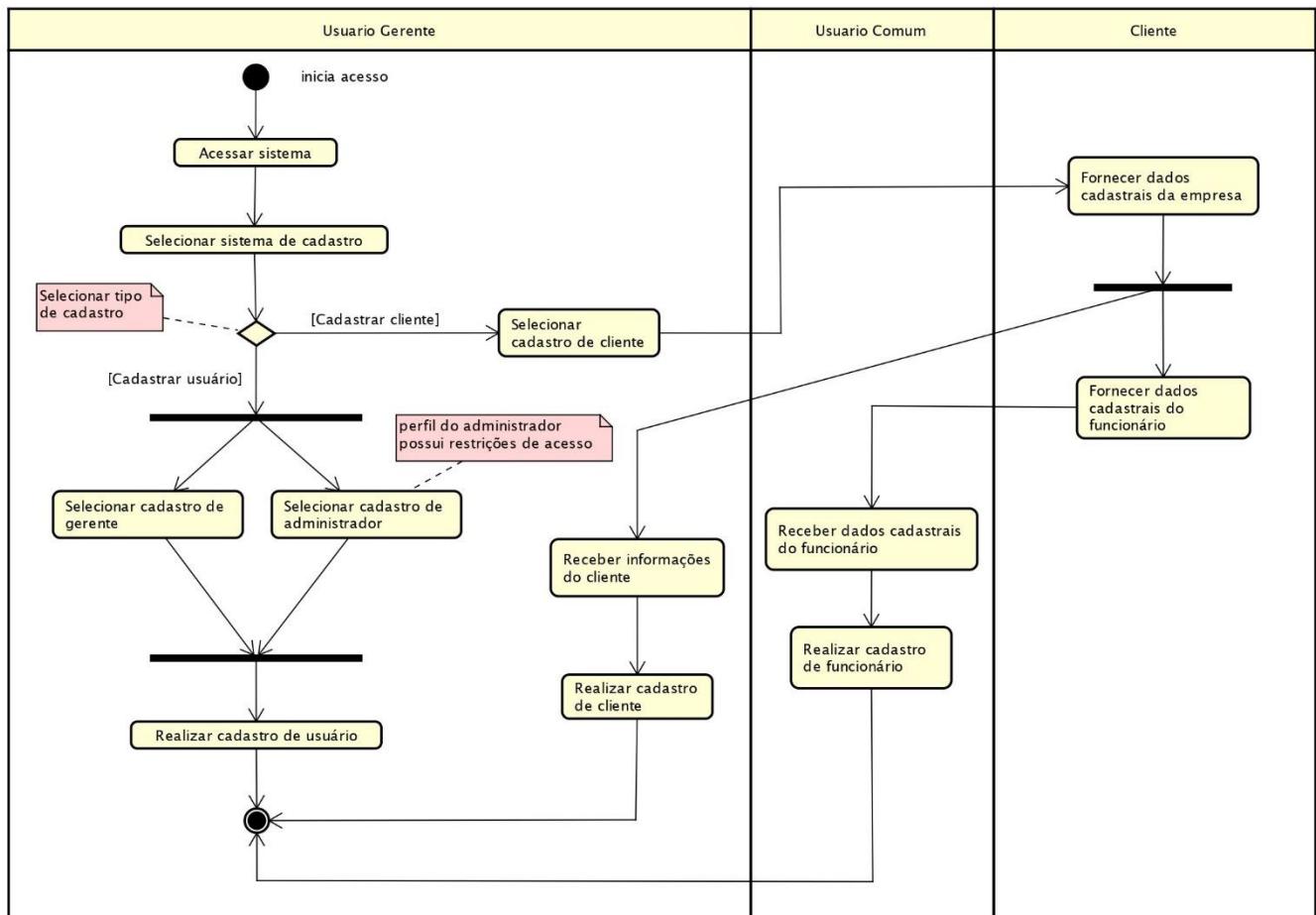
4.2.1.3 Diagramas de atividades

Figura 4: Diagrama de atividades de login.



Fonte: Próprios autores, 2023.

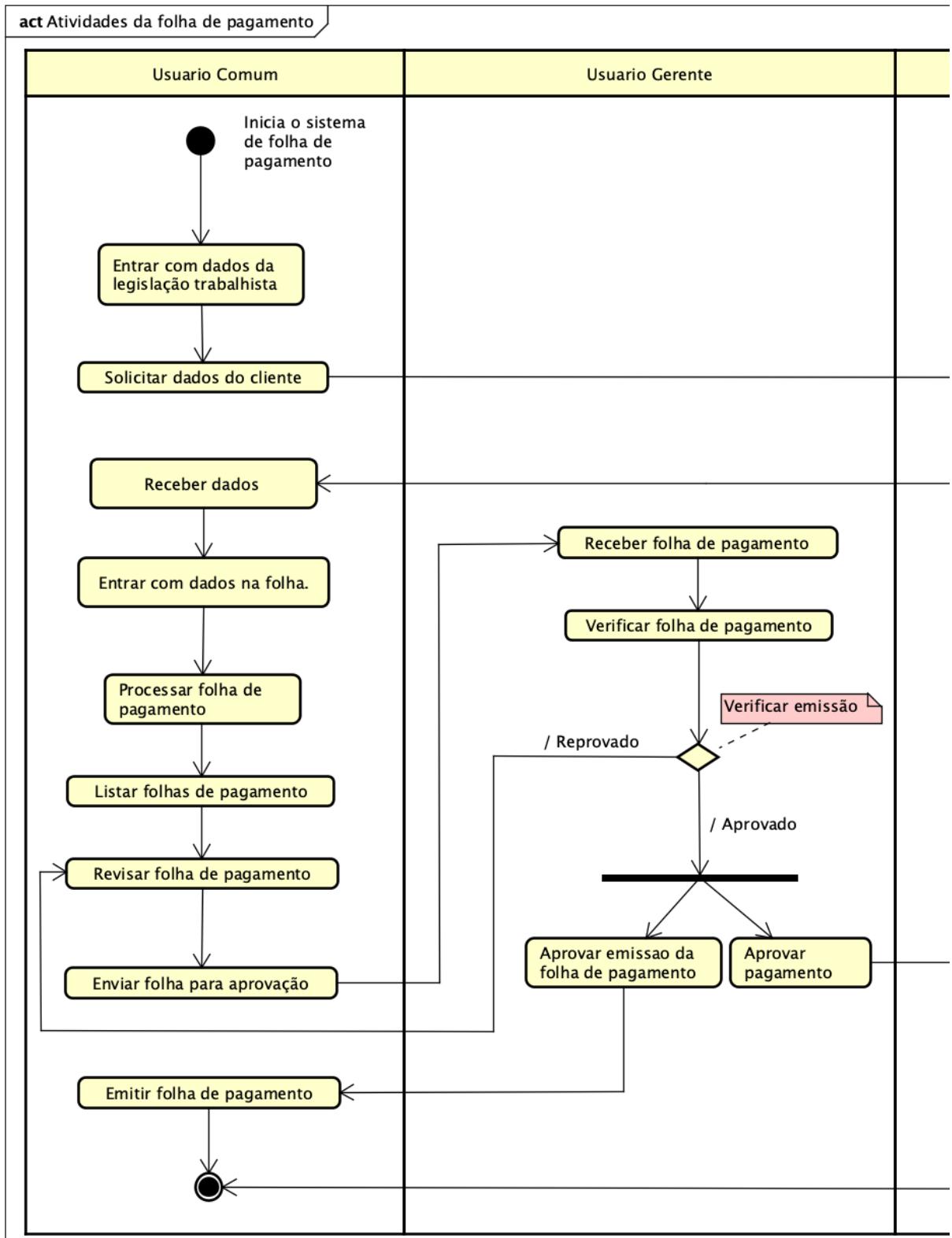
Figura 5: Diagrama de atividades de cadastro.



Fonte: Próprios autores, 2023.

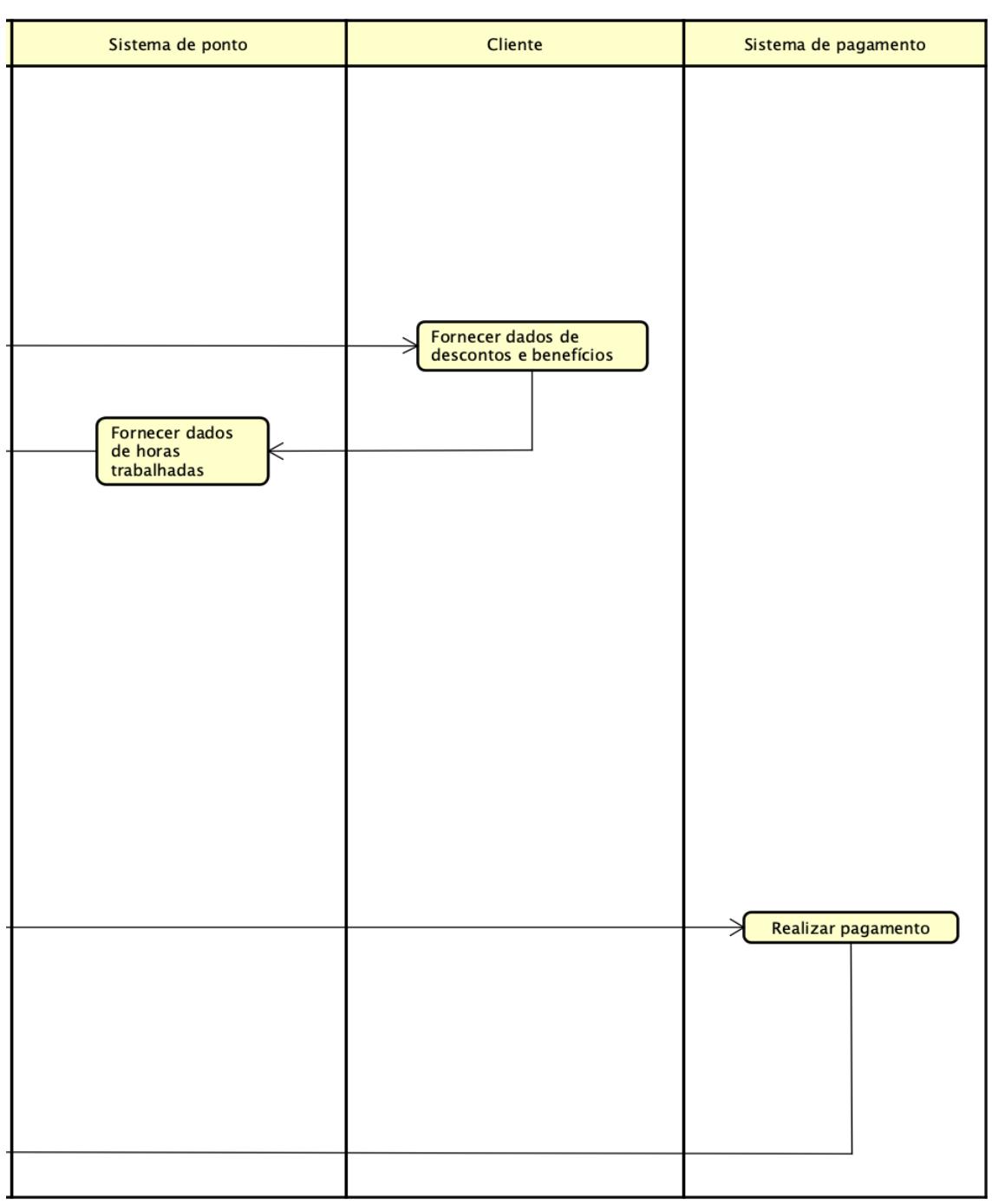
Os diagramas a seguir serão exibidos em partes para melhor visualização e em seguida será exibido o mesmo de forma completa.

Figura 6: Diagrama de atividades da folha de pagamento recorte 1.



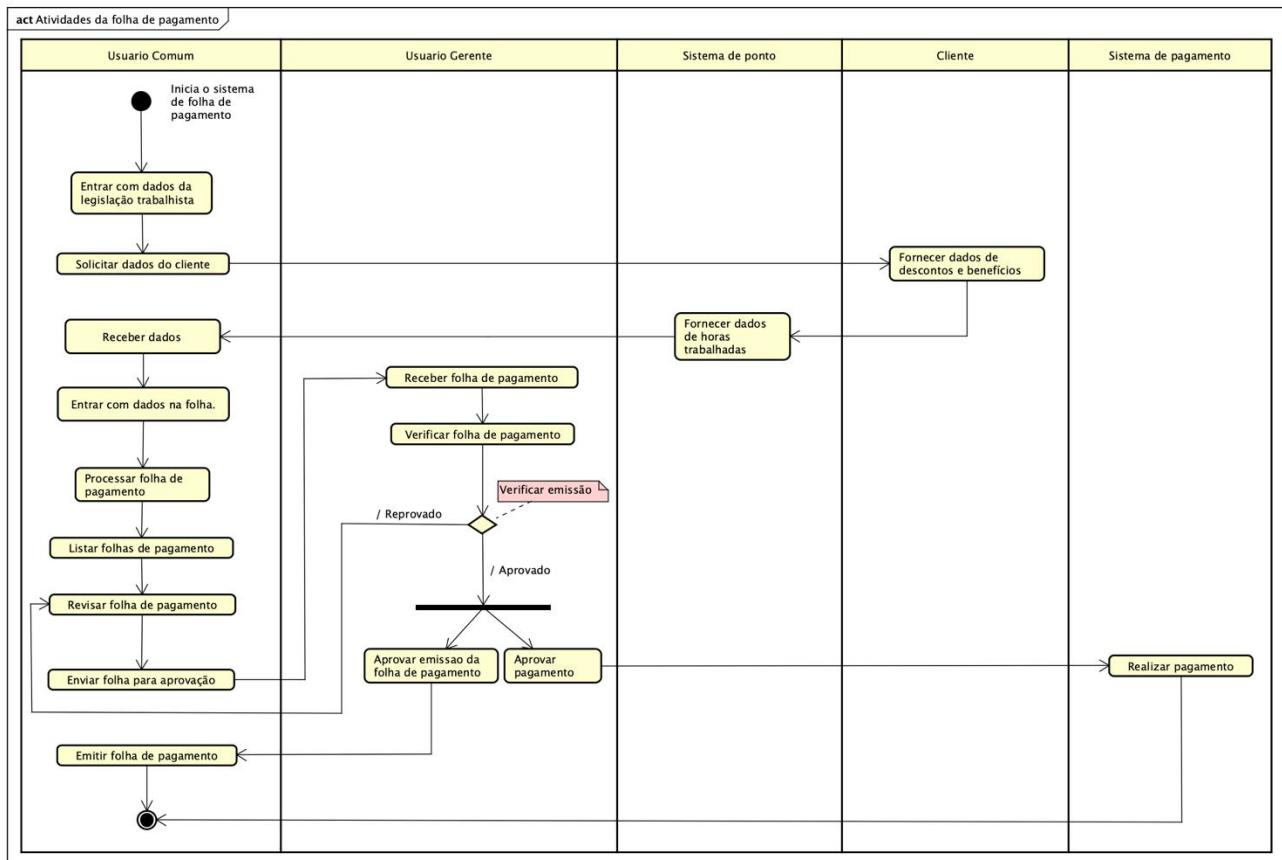
Fonte: Próprios autores, 2023.

Figura 7: Diagrama de atividades da folha de pagamento recorte 2.



Fonte: Próprios autores, 2023.

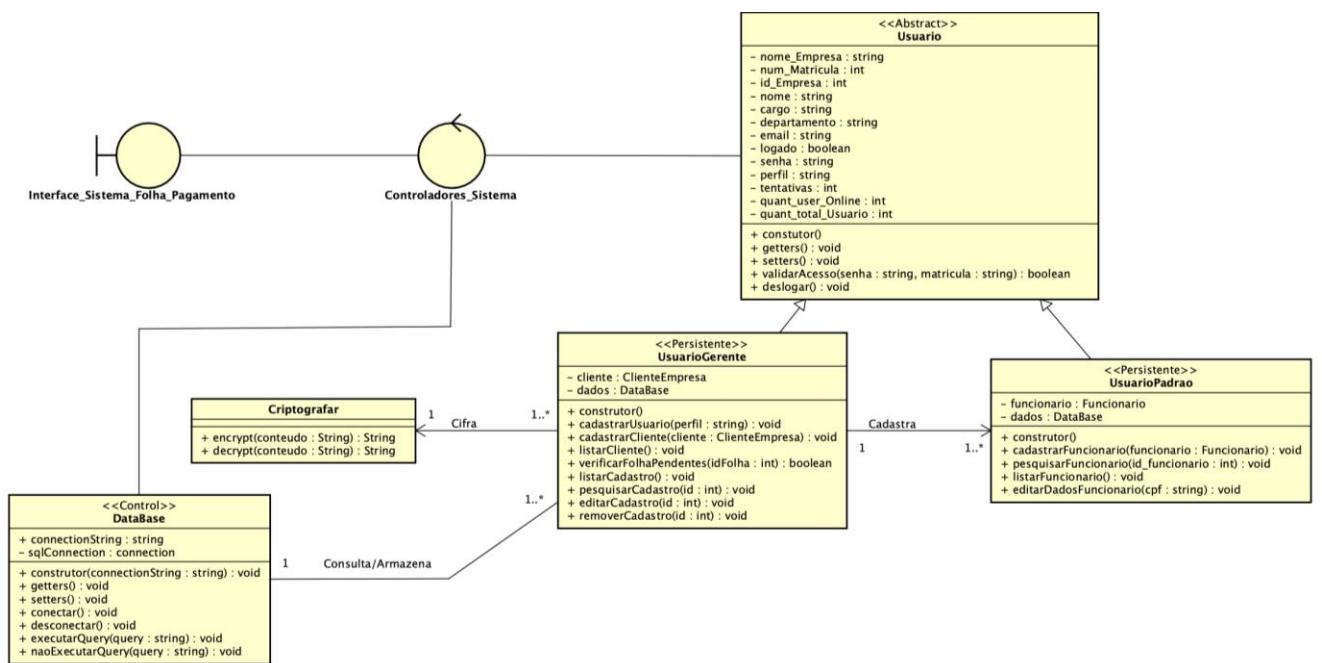
Figura 8: Diagrama de atividades da folha de pagamento completo.



Fonte: Próprios autores, 2023.

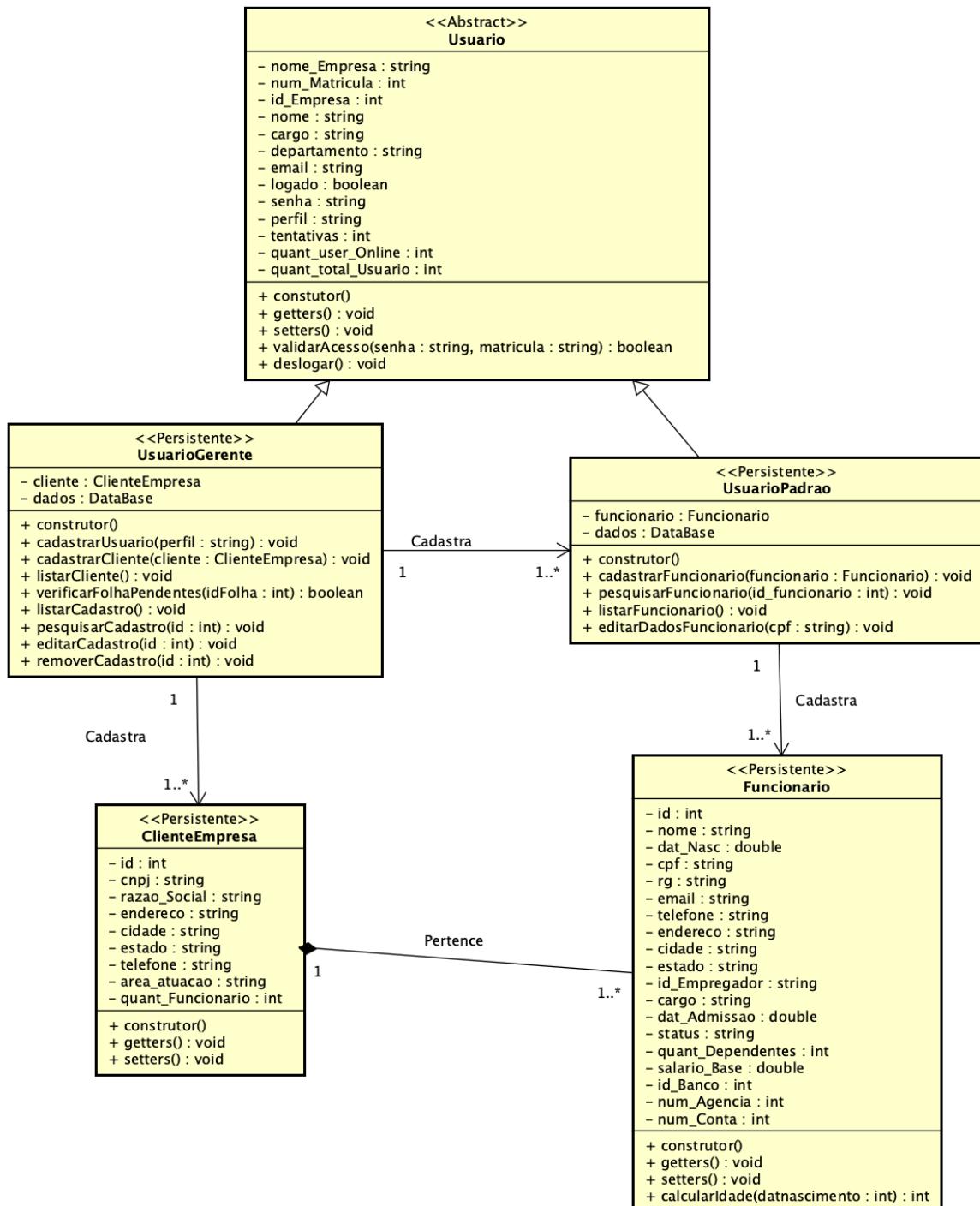
4.2.1.4 Diagrama de classes

Figura 9: Diagrama de classes parte 1/3.



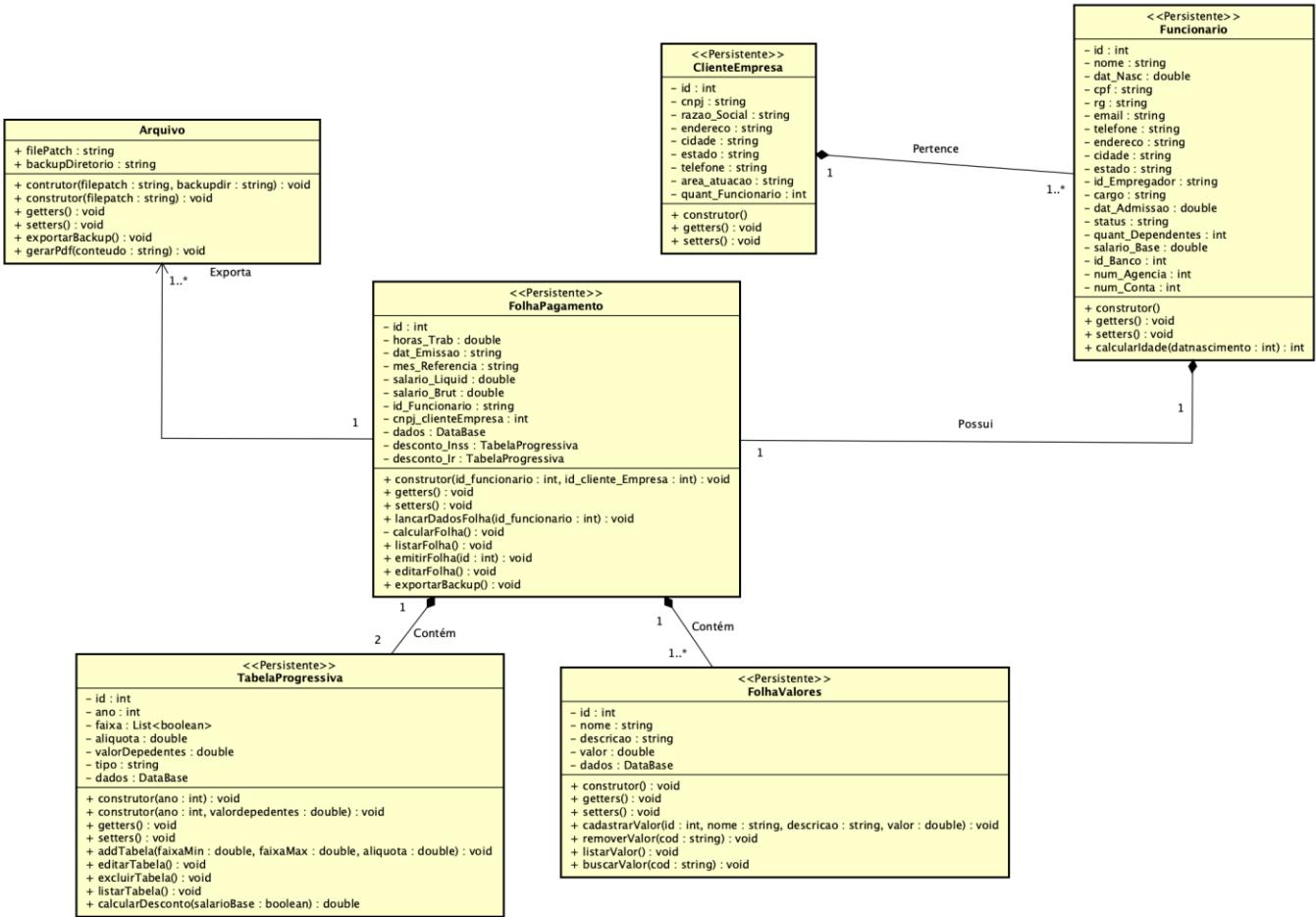
Fonte: Próprios autores, 2023.

Figura 10: Diagrama de classes parte 2/3.



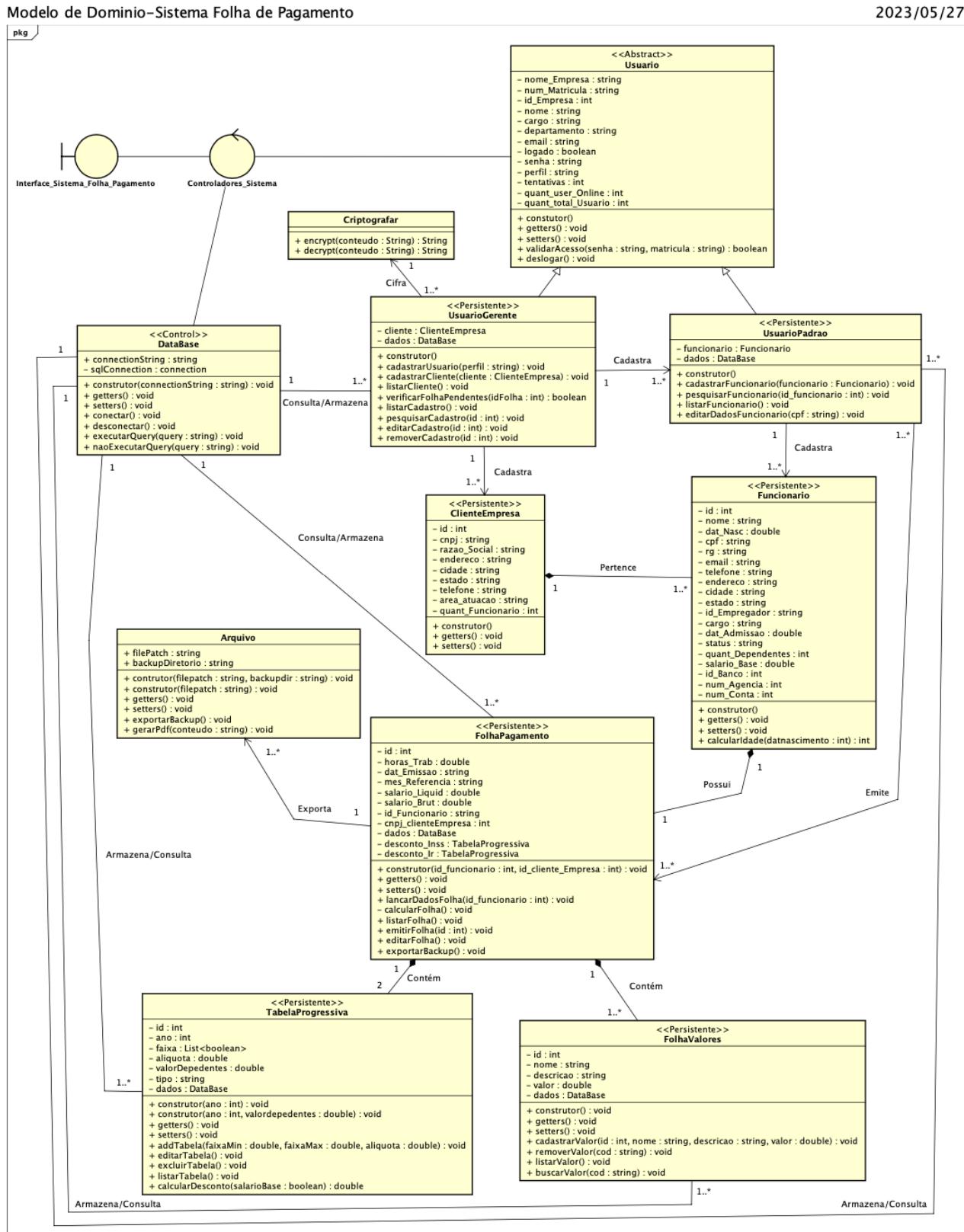
Fonte: Próprios autores, 2023.

Figura 11: Diagrama de classes parte 3/3.



Fonte: Próprios autores, 2023.

Figura 12: Diagrama de classes completo.

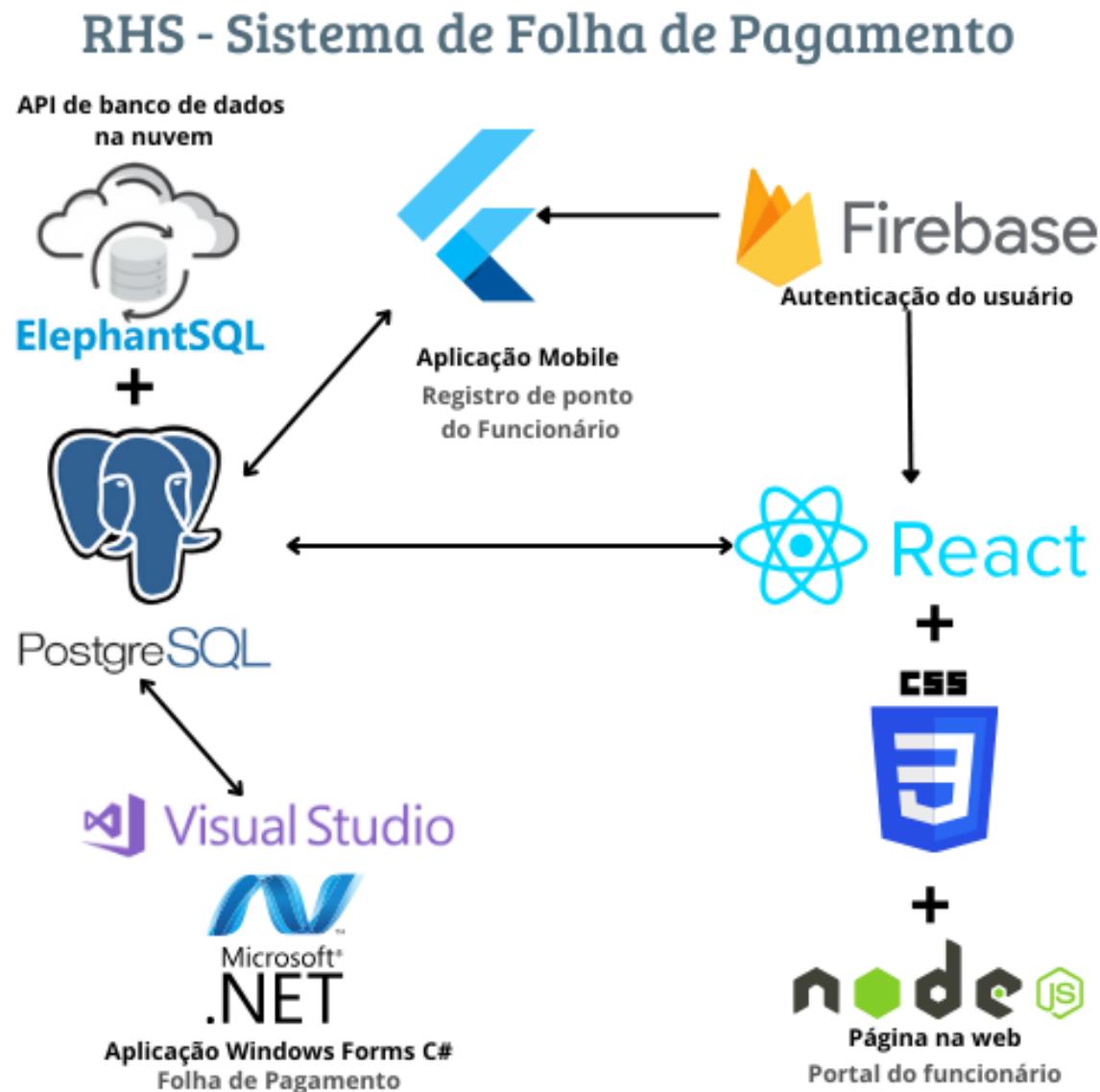


Fonte: Próprios autores, 2023.

4.3 Arquitetura do sistema

Após levantar todos os requisitos. Começamos a trabalhar na arquitetura do sistema. Onde foi criado um organismo que possui 3 aplicações: folha de pagamento, registro de ponto e um portal onde o funcionário pode consultar suas informações na web. Todos conectados e o único banco de dados para dados persistentes que está na nuvem. Além disso, vale ressaltar que autenticação do usuário feito na plataforma web e na plataforma mobile foi feita através do Firebase. Que é um outro tipo de serviço de banco de dados fornecido pela Google.

Figura 13: Arquitetura da integração dos sistemas.

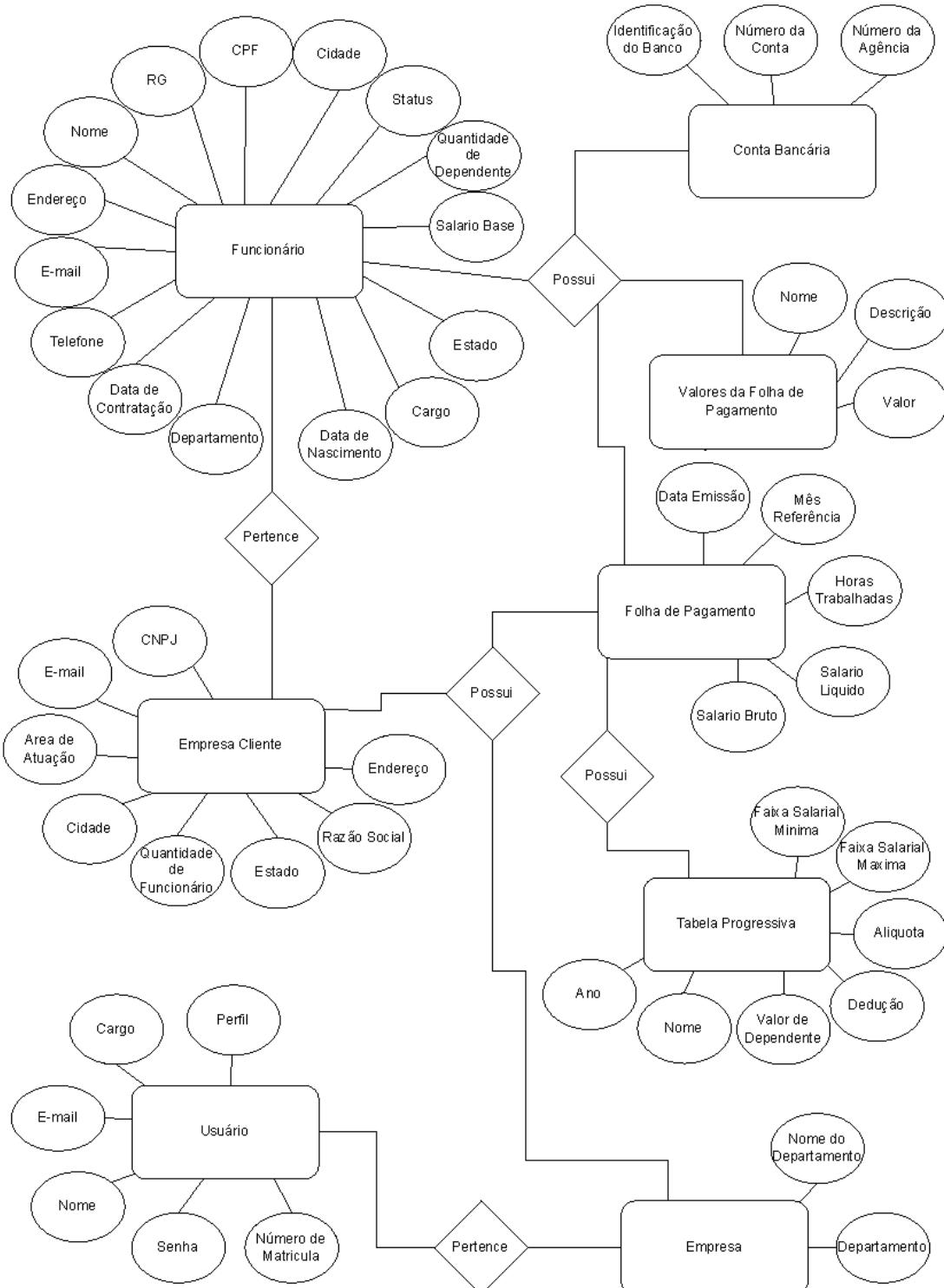


Fonte: Próprios autores, 2023.

4.4 Banco de dados

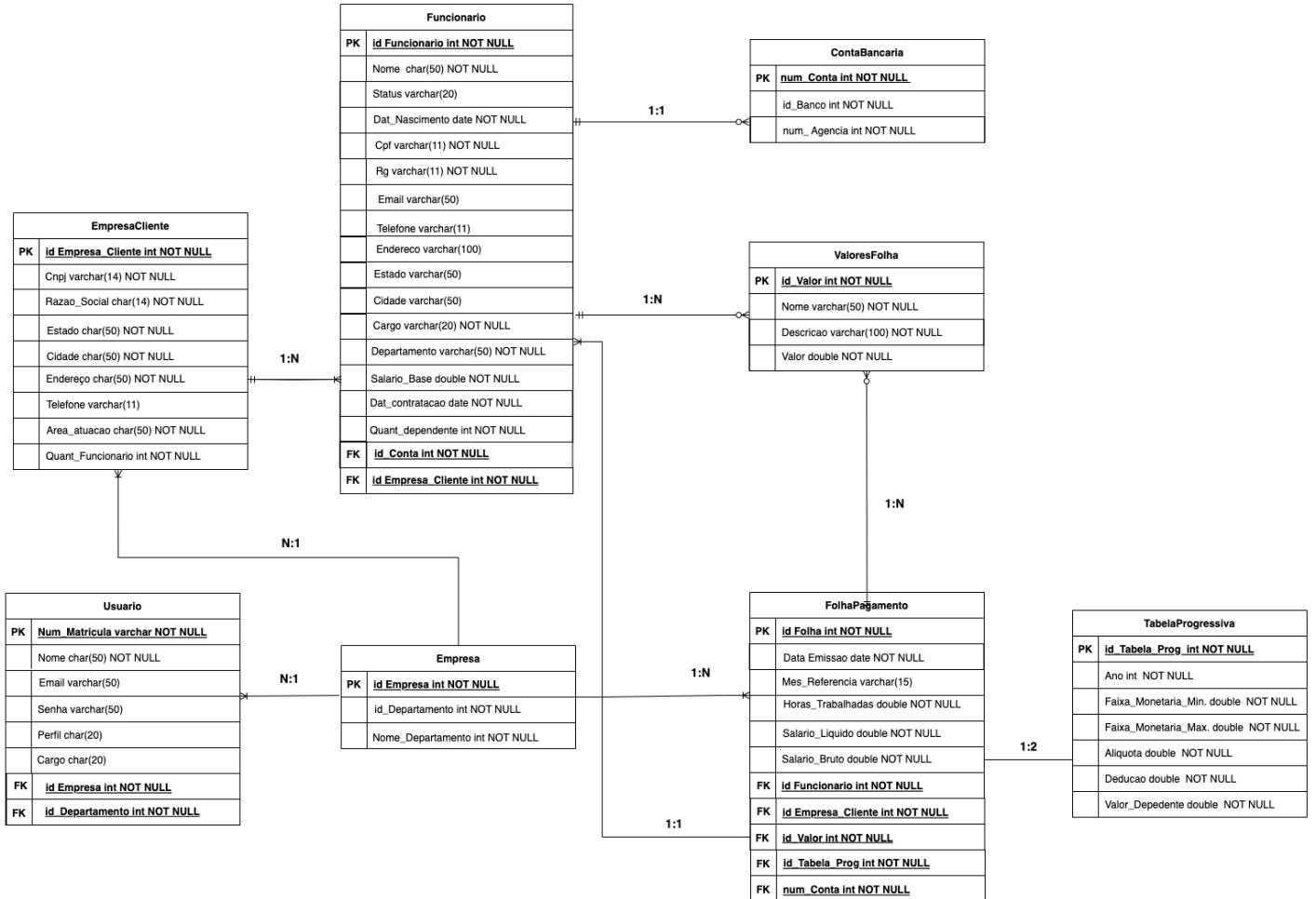
4.4.1 Modelo Entidade-Relacionamento

Figura 14: MER conceitual do banco de dados.



Fonte: Próprios autores, 2023.

Figura 15: MER do banco de dados.



Fonte: Próprios autores, 2023.

4.4.2 PostgreSQL

Para o desenvolvimento deste projeto utilizamos o banco de dados PostgreSQL e o SGBD pgAdmin 4 com a API ElephantSQL para disponibilizar o armazenamento do banco de dados em nuvem, o objetivo foi tornar os dados compartilháveis, compatíveis e acessíveis entre as 3 aplicações que compõe o sistema de folha de pagamento. A seguir abordaremos alguns conceitos importantes sobre essas tecnologias.

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto, e o ElephantSQL é um serviço de banco de dados PostgreSQL na nuvem. A interação entre o PostgreSQL e o ElephantSQL é simples:

- Provisionamento: Primeiro, criamos uma conta no ElephantSQL e provisiona um banco de dados PostgreSQL. Isso envolve selecionar a quantidade de recursos desejados e configurar as credenciais de acesso;
- Conexão: Depois de provisionar o banco de dados, conectamos a ele usando as credenciais fornecidas;
- Gerenciamento: O ElephantSQL oferece ferramentas para monitorar e gerenciar seu banco de dados PostgreSQL na nuvem. Isso inclui recursos como backups, escalabilidade e outras opções de gerenciamento;
- Aplicação: Com o banco de dados PostgreSQL hospedado no ElephantSQL, integramos o backend de nossos 3 aplicativos.

Figura 16: SGBD pgAdmin 4.

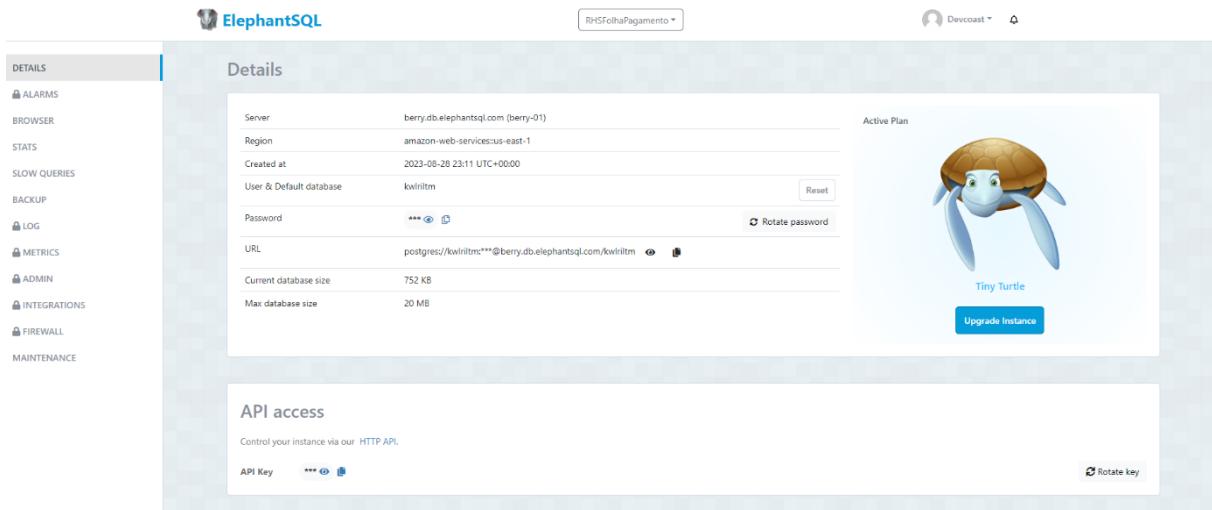
The screenshot shows the pgAdmin 4 interface. The top menu bar includes File, Object, Tools, and Help. The title bar shows the connection details: 'kwlrltm/kwlrltm@B_SERVER_RHS_FOLHA_DE_PAGAMENTO'. The left sidebar, 'Object Explorer', lists several schemas: 'kwlrltm', 'RHS' (which is expanded to show Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables), and 'kwlqldims'. The main workspace contains a 'Query' tab with the SQL query 'SELECT * FROM "RHS"."tb_funcionario";' and a 'Data Output' tab displaying the results:

	Id_funcionario	nome	status	dat_nascimento	cpf	rg	email	telefone
1	30	Luz Carlos de Araújo Machado	Ativo	01/05/1994	424.177.188-12	45.789.123-1	luzsaw@gmail.com	(13) 99709-72
2	31	Kaio Alves da Silva	Ativo	12/04/1998	345.122.772-11	45.897.007-3	Kaio@gmail.com	(13) 99845-44

At the bottom of the Data Output tab, it says 'Total rows: 2 of 2 Query complete 00:00:01.128 Ln 1, Col 1'.

Fonte: Próprios autores, 2023.

Figura 17: Tela inicial ElephantSQL.



Fonte: Próprios autores, 2023.

4.4.3 Scripts do Banco de dados

Figura 18: Script Banco de dados: criação da tabela conta bancária.

```
CREATE TABLE IF NOT EXISTS "RHS".tb_conta_bancaria
(
    num_conta character varying(20) COLLATE pg_catalog."default" NOT NULL,
    num_agencia character varying(20) COLLATE pg_catalog."default" NOT NULL,
    nome_banco character varying(20) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT tb_conta_bancaria_pkey PRIMARY KEY (num_conta)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_conta_bancaria
OWNER to kwlrltm;
```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 19: Script Banco de dados: criação da tabela empresa.

```
CREATE TABLE IF NOT EXISTS "RHS".tb_empresa
(
    id integer NOT NULL DEFAULT nextval('"RHS".tb_empresa_id_seq'::regclass),
    cnpj character varying(20) COLLATE pg_catalog."default" NOT NULL,
    razao_social character varying(30) COLLATE pg_catalog."default" NOT NULL,
    estado character varying(30) COLLATE pg_catalog."default" NOT NULL,
    cidade character varying(30) COLLATE pg_catalog."default" NOT NULL,
    endereco character varying(50) COLLATE pg_catalog."default" NOT NULL,
    telefone character varying(15) COLLATE pg_catalog."default" NOT NULL,
    area_atuacao character varying(50) COLLATE pg_catalog."default" NOT NULL,
    quant_funcionario integer NOT NULL,
    CONSTRAINT tb_empresa_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_empresa
OWNER to kwlrltm;
```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 20: Script Banco de dados: criação da tabela folha de pagamento.

```

CREATE TABLE IF NOT EXISTS "RHS".tb_folha_pagamento
(
    id integer NOT NULL DEFAULT nextval('"RHS".tb_folha_pagamento_id_seq'::regclass),
    id_funcionario integer NOT NULL,
    nome_arquivo character varying(20) COLLATE pg_catalog."default" NOT NULL,
    data_emissao character varying(20) COLLATE pg_catalog."default" NOT NULL,
    arquivo bytea NOT NULL,
    CONSTRAINT tb_folha_pagamento_pkey PRIMARY KEY (id),
    CONSTRAINT id_funcionario FOREIGN KEY (id_funcionario)
        REFERENCES "RHS".tb_funcionario (id_funcionario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_folha_pagamento
    OWNER to kwlrlt;

```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 21: Script Banco de dados: criação da tabela registro de ponto.

```

CREATE TABLE IF NOT EXISTS "RHS".tb_registro_ponto
(
    id integer NOT NULL DEFAULT nextval('"RHS".registro_ponto_id_seq'::regclass),
    id_funcionario integer NOT NULL,
    data character varying COLLATE pg_catalog."default" NOT NULL,
    hora character varying COLLATE pg_catalog."default" NOT NULL,
    tipo character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT registro_ponto_pkey PRIMARY KEY (id),
    CONSTRAINT id_funcionario FOREIGN KEY (id_funcionario)
        REFERENCES "RHS".tb_funcionario (id_funcionario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_registro_ponto
    OWNER to kwlrlt;

```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 22: Script Banco de dados: criação da tabela funcionário.

```

CREATE TABLE IF NOT EXISTS "RHS".tb_funcionario
(
    id_funcionario integer NOT NULL DEFAULT nextval('"RHS".tb_funcionario_id_seq'::regclass),
    nome character(50) COLLATE pg_catalog."default" NOT NULL,
    status character varying(20) COLLATE pg_catalog."default" NOT NULL,
    dat_nascimento character varying(10) COLLATE pg_catalog."default" NOT NULL,
    cpf character varying(20) COLLATE pg_catalog."default" NOT NULL,
    rg character varying(20) COLLATE pg_catalog."default" NOT NULL,
    email character varying(50) COLLATE pg_catalog."default" NOT NULL,
    telefone character varying(20) COLLATE pg_catalog."default" NOT NULL,
    endereco character varying(100) COLLATE pg_catalog."default" NOT NULL,
    estado character varying(50) COLLATE pg_catalog."default" NOT NULL,
    cidade character varying(50) COLLATE pg_catalog."default" NOT NULL,
    cargo character varying(50) COLLATE pg_catalog."default" NOT NULL,
    departamento character varying(50) COLLATE pg_catalog."default" NOT NULL,
    salario_base numeric(18,2) NOT NULL,
    quant_dependente integer NOT NULL,
    dat_contratacao character varying(10) COLLATE pg_catalog."default" NOT NULL,
    carga_horaria numeric(5,2) NOT NULL,
    id_empresa integer NOT NULL,
    num_conta character varying(20) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT tb_funcionario_pkey PRIMARY KEY (id_funcionario),
    CONSTRAINT id_empresa FOREIGN KEY (id_empresa)
        REFERENCES "RHS".tb_empresa (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT num_conta FOREIGN KEY (num_conta)
        REFERENCES "RHS".tb Conta_bancaria (num_conta) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 23: Script Banco de dados: criação da tabela salário família.

```

CREATE TABLE IF NOT EXISTS "RHS".tb_salario_familia
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    mes_ano_vigente character varying(50) COLLATE pg_catalog."default" NOT NULL,
    faixa_monetaria_min numeric(10,2) NOT NULL,
    faixa_monetaria_max numeric(10,2) NOT NULL,
    valor_unit numeric(5,2) NOT NULL,
    CONSTRAINT salario_familia_pkey PRIMARY KEY (id)
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_salario_familia
OWNER TO kwlrltmm;

```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 24: Script Banco de dados: criação da tabela progressiva inss e irrf.

```

CREATE TABLE IF NOT EXISTS "RHS".tb_tab_progressiva
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    tipo character varying(20) COLLATE pg_catalog."default" NOT NULL,
    mes_ano_vigente character varying(20) COLLATE pg_catalog."default" NOT NULL,
    faixa_monetaria_min numeric(10,2) NOT NULL,
    faixa_monetaria_max numeric(10,2),
    aliquota numeric(5,4) NOT NULL,
    deducao numeric(10,2),
    CONSTRAINT tb_tab_progressiva_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_tab_progressiva
OWNER to kwlrlt;

```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 25: Script Banco de dados: criação da tabela usuário.

```

CREATE TABLE IF NOT EXISTS "RHS".tb_usuario
(
    id integer NOT NULL DEFAULT nextval('"RHS".tb_usuario_id_seq'::regclass),
    nome character varying(100) COLLATE pg_catalog."default" NOT NULL,
    tipo character varying(20) COLLATE pg_catalog."default" NOT NULL,
    login character varying(20) COLLATE pg_catalog."default" NOT NULL,
    senha character varying(256) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT tb_usuario_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_usuario
OWNER to kwlrlt;

```

Fonte: PostgreSQL, pgAdmin 4, 2023.

Figura 26: Script Banco de dados: criação da tabela dependentes.

```

CREATE TABLE IF NOT EXISTS "RHS".tb_valor_dependente
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    mes_ano_vigente character varying(50) COLLATE pg_catalog."default" NOT NULL,
    valor numeric(5,2),
    CONSTRAINT tb_valor_dependente_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "RHS".tb_valor_dependente
OWNER to kwlrlt;

```

Fonte: PostgreSQL, pgAdmin 4, 2023.

4.4.4 Classes Banco de dados

A seguir algumas classes no backend que usamos para integrar nosso banco de dados.

Em C# no Visual Studio no sistema folha de pagamento:

Figura 27: Classe BancoDados desenvolvido em C#.

```

21 RHS_Folha_de_Pagamento
22 21 referências
23 internal class BancoDados
24 {
25     private static string servidor = "berry.db.elephantsql.com";
26     private static string bancoDados = "kwlrilm";
27     private static string usuario = "kwlrilm";
28     private static string senha = "X5vdShkoeld8aCwESg7DQsJ7ay6ADyEz5";
29     private static string porta = "5432";
30     private static NpgsqlConnection conexao;
31     3 referências
32     public bool estarConectado { get; set; }
33     1 referência
34     public static string StrConexao...
35     10 referências
36     public BancoDados()...
37     10 referências
38     public void Conectar()...
39     17 referências
40     public void Desconectar()...
41     //Funcao que deleta um registro de uma tabela, SEM CHAVES ESTRANGEIRAS, usando o id
42     3 referências
43     public bool DeletarRegistro(string nomeTabela, string nomeColuna, int id)...
44     1 referência
45     public bool DeletarRegistro(string nomeTabela, string nomeColuna, string id)...
46     //Função que executa uma query e preenche os valores de uma coluna em um comboBox
47     2 referências
48     public void PreencherComboBox(string query, string coluna, ComboBox comboBox)...
49     //Função que executa uma query e retorna apenas um valor string de uma consulta
50     7 referências
51     public string ObterValor(string query)...
52     //Função que executa uma query e retorna os valores de colunas de uma tabela em uma matriz bidimensional
53     1 referência
54     public string[,] ObterValoresDaTabelaProgressiva(string tipo, string ano, List<string> nomeColuna, string nomeTabela)...
55     //Função que executa uma query e retorna valores em um listView
56     1 referência
57     public void ExecutarQueryListView(string query, ListView lista, List<string> nomeColunas)...
58     /*A função InserirDados recebe o nome da tabela, um array com os nomes das colunas e um array com os valores correspondentes para cada coluna.*/
59     Ela monta a query de inserção dinamicamente, substituindo os nomes das colunas e os valores pelos parâmetros adequados.*/
60     4 referências
61     public void InserirDados(string tabelaNomes, string[] colunaNomes, object[] valores)...
62     // Método para obter o tipo de dado NpgsqlDbType com base no tipo do valor
63     1 referência
64     public void inserirFolhaPagamento(string caminhoArquivo, int id_funcionario, string dataemissao)...
65     1 referência
66     public string buscarCredenciaisUser(string user, string senha)...
67     // Método para calcular o hash da senha usando SHA256
68     1 referência
69     private string CalcularHashSenha(string senha)...
70     1 referência
71     public int retornarIdEmpresa(string nomeTabela, string nomeColuna, string valorColuna)...
72 }
73 
```

Fonte: Próprios autores, 2023.

Em Dart no Flutter no sistema registro de ponto:

Figura 28: Classe BancoDados desenvolvido em Dart, parte 1/2.

```

1 > pages > banco_bados.dart
2   import 'package:flutter/material.dart';
3   import 'package:postgres/postgres.dart';
4
5   class PostgreSQLService {
6     late PostgreSQLConnection _connection;
7
8     PostgreSQLService({
9       required String host,
10      required int port,
11      required String databaseName,
12      required String username,
13      required String password,
14    }) {
15       _connection = PostgreSQLConnection(
16         host,
17         port,
18         databaseName,
19         username: username,
20         password: password,
21       );
22     }
23
24     Future<void> openConnection() async {
25       if (_connection.isClosed) {
26         await _connection.open();
27     }
28   }
29 }
```

Fonte: Próprios autores, 2023.

Figura 29: Classe BancoDados desenvolvido em Dart, parte 2/2.

```

29 |     Future<int> getIdFuncionario(String email) async {
30 |         await openConnection();
31 |         try {
32 |             final results = await _connection.query(
33 |                 'SELECT id_funcionario FROM "RHS"."tb_funcionario" WHERE email = @email',
34 |                 substitutionValues: {'email': email},
35 |             );
36 |
37 |             if (results.isNotEmpty) {
38 |                 return results[0][0] as int?;
39 |             } else {
40 |                 return null; // Email não encontrado
41 |             }
42 |         } finally {}
43 |     }
44 |
45 |     Future<void> inserirRegistroPonto(
46 |         int id, String data, TimeOfDay hora, String tipo) async {
47 |         await openConnection();
48 |         try {
49 |             final horaFormatada =
50 |                 '${hora.hour.toString().padLeft(2, '0')}:${hora.minute.toString().padLeft(2, '0')}';
51 |
52 |             await _connection.query(
53 |                 'INSERT INTO "RHS"."tb_registro_ponto" (id_funcionario, data, hora, tipo) '
54 |                 'VALUES (@id_funcionario, @data, @hora, @tipo)',
55 |                 substitutionValues: {
56 |                     'id_funcionario': id,
57 |                     'data': data,
58 |                     'hora': horaFormatada,
59 |                     'tipo': tipo,
60 |                 },
61 |             );
62 |         } finally {}
63 |     }
64 |
65 |     Future<void> dispose() async {
66 |         await _connection.close();
67 |     }
68 | }
69 |

```

Fonte: Próprios autores, 2023.

Figura 30: Scripts Banco de dados executados em um servidor.js, parte 1/3.

```

js server.js > ...
1
2  const express = require('express');
3  const { Pool } = require('pg');
4  const cors = require('cors');
5  const app = express();
6  const port = 3333;
7  const dbConfig = {
8      host: 'berry.db.elephantsql.com',
9      database: 'kwlrilm',
10     port: '5432',
11     user: 'kwlrilm',
12     password: 'XsvdSkoe1d8aCwE5g7DQsJ7ay6ADyEz5',
13 };
14
15 const pgp = require('pg-promise')();
16 const db = pgp(dbConfig);
17
18 app.use(cors());
19
20 app.get('/api/obter-id-usuario', async (req, res) => {
21     const { email } = req.query;
22
23     if (!email) {
24         return res.status(400).json({ error: 'O email não foi fornecido como parâmetro' });
25     }
26
27     try {
28         const result = await db.query('SELECT id_funcionario FROM "RHS"."tb_funcionario" WHERE email = $1', [email]);
29         if (result && result.length > 0) {
30             const userId = result[0].id_funcionario;
31             return res.json({ userId });
32         } else {
33             return res.status(404).json({ error: 'Usuário não encontrado' });
34         }
35     } catch (error) {
36         console.error('Erro ao consultar o banco de dados', error);
37         return res.status(500).json({ error: 'Erro ao consultar o banco de dados' });
38     }
39 });
40
41 app.get('/api/pontos/:userId', async (req, res) => {
42     const { userId } = req.params;

```

Fonte: Próprios autores, 2023.

Figura 31: Scripts Banco de dados executados em um servidor.js, parte 2/3.

```

JS serverjs > ...
40
41 app.get('/api/pontos/:userId', async (req, res) => {
42   const { userId } = req.params;
43
44   if (!userId) {
45     return res.status(400).json({ error: 'O ID do usuário não foi fornecido como parâmetro' });
46   }
47
48   try {
49     const result = await db.query('SELECT * FROM "RHS"."tb_registro_ponto" WHERE id_funcionario = $1', [userId]);
50     if (result && result.length > 0) {
51       return res.json(result);
52     } else {
53       return res.status(404).json({ error: 'Folha de pagamento não encontrada' });
54     }
55   } catch (error) {
56     console.error('Erro ao consultar o banco de dados', error);
57     return res.status(500).json({ error: 'Erro ao consultar o banco de dados' });
58   }
59 });
60
61 app.get('/api/folha-pagamento/:userId', async (req, res) => {
62   const { userId } = req.params;
63
64   if (!userId) {
65     return res.status(400).json({ error: 'O ID da folha de pagamento não foi fornecido como parâmetro' });
66   }
67
68   try {
69     const result = await db.query('SELECT * FROM "RHS"."tb_folha_pagamento" WHERE id_funcionario = $1', [userId]);
70     if (result && result.length > 0) {
71       return res.json(result);
72     } else {
73       return res.status(404).json({ error: 'Folha de pagamento não encontrada' });
74     }
75   } catch (error) {
76     console.error('Erro ao consultar o banco de dados', error);
77     return res.status(500).json({ error: 'Erro ao consultar o banco de dados' });
78   }
79 });
80
81 app.get('/api/download-folha-pagamento/:userId', async (req, res) => {
82   const { userId } = req.params;

```

Fonte: Próprios autores, 2023.

Figura 32: Scripts Banco de dados executados em um servidor.js, parte 3/3.

```

81 app.get('/api/download-folha-pagamento/:userId', async (req, res) => {
82   const { userId } = req.params;
83
84   if (!userId) {
85     return res.status(400).json({ error: 'O ID da folha de pagamento não foi fornecido como parâmetro' });
86   }
87
88   try {
89     const result = await db.query('SELECT arquivo, nome_arquivo FROM "RHS"."tb_folha_pagamento" WHERE id_funcionario = $1', [userId]);
90     if (result && result.length > 0) {
91       const { arquivo, nome_arquivo } = result[0];
92
93       // cabeçalho de resposta para o tipo de conteúdo apropriado (PDF neste caso)
94       res.setHeader('Content-Type', 'application/pdf');
95       res.setHeader('Content-Disposition', `attachment; filename="${nome_arquivo}"`);
96
97       res.send(arquivo);
98     } else {
99       return res.status(404).json({ error: 'Folha de pagamento não encontrada' });
100    }
101  } catch (error) {
102    console.error('Erro ao consultar o banco de dados', error);
103    return res.status(500).json({ error: 'Erro ao consultar o banco de dados' });
104  }
105 });
106
107
108 app.listen(port, () => {
109   console.log(`Servidor está rodando na porta ${port}`);
110 });

```

Fonte: Próprios autores, 2023.

4.4.5 Dicionário de dados

Tabela 6: Entidade: Funcionário.

Entidade: Funcionário					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	Id_Funcionario	Int		Não	Identificador do funcionário
	Dat_Nascimento	Date	50	Não	Data de nascimento do funcionário no formato dd/mm/aa
	Cpf	Varchar	11	Não	Cpf do funcionário
	Rg	Varchar	9	Não	Rg do funcionário
	Email	Varchar	50		Email do funcionário
	Telefone	Varchar	9		Telefone do funcionário
	Endereco	Varchar	100		Endereço do funcionário
	Cidade	Varchar	50		Cidade onde o funcionário reside
	Estado	Varchar	50		Estado onde o funcionário reside
	Cargo	Varchar	20	Não	Cargo exercido pelo funcionário na empresa
	Salario_Base	Double		Não	Salário base ganho pelo funcionário
	Dat_Contratação	Date		Não	Data de contrato do funcionário no formato dd/mm/aa
	Cont_Dependente	int		Não	Quantidade de dependentes do funcionário
	Status	Varchar	20		Situação atual do funcionário
FK	Id_Empresa	Int		Não	Identificador da empresa a qual o funcionário faz parte (Chave estrangeira da entidade EmpresaCliente)
FK	num_Conta	int	1	Não	Identificador da conta (Chave estrangeira da entidade ContaBancaria)

Fonte: Próprios autores, 2023.

Tabela 7: Entidade: Empresa.

Entidade: Empresa					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	Id_Empresa	Int		Não	Identificador da empresa
	Id_Departamento	Int		Não	Identificador de cada departamento da empresa
	Nome_Departamento	Char	50	Não	Nome do departamento da empresa

Fonte: Próprios autores, 2023.

Tabela 8: Entidade: ValoresFolha.

Entidade: ValoresFolha					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	Id_Valor	Int		Não	Identificador do valor aplicado
	Nome	Char	50	Não	Nome do desconto aplicado
	Valor	Double		Não	Valor do desconto aplicado
	Descricao	varchar	100	Não	Descrição do valor, podendo ser desconto ou remuneração

Fonte: Próprios autores, 2023.

Tabela 9: Entidade: ContaBancaria.

Entidade: ContaBancaria					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	num_Contra	Int		Não	Número da conta bancária
	id_Banco	int		Não	Identificador do Banco
	num_Agencia	Double		Não	Número da Agência

Fonte: Próprios autores, 2023.

Tabela 10: Entidade: Empresa.

Entidade: Empresa					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	Id_Empresa_Cliente	Int		Não	Identificador do Benefício
	Razao_Social	Char	50	Não	Nome do Benefício
	Cnpj	Varchar	14	Não	Cnpj do Cliente
	Endereço	Varchar	50	Não	Endereço do cliente
	Cidade	Varchar	50	Não	Cidade onde a empresa está localizada
	Estado	Varchar	50	Não	Estado onde a empresa está localizada
	Telefone	Varchar	9		Telefone do cliente
	Area_Atuacao	Varchar	20	Não	A área de atuação em que o cliente se encontra
	Quant_Funcionario	Int		Não	Quantidade de funcionários que a empresa possui

Fonte: Próprios autores, 2023.

Tabela 11: Entidade: FolhaPagamento.

Entidade: FolhaPagamento					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	Id_Folha	Int		Não	Identificador da Folha de pagamento
	Data_Emissao	Date		Não	Data de emissão da folha de pagamento no formato dd/mm/aa
	Mes_Refencia	Varchar	14	Não	Referência do mês em que foram utilizadas as informações do funcionário
	Horas_Trabalhadas	Double		Não	Horas trabalhadas do funcionário
	Salario_Liquido	Double		Não	Salário Líquido do funcionário
	Salario_Bruto	Double		Não	Salário Bruto do funcionário
FK	id_Valor	int		Não	Chave estrangeira da entidade: ValoresFolha
FK	id_Tabela_Prog	Int		Não	Chave estrangeira da entidade: TabelaProgressiva
FK	Num_Conta	Int		Não	Chave estrangeira da entidade: ContaBancaria
FK	Id_Empresa_Cliente	Int		Não	Chave estrangeira da entidade EmpresaCliente
FK	Id_Funcionario	Int		Não	Chave estrangeira da entidade: Funcionario

Fonte: Próprios autores, 2023.

Tabela 12: Entidade: TabelaProgressiva.

Entidade: TabelaProgressiva					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	Id_Tabela_Prog	Int		Não	Identificador da TabelaProgressiva
	Ano	int		Não	Ano vigente da Tabela
	Faixa_Monetaria_Min	Double		Não	Salário-mínimo para comparação e aplicação correta do inss
	Faixa_Monetaria_Max	Double		Não	Salário máximo para comparação e aplicação correta do inss
	Aliquota	Double		Não	Porcentagem da alíquota
	Deducao	Double		Não	Valor de dedução do IR
	Valor_Dependente	Double		Não	Valor para cada dependente do funcionário

Fonte: Próprios autores, 2023.

Tabela 13: Entidade: Usuario.

Entidade: Usuario					
	Atributo	Tipo de Dado	Tamanho	Nulo	Descrição
PK	Num_Matricula	Varchar	20	Não	Identificador do usuário
	Nome	Char	20	Não	Nome do usuário
	Email	Varchar	50	Não	Email do usuário
	Senha	Varchar	50	Não	Senha do usuário
	Perfil	Char	20	Não	Nível de acesso do usuário
	Cargo	Char	20	Não	Cargo do usuário
FK	id_Departamento	int		Não	Chave estrangeira da entidade: Empresa

Fonte: Próprios autores, 2023.

4.5 Design (UX)

A (NBR 9241-11, 2002) estabelece princípios de ergonomia para o projeto de sistemas interativos. Seu objetivo é garantir a usabilidade e a satisfação do usuário ao desenvolver interfaces eficazes, eficientes e agradáveis de usar.

A norma enfatiza a importância da ergonomia cognitiva, abrangendo aspectos como facilidade de aprendizado, eficiência de uso, recuperação de erros e compatibilidade com o usuário. A interface utilizada neste projeto segue a norma (NBR 9241-11, 2002) e tem como foco facilitar a experiência do usuário e estimular a interação com o produto, exercendo um papel importante em apresentar uma interface simples, eficaz e intuitiva para que o indivíduo consiga ver de forma clara o que ele necessita e o que cada função desempenha.

Para nós da Devcoast a identidade visual de um software é muito importante pois acreditamos que isso desempenha um papel fundamental na forma como nossa marca é percebida, reconhecida e lembrada pelos nossos usuários. Para nós ela possui forte influência a na credibilidade, a atratividade, eficácia das estratégias de marketing e na usabilidade.

A seguir falaremos sobre quais conceitos usamos no design de cada aplicação.

No app Registro de Ponto e o web site Portal do Funcionário o usuário é o próprio funcionário do cliente, por isso optamos por um design simples que possui

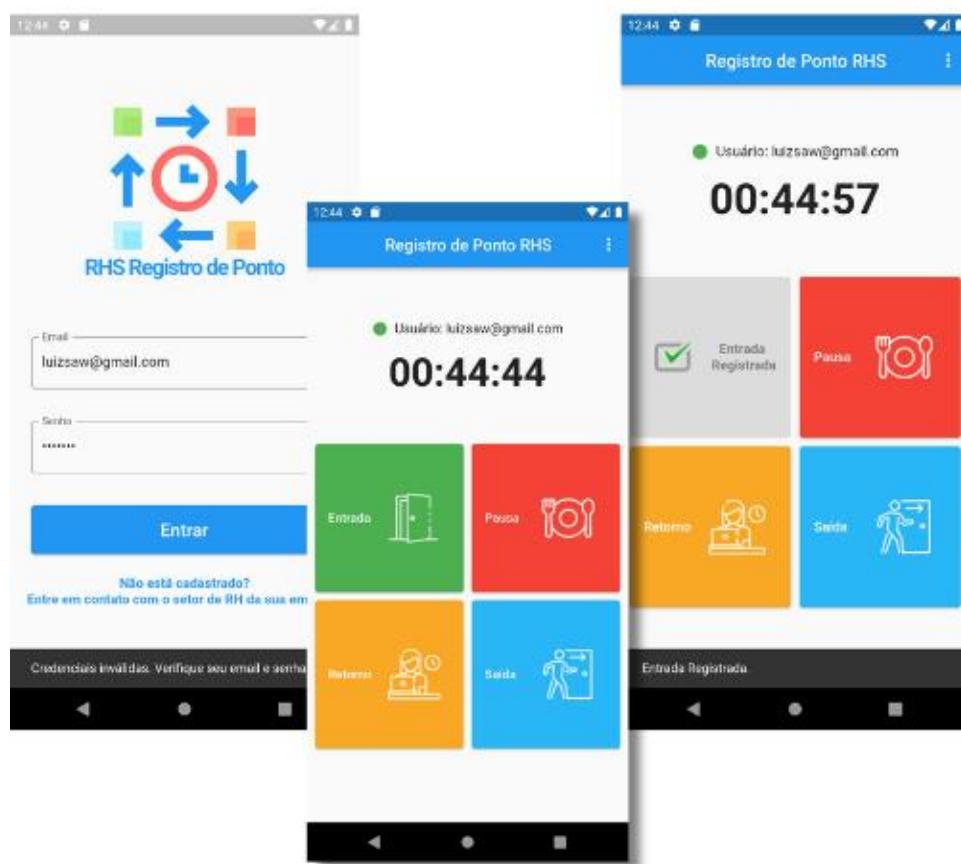
cores vibrantes e que oferece uma usabilidade intuitiva onde não é necessário guias complexos para sua utilização.

Figura 33: Logo do app Registro de Ponto.



Fonte: Próprios autores, 2023.

Figura 34: Apresentação do app Registro de ponto.



Fonte: Próprios autores, 2023.

Figura 35: Logo do site Portal do Funcionário.



Fonte: Próprios autores, 2023.

Figura 36: Apresentação site Portal do Funcionário.

ID	Funcionário	Data	Hora	Tipo
1	30	30/10/2023	23:12	Entrada
2	30	30/10/2023	23:12	Pausa
3	30	30/10/2023	23:12	Retorno
4	30	30/10/2023	23:12	Saída
9	30	8/11/2023	09:14	Entrada
10	30	8/11/2023	09:14	Pausa
11	30	8/11/2023	09:14	Retorno
12	30	8/11/2023	09:14	Saída

Fonte: Próprios autores, 2023.

O design da aplicação de folha de pagamento foi desenvolvido para atender às necessidades dos administradores do sistema. Optamos por uma paleta de cores com tons mais suaves, visando mitigar a fadiga ocular, uma vez que os utilizadores passam longas horas diante da tela, já que todas as operações relacionadas à folha de pagamento são executadas neste sistema.

Por se tratar de um sistema com um nível de detalhamento maior foi necessário desenvolver um guia para a utilização do sistema.

Figura 37: Logo do app Folha de Pagamento.



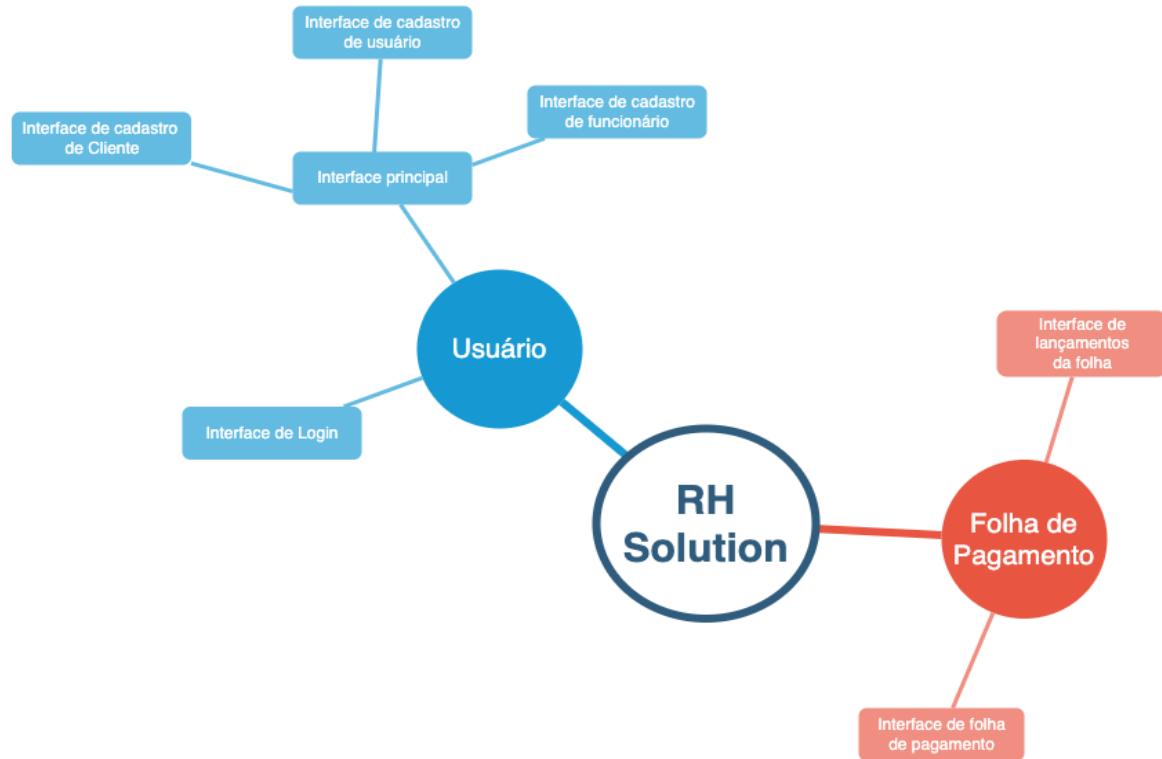
Fonte: Próprios autores, 2023.

Figura 38: Apresentação app Folha de pagamento.

Fonte: Próprios autores, 2023.

4.5.1 Interface Gráfica

Figura 39: Mind Map do projeto de interface da folha de pagamento.



Fonte: Próprios autores, 2023.

Tabela 14: Tabela da complexidade dos campos.

Requisitos	Descrição	Complexidade
[RF001]	Interface de Login	Baixa
[RF002]	Interface principal do sistema	Baixa
[RF003]	Interface de cadastro de usuário	Baixa
[RF004]	Interface de cadastro de empresa	Baixa
[RF005]	Interface de cadastro de funcionário	Média
[RF006]	Interface de lançamento da folha de pagamento	Baixa
[RF007]	Interface da folha de pagamento	Baixa
[RF008]	Interface de Login do registro de ponto	Baixa
[RF009]	Interface tela Inicial registro de ponto	Baixa
[RF010]	Interface login portal do funcionário	Baixa
[RF011]	Interface visualização folha de pagamento	Baixa

Fonte: Próprios autores, 2023.

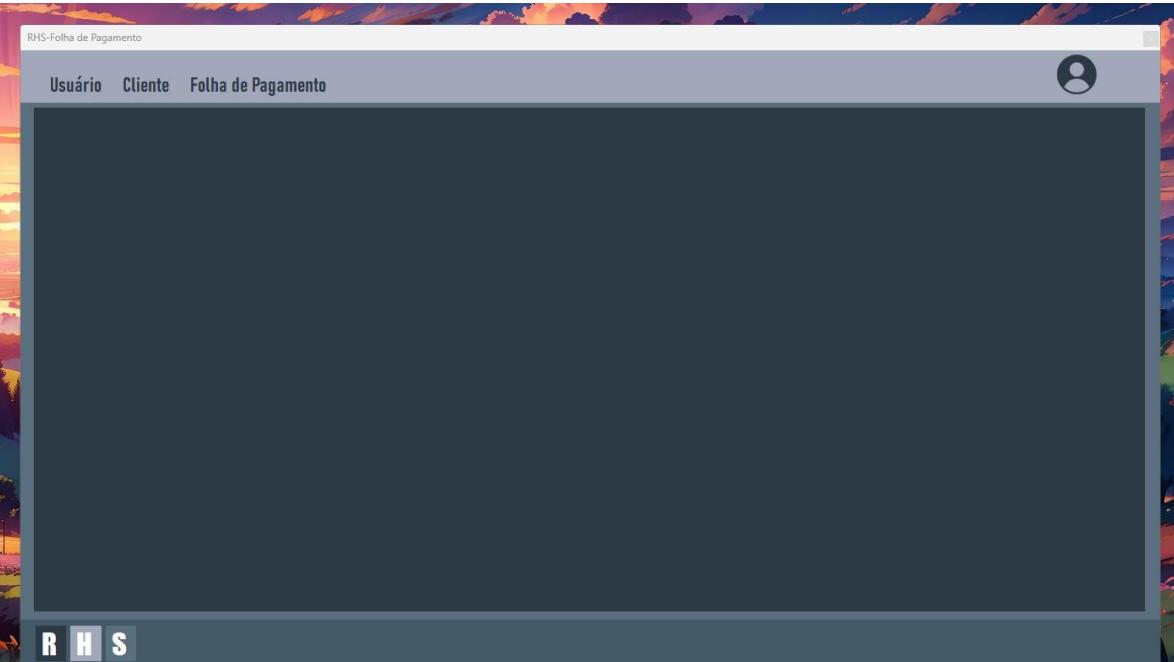
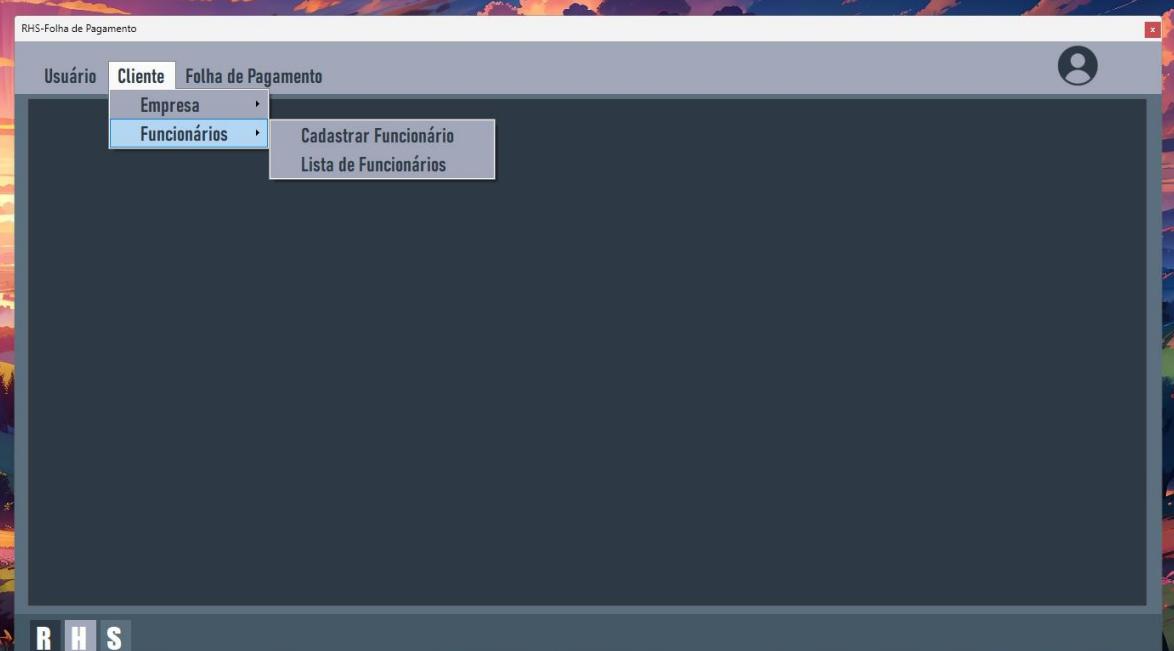
Tabela 15: Requisito de interface RF001.

<p>[RF001]</p> <p>Interface de Login</p> <p>A interface de login e senha possui 2 campos: Login e Senha e um botão de Entrar, no campo de "Login" o usuário deverá inserir o número de matrícula e a senha no campo "Senha", após clicar no botão "Entrar" o sistema realiza a validação, se caso validado o usuário acessa a interface principal do sistema, caso não validado deverá ser exibido uma mensagem de senha ou Login inválido.</p>

<p>Impacto: Esta interface está relacionada com o acesso ao sistema, porém não causa impacto em outras interfaces.</p>

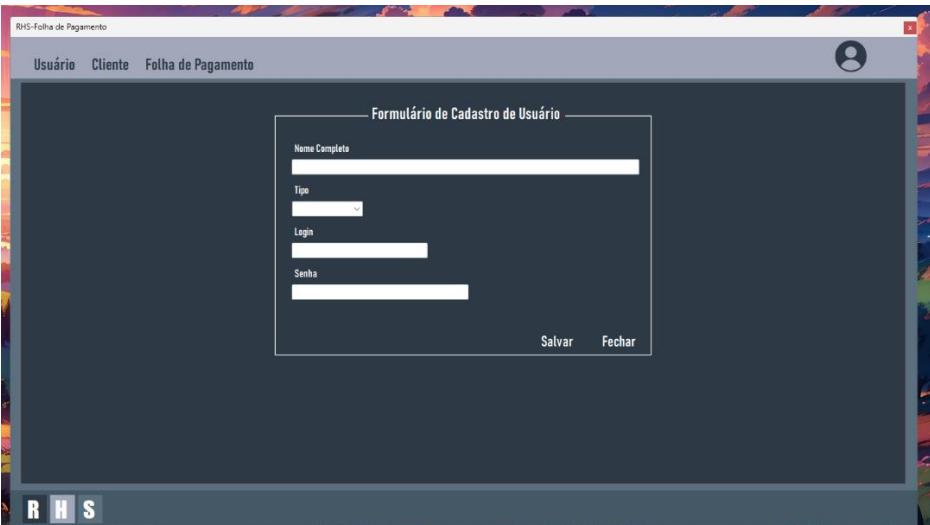
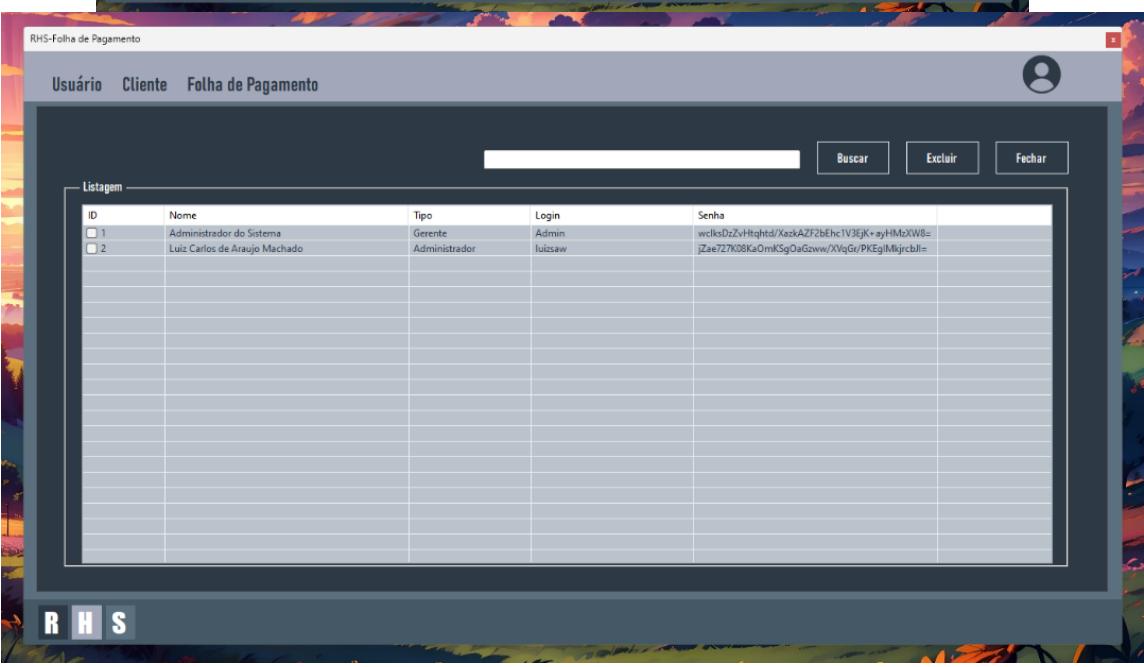
Fonte: Próprios autores, 2023.

Tabela 16: Requisito de interface RF002.

[RF002]
Interface Principal do sistema
A interface principal onde o usuário pode navegar pelos menus e submenus, como: Folha de pagamento, Cadastrar Empresas, Cadastrar Funcionário e lançamento da folha.


Impacto: Esta interface está relacionada com a navegação do sistema causando impacto em outras interfaces, pois sem ela não será possível interagir com nenhuma das demais interfaces do sistema.

Fonte: Próprios autores, 2023.

Tabela 17: Requisitos de interface RF003, RF004 e RF005.

[RF003], [RF004], [RF005]															
Interface de Cadastros															
<p>Na interface de cadastro de usuários possui os seguintes campos a serem preenchidos: nome, número de matrícula, departamento, cargo, e-mail, selecionar perfil e definir senha.</p> <p>Na interface de cadastro de empresa possui os seguintes campos: razão social, CNPJ, endereço, cidade, estado, telefone, área de atuação, quantidade de funcionários, esta aba deverá ser chamada de "Cadastro".</p> <p>Na interface de cadastro de funcionário possuí os campos divididos em 2 áreas, como informações pessoais, e bancárias, esta aba é chamada de "Cadastro de Funcionários". Na área de informações pessoais tem os seguintes campos: nome, data de nascimento, RG, CPF, endereço, cidade, estado, e-mail, telefone, cargo, data de admissão, status, salário base e quantidades de dependentes. E no setor informações bancárias possui os seguintes campos: instituição bancária, número da agência, número da conta.</p>															
 <p>The screenshot shows a modal window titled 'Formulário de Cadastro de Usuário'. It contains four input fields: 'Nome Completo', 'Tipo' (with a dropdown arrow), 'Login', and 'Senha'. At the bottom right of the modal are 'Salvar' and 'Fechar' buttons.</p>															
 <p>The screenshot shows a main application window titled 'RHS-Folha de Pagamento'. The top navigation bar includes 'Usuário', 'Cliente', and 'Folha de Pagamento'. Below the navigation is a search bar with 'Buscar', 'Excluir', and 'Fechar' buttons. A table titled 'Listagem' displays user data:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Nome</th> <th>Tipo</th> <th>Login</th> <th>Senha</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Administrador do Sistema</td> <td>Gerente</td> <td>Admin</td> <td>wclksDzV-Htqhtd/XezkAZF2bEhcIV3Ejk+ayHMtzXW8=jZae727K08Ka0mKSgQaGzwww/XVqGr/PKEgIMkjrcbjI=</td> </tr> <tr> <td>2</td> <td>Luiz Carlos de Araújo Machado</td> <td>Administrador</td> <td>luizsaw</td> <td></td> </tr> </tbody> </table>	ID	Nome	Tipo	Login	Senha	1	Administrador do Sistema	Gerente	Admin	wclksDzV-Htqhtd/XezkAZF2bEhcIV3Ejk+ayHMtzXW8=jZae727K08Ka0mKSgQaGzwww/XVqGr/PKEgIMkjrcbjI=	2	Luiz Carlos de Araújo Machado	Administrador	luizsaw	
ID	Nome	Tipo	Login	Senha											
1	Administrador do Sistema	Gerente	Admin	wclksDzV-Htqhtd/XezkAZF2bEhcIV3Ejk+ayHMtzXW8=jZae727K08Ka0mKSgQaGzwww/XVqGr/PKEgIMkjrcbjI=											
2	Luiz Carlos de Araújo Machado	Administrador	luizsaw												

RHS-Folha de Pagamento

Usuário Cliente Folha de Pagamento

Formulário de cadastro de Empresa

Razão Social	CNPJ	
Logradouro	Cidade	Estado
Nome do logradouro, n.º, complemento (se houver)		
Telefone	Área de Atuação	Total de Funcionários
<input type="button" value="Salvar"/> <input type="button" value="Fechar"/>		

R H S

RHS-Folha de Pagamento

Usuário Cliente Folha de Pagamento

Formulário de cadastro de Funcionário

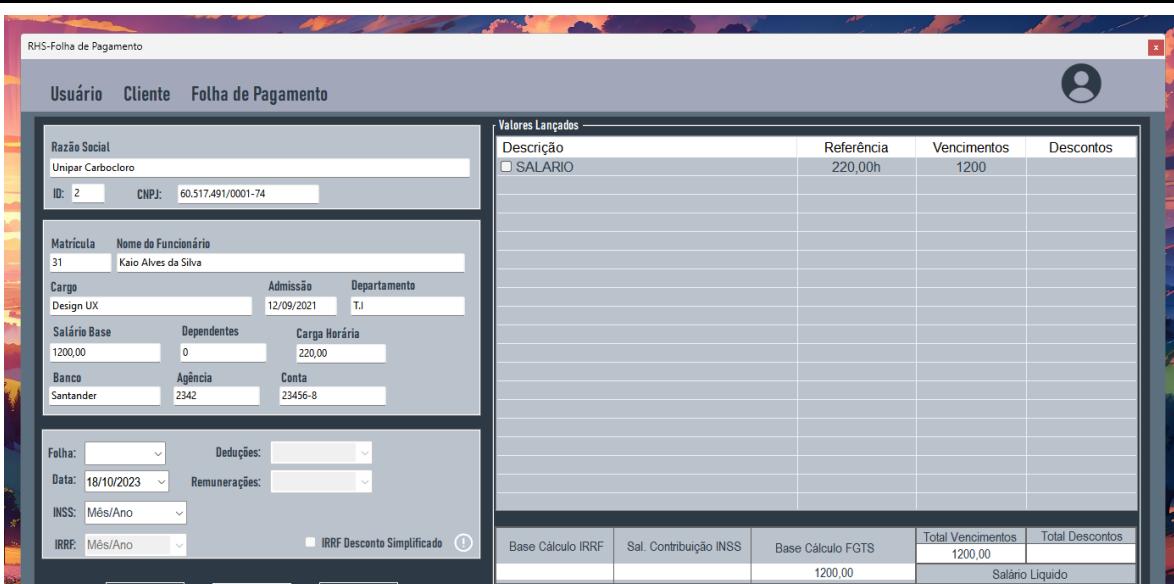
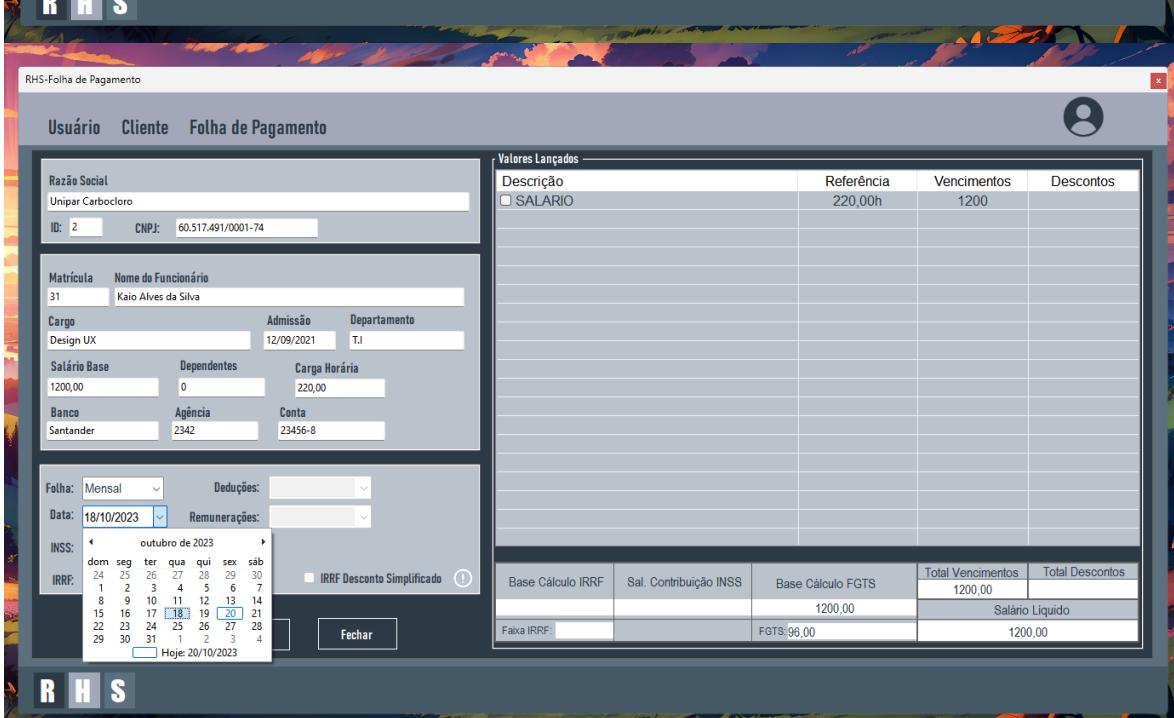
Empresa	Nome completo	Status										
Unipar Carbocloro	Kaio Alves da Silva	Ativo										
Data de Nascimento	CPF	RG	E-mail									
12/04/1998	345.122.772-11	45.897.007-3	Kaio@email.com									
Ex: 01/04/1992	Ex: 325.767.940-84	Ex: 17.241.997-9										
Telefone	Logradouro	Estado										
(13) 99945-4421	Avenida Presidente Wilson, 411, Apto 32	São Paulo										
Cidade	Cargo	Departamento										
São Vicente	Design UX	T.I.										
Salário Base	Quantidade de Dependentes	Data da Contratação	Carga Horária									
1200	0	12/09/2021	220									
Ex: 01/04/1992												
Dados Bancários <table border="1"> <tr> <td>N.º Conta</td> <td>N.º Agência</td> <td>Nome do Banco</td> </tr> <tr> <td>23456-8</td> <td>2342</td> <td>Santander</td> </tr> <tr> <td>C/ digito, Ex: 1371807-5</td> <td>Ex: 1234</td> <td></td> </tr> </table>				N.º Conta	N.º Agência	Nome do Banco	23456-8	2342	Santander	C/ digito, Ex: 1371807-5	Ex: 1234	
N.º Conta	N.º Agência	Nome do Banco										
23456-8	2342	Santander										
C/ digito, Ex: 1371807-5	Ex: 1234											
<input type="button" value="Salvar"/> <input type="button" value="Fechar"/>												

R H S

Impacto: As Interfaces de cadastros são fundamentais para todo o funcionamento de todo o sistema.

Fonte: Próprios autores, 2023.

Tabela 18: Requisito de interface RF006.

[RF006]
Interface de lançamento da folha de pagamento
<p>Na Tela "Lançamentos da Folha" devem ser preenchidas com as informações recuperadas do cadastro do funcionário e da folha de ponto (registrado no sistema de ponto mobile). Esta interface contém os seguintes campos: selecionar o funcionário, adicionar valores de benefícios, valores de descontos, horas trabalhadas, faltas, selecionar o ano das tabelas de desconto do INSS e IR.</p> 


RHS-Folha de Pagamento

Usuário Cliente Folha de Pagamento

Razão Social
 Unipar Carbocloro
 ID: 2 CNPJ: 60.517.491/0001-74

Valores Lançados

Descrição	Referência	Vencimentos	Descontos
<input type="checkbox"/> SALARIO	220,00h	1200	
<input type="checkbox"/> I.N.S.S	7,50%	90	
<input type="checkbox"/> I.R.R.F	0,00%	0	

Matrícula Nome do Funcionário
 31 Kaio Alves da Silva

Cargo
 Design UX **Admissão** 12/09/2021 **Departamento** T.I

Salário Base 1200,00 **Dependentes** 0 **Carga Horária** 220,00

Banco Santander **Agência** 2342 **Conta** 23456-8

Folha: Mensal **Deduções:** Plano de Saúde

Valor Plano de Saúde: 130 R\$
A parcela mensal não pode comprometer mais que 30% do salário do trabalhador.

Data: 18/10/2023 **Remunerações:**

IRRF Desconto Simplificado

INSS: Maio/2023

IRRF: Maio/2023

Apagar
Emitir
Fechar

R H S

RHS-Folha de Pagamento

Usuário Cliente Folha de Pagamento

Razão Social
 Unipar Carbocloro
 ID: 2 CNPJ: 60.517.491/0001-74

Valores Lançados

Descrição	Referência	Vencimentos	Descontos
<input type="checkbox"/> SALARIO	220,00h	1200	
<input type="checkbox"/> I.N.S.S	7,50%	90	
<input type="checkbox"/> I.R.R.F	0,00%	0	
<input type="checkbox"/> DESC. PLANO DE SAÚDE		130,00	
<input type="checkbox"/> VALE TRANSPORTE	6%	72,00	
<input type="checkbox"/> EMPRÉSTIMO 1/3		122,22	

Matrícula Nome do Funcionário
 31 Kaio Alves da Silva

Cálculo de Horas Extras

Salário
 1200
Carga Horária Mensal
 220
Quantidade de horas extras
 6
% da hora extra
 50

Folha: Mensal **Deduções:** Outros

IRRF Desconto Simplificado

Data: 18/10/2023 **Remunerações:** Hora Extras

INSS: Maio/2023

IRRF: Maio/2023

Apagar
Emitir
Fechar

R H S

The image displays two side-by-side screenshots of the RHS-Folha de Pagamento application. Both screenshots show a salary slip for employee Kaio Alves da Silva, with the following details:

- Razão Social:** Unipar Carbocloro
- ID:** 2
- CNPJ:** 60.517.491/0001-74
- Matrícula:** 31
- Nome do Funcionário:** Kaio Alves da Silva
- Cargo:** Design UX
- Admissão:** 12/09/2021
- Departamento:** T.I
- Salário Base:** 1200,00
- Dependentes:** 0
- Carga Horária:** 220,00
- Banco:** Santander
- Agência:** 2342
- Conta:** 23456-8

Folha: Mensal | **Deduções:** Outros | **Data:** 18/10/2023 | **Remunerações:** Salario Família | **INSS:** Maio/2023 | **IRRF:** Maio/2023 | **IRRF Desconto Simplificado:**

Apagar | **Emitir** | **Fechar**

Valores Lançados

Descrição	Referência	Vencimentos	Descontos
<input type="checkbox"/> SALARIO	220,00h	1200	93,68
<input type="checkbox"/> I.N.S.S	7,50%		0
<input type="checkbox"/> I.R.R.F	0,00%		130,00
<input type="checkbox"/> DESC. PLANO DE SAÚDE			72,00
<input type="checkbox"/> VALE TRANSPORTE	6%		122,22
<input type="checkbox"/> EMPRÉSTIMO 1/3			
<input type="checkbox"/> HORAS EXTRAS 50%	6h	49,09	
<input type="checkbox"/> SALARIO FAMÍLIA	1	59,82	

Base Cálculo IRRF 1155,41 | **Sal. Contribuição INSS** 1249,09 | **Base Cálculo FGTS** 1249,09 | **Total Vencimentos** 1308,91 | **Total Descontos** 417,90

Faixa IRRF: 0,00% | **FGTS:** 99,93 | **Salário Líquido** 891,01

R H S

The second screenshot shows a modal dialog box for "Adicionais" (Additionals) with the following fields:

- Adicional noturno:**
 - Salário: 1200,00
 - Carga Horária Mensal: 220
 - Horas Noturnas: 12
 - Percentual: 20 %
em percentagem (Ex: 20, 25)
- Adicional de insalubridade:**
 - Valor: 30 %
em percentagem (Ex: 10, 20, 30)

OK

Valores Lançados

Descrição	Referência	Vencimentos	Descontos
<input type="checkbox"/> SALARIO	220,00h	1200	93,68
<input type="checkbox"/> I.N.S.S	7,50%		0
<input type="checkbox"/> I.R.R.F	0,00%		130,00
<input type="checkbox"/> DESC. PLANO DE SAÚDE			72,00
<input type="checkbox"/> VALE TRANSPORTE	6%		122,22
<input type="checkbox"/> EMPRÉSTIMO 1/3			
<input type="checkbox"/> HORAS EXTRAS 50%	6h	49,09	
<input type="checkbox"/> SALARIO FAMÍLIA	1	59,82	

Base Cálculo IRRF 1155,41 | **Sal. Contribuição INSS** 1249,09 | **Base Cálculo FGTS** 1249,09 | **Total Vencimentos** 1308,91 | **Total Descontos** 417,90

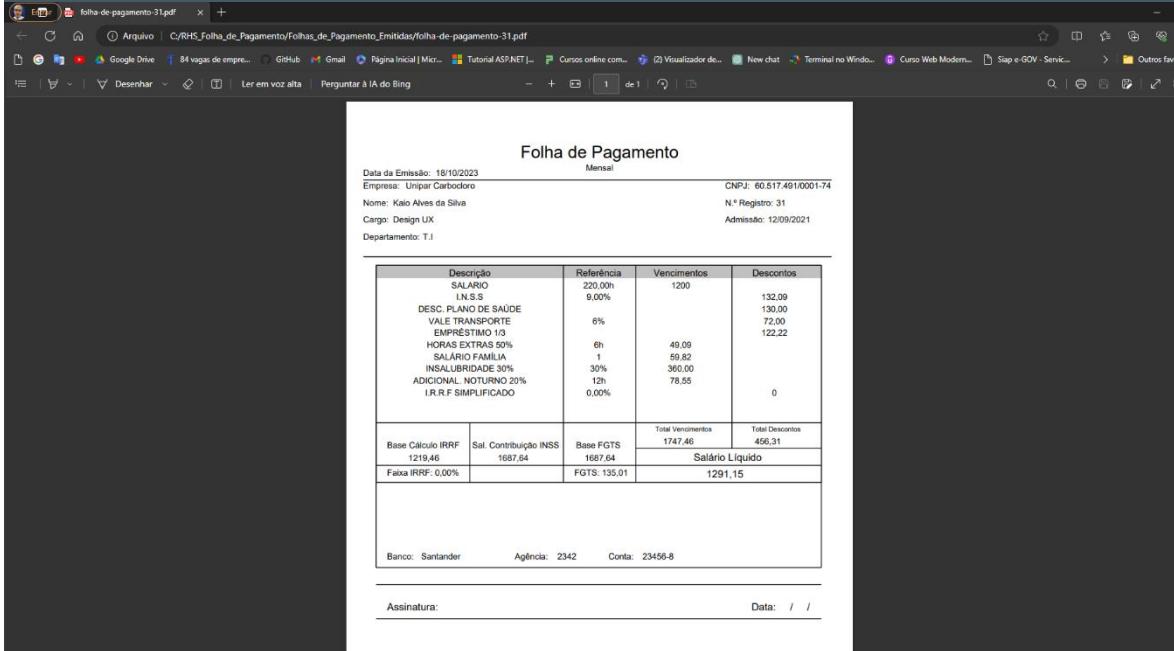
Faixa IRRF: 0,00% | **FGTS:** 99,93 | **Salário Líquido** 891,01

R H S

Impacto: Esta interface está relacionada diretamente com a interface da folha de pagamento.

Fonte: Próprios autores, 2023.

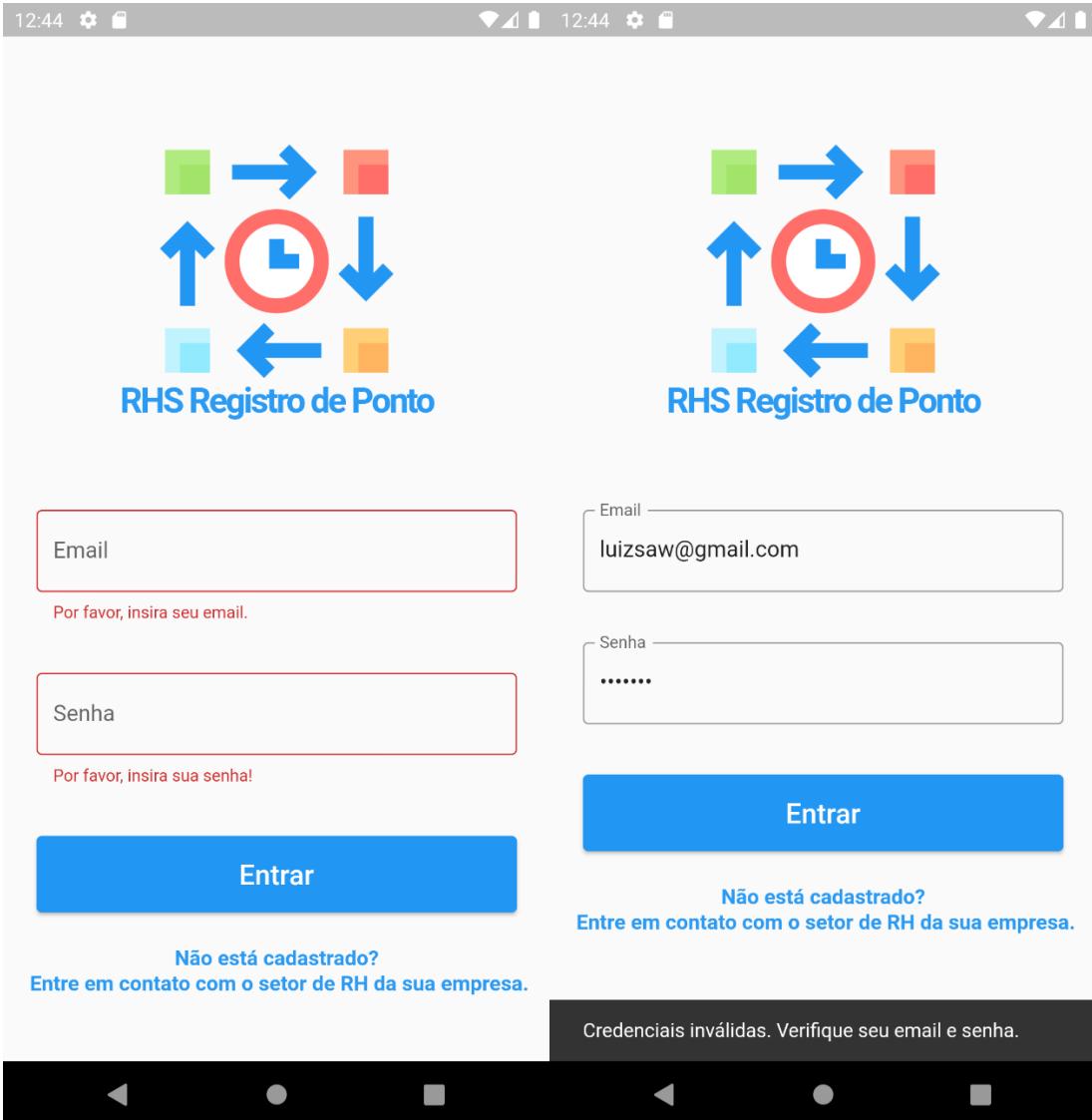
Tabela 19: Requisito de interface RF007.

[RF007]																																																																
Interface da folha de pagamento																																																																
Na interface da Folha de Pagamento será exibido todas as informações lançadas e calculadas na folha de lançamento, nesta tela possuirá um botão "Emitir" que exportará a folha de pagamento em arquivo pdf para o desktop.																																																																
 <p>The screenshot shows a software interface for generating a salary slip. It includes fields for selecting the pay period (Folha: Mensal, Semanal, Quízenal), date (Data: Mês/Ano), deductions (Deduções), remunerations (Remunerações), and simplified tax deduction (IRRIF Desconto Simplificado). At the bottom, there are three buttons: 'Apagar' (Delete), 'Emitir' (Generate), and 'Fechar' (Close). The 'Emitir' button is circled in red.</p>																																																																
 <p>The screenshot shows a PDF document titled "Folha de Pagamento" for a monthly period (Mensal). The document contains the following information:</p> <ul style="list-style-type: none"> Employee Details: Nome: Kael Alves da Silva, Cargo: Design UX, Departamento: T.I. Period: Data da Emissão: 18/10/2023, Empresa: Unipar Cardoso, CNPJ: 60.517.491/0001-74. Salary Breakdown: <table border="1"> <thead> <tr> <th>Descrição</th> <th>Referência</th> <th>Vencimentos</th> <th>Descontos</th> </tr> </thead> <tbody> <tr> <td>SALÁRIO</td> <td>220.00h</td> <td>1200</td> <td>132,09</td> </tr> <tr> <td>I.N.S.S.</td> <td>9,00%</td> <td></td> <td>130,00</td> </tr> <tr> <td>DESC. PLANO DE SAÚDE</td> <td></td> <td></td> <td>72,00</td> </tr> <tr> <td>VALE TRANSPORTE</td> <td>6%</td> <td></td> <td>122,22</td> </tr> <tr> <td>EMPRÉSTIMO 1/2</td> <td></td> <td></td> <td></td> </tr> <tr> <td>HORAS EXTRAS 50%</td> <td>6h</td> <td>69,99</td> <td></td> </tr> <tr> <td>SALÁRIO FAMILIA</td> <td>1</td> <td>69,99</td> <td></td> </tr> <tr> <td>INSALUBRIDADE 30%</td> <td>30%</td> <td>360,00</td> <td></td> </tr> <tr> <td>ADICIONAL NOTURNO 20%</td> <td>12h</td> <td>78,55</td> <td></td> </tr> <tr> <td>IRR.F SIMPLIFICADO</td> <td>0,00%</td> <td></td> <td>0</td> </tr> </tbody> </table> Summary Table: <table border="1"> <thead> <tr> <th>Base Cálculo IRRF</th> <th>Sal. Contribuição INSS</th> <th>Base FGTS</th> <th>Total Vencimentos</th> <th>Total Descontos</th> </tr> </thead> <tbody> <tr> <td>1219,46</td> <td>1687,64</td> <td>1687,64</td> <td>1747,46</td> <td>456,31</td> </tr> <tr> <td colspan="2">Folha IRRF: 0,00%</td> <td>FGTS: 135,01</td> <td>Salário Líquido</td> <td></td> </tr> <tr> <td colspan="2"></td> <td></td> <td>1291,15</td> <td></td> </tr> </tbody> </table> Bottom Information: Banco: Santander, Agência: 2342, Conta: 23456-8, Assinatura: _____, Data: / / 	Descrição	Referência	Vencimentos	Descontos	SALÁRIO	220.00h	1200	132,09	I.N.S.S.	9,00%		130,00	DESC. PLANO DE SAÚDE			72,00	VALE TRANSPORTE	6%		122,22	EMPRÉSTIMO 1/2				HORAS EXTRAS 50%	6h	69,99		SALÁRIO FAMILIA	1	69,99		INSALUBRIDADE 30%	30%	360,00		ADICIONAL NOTURNO 20%	12h	78,55		IRR.F SIMPLIFICADO	0,00%		0	Base Cálculo IRRF	Sal. Contribuição INSS	Base FGTS	Total Vencimentos	Total Descontos	1219,46	1687,64	1687,64	1747,46	456,31	Folha IRRF: 0,00%		FGTS: 135,01	Salário Líquido					1291,15	
Descrição	Referência	Vencimentos	Descontos																																																													
SALÁRIO	220.00h	1200	132,09																																																													
I.N.S.S.	9,00%		130,00																																																													
DESC. PLANO DE SAÚDE			72,00																																																													
VALE TRANSPORTE	6%		122,22																																																													
EMPRÉSTIMO 1/2																																																																
HORAS EXTRAS 50%	6h	69,99																																																														
SALÁRIO FAMILIA	1	69,99																																																														
INSALUBRIDADE 30%	30%	360,00																																																														
ADICIONAL NOTURNO 20%	12h	78,55																																																														
IRR.F SIMPLIFICADO	0,00%		0																																																													
Base Cálculo IRRF	Sal. Contribuição INSS	Base FGTS	Total Vencimentos	Total Descontos																																																												
1219,46	1687,64	1687,64	1747,46	456,31																																																												
Folha IRRF: 0,00%		FGTS: 135,01	Salário Líquido																																																													
			1291,15																																																													

Impacto: Esta interface é fundamental para o sistema, pois ela é o resultado de todos os processos do sistema.

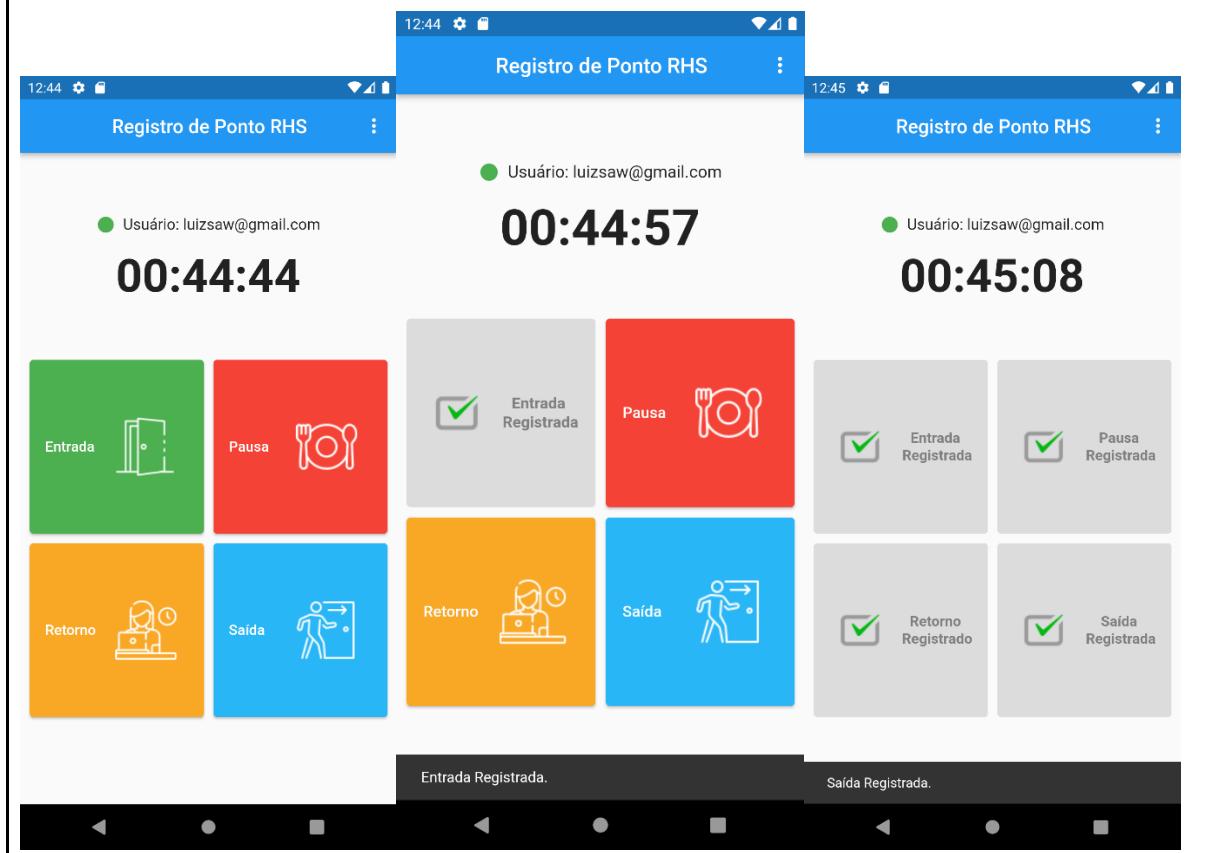
Fonte: Próprios autores, 2023.

Tabela 20: Requisito de interface RF008.

[RF008]
Interface de Login do registro de ponto
<p>A interface de login e senha no app registro de ponto possui 2 campos: Login e Senha e um botão de Entrar, no campo de "Login" o usuário deverá inserir o número de matrícula e a senha no campo "Senha", após clicar no botão "Entrar" o sistema realiza a validação, se caso validado o usuário acessa a interface principal do sistema, caso não validado deverá ser exibido uma mensagem de senha ou Login inválido.</p> 
<p>Impacto: Embora esta interface esteja relacionada com o acesso ao sistema, ela causa um impacto crítico sobre as demais interfaces, pois através desta que identificamos o usuário para ser registrado o ponto.</p>

Fonte: Próprios autores, 2023.

Tabela 21: Requisito de interface RF009.

[RF009]
Interface tela Inicial registro de ponto
Após efetuar o login com sucesso. É apresentado a tela inicial do aplicativo, onde o funcionário pode realizar os seus registros de horário de ponto.
 <p>The screenshot shows the 'Registro de Ponto RHS' application interface. At the top, there are two status bars: one for '12:44' and one for '12:45'. The main screen has a blue header bar with the title 'Registro de Ponto RHS'. Below the header, the text 'Usuário: luizsaw@gmail.com' is displayed. The center of the screen features a large digital clock showing '00:44:57'. To the left of the clock is the text '00:44:44' and to the right is '00:45:08'. Below the clock are four colored buttons: green ('Entrada'), red ('Pausa'), orange ('Retorno'), and blue ('Saída'). To the right of these buttons are two grey boxes labeled 'Entrada Registrada' and 'Pausa Registrada'. Below these are two more grey boxes labeled 'Retorno Registrado' and 'Saída Registrada'. At the bottom of the screen, there are two dark grey bars with the text 'Entrada Registrada.' and 'Saída Registrada.' respectively, followed by a series of navigation icons (back, forward, etc.).</p>
Impacto: é interface mais crítica do aplicativo, pois ela está relacionada diretamente no registro de ponto do funcionário onde é armazenado os dados no banco de dados que está na nuvem.

Fonte: Próprios autores, 2023.

Tabela 22: Requisito de interface RF010.

[RF010]
Interface login portal do funcionário
A interface de login e senha do portal do funcionário possui 2 campos: Login e Senha e um botão de Entrar, no campo de "Login" o usuário deverá inserir o número de matrícula e a senha no campo "Senha", após clicar no botão "Logar." o sistema realiza a validação, se caso validado o usuário acessa a interface principal do sistema, caso não validado deverá ser exibido uma mensagem de senha ou Login inválido.
Impacto: Esta interface está relacionada com o acesso ao sistema, porém não causa impacto em outras interfaces.

Fonte: Próprios autores, 2023.

Tabela 23: Requisito de interface RF011.

[RF011]																																																		
Interface visualização folha de pagamento																																																		
Nesta interface, o usuário poderá visualizar as abas de navegação do site. Nela, possui os seguintes botões, home para direcionar para a página Inicial, mensagens, onde é possível visualizar as mensagens que são recebidas, folha de pagamento, onde é possível visualizar as folhas de pagamento mensalmente e, por último, o botão controle de ponto onde o funcionário tem acesso aos registros de pontos realizados no aplicativo mobile.																																																		
 <table border="1"> <thead> <tr> <th>ID</th><th>ID Funcionário</th><th>Nome do Arquivo</th><th>Data de Emissão</th><th>Arquivo</th></tr> </thead> <tbody> <tr> <td>19</td><td>30</td><td>folha-de-pagamento-30.pdf</td><td>18/10/2023</td><td>Baixar PDF</td></tr> <tr> <td>20</td><td>30</td><td>folha-de-pagamento-30.pdf</td><td>08/11/2023</td><td>Baixar PDF</td></tr> <tr> <td>21</td><td>30</td><td>folha-de-pagamento-30.pdf</td><td>01/09/2023</td><td>Baixar PDF</td></tr> </tbody> </table>						ID	ID Funcionário	Nome do Arquivo	Data de Emissão	Arquivo	19	30	folha-de-pagamento-30.pdf	18/10/2023	Baixar PDF	20	30	folha-de-pagamento-30.pdf	08/11/2023	Baixar PDF	21	30	folha-de-pagamento-30.pdf	01/09/2023	Baixar PDF																									
ID	ID Funcionário	Nome do Arquivo	Data de Emissão	Arquivo																																														
19	30	folha-de-pagamento-30.pdf	18/10/2023	Baixar PDF																																														
20	30	folha-de-pagamento-30.pdf	08/11/2023	Baixar PDF																																														
21	30	folha-de-pagamento-30.pdf	01/09/2023	Baixar PDF																																														
 <table border="1"> <thead> <tr> <th>ID</th><th>Funcionário</th><th>Data</th><th>Hora</th><th>Tipo</th></tr> </thead> <tbody> <tr> <td>1</td><td>30</td><td>30/10/2023</td><td>23:12</td><td>Entrada</td></tr> <tr> <td>2</td><td>30</td><td>30/10/2023</td><td>23:12</td><td>Pausa</td></tr> <tr> <td>3</td><td>30</td><td>30/10/2023</td><td>23:12</td><td>Retorno</td></tr> <tr> <td>4</td><td>30</td><td>30/10/2023</td><td>23:12</td><td>Saida</td></tr> <tr> <td>9</td><td>30</td><td>08/11/2023</td><td>09:14</td><td>Entrada</td></tr> <tr> <td>10</td><td>30</td><td>08/11/2023</td><td>09:14</td><td>Pausa</td></tr> <tr> <td>11</td><td>30</td><td>08/11/2023</td><td>09:14</td><td>Retorno</td></tr> <tr> <td>12</td><td>30</td><td>08/11/2023</td><td>09:14</td><td>Saida</td></tr> </tbody> </table>						ID	Funcionário	Data	Hora	Tipo	1	30	30/10/2023	23:12	Entrada	2	30	30/10/2023	23:12	Pausa	3	30	30/10/2023	23:12	Retorno	4	30	30/10/2023	23:12	Saida	9	30	08/11/2023	09:14	Entrada	10	30	08/11/2023	09:14	Pausa	11	30	08/11/2023	09:14	Retorno	12	30	08/11/2023	09:14	Saida
ID	Funcionário	Data	Hora	Tipo																																														
1	30	30/10/2023	23:12	Entrada																																														
2	30	30/10/2023	23:12	Pausa																																														
3	30	30/10/2023	23:12	Retorno																																														
4	30	30/10/2023	23:12	Saida																																														
9	30	08/11/2023	09:14	Entrada																																														
10	30	08/11/2023	09:14	Pausa																																														
11	30	08/11/2023	09:14	Retorno																																														
12	30	08/11/2023	09:14	Saida																																														
Impacto: Esta interface é a mais crítica do sistema, pois ela é a coluna central de todo o sistema do portal do funcionário, onde é possível, navegar e visualizar todas as funções do site.																																																		

Fonte: Próprios autores, 2023.

4.6 Lei Geral de Proteção de Dados (LGPD)

Segundo o (Senado, 2020), a Lei Geral de Proteção de Dados (LGPD) é uma legislação brasileira que entrou em vigor em setembro de 2020. Ela foi criada com o objetivo de regulamentar o tratamento de dados pessoais por organizações públicas e privadas, com foco na proteção da privacidade e dos direitos dos cidadãos. A LGPD é fortemente inspirada no Regulamento Geral de Proteção de Dados (GDPR) da União Europeia e estabelece diretrizes para a coleta, armazenamento, processamento e compartilhamento de informações pessoais.

O sistema RHS de folha de pagamento trabalha de acordo com a Lei LGPD, isto é, antes das coletas dos dados, dos funcionários ou dos usuários, é apresentado um termo de consentimento sobre a utilização dos seus dados privados. Vide **anexo A**.

Além disso, o sistema de folha de pagamento conta com um sistema de criptografia de senhas apresentado a seguir:

Figura 40: Firebase console Authentication.

Identificador	Provedores	Data de criação	Último login	UID do usuário
gabriel@gmail.com		25 de set. d...	22 de out. d...	WMgTpzBAV8Fy20S1s054mGU...
luizsaw@gmail.com		17 de set. d...	24 de out. d...	mhhO6yB2UAgq7W2HMQ8nSXyf...

Fonte: Google Firebase, 2023.

No sistema de registro de ponto e portal do funcionário, toda a segurança de dados é gerenciada pelo console do Firebase, após criar um usuário e senha o Firebase authentication armazena as credenciais do usuário já criptografadas conforme a imagem acima.

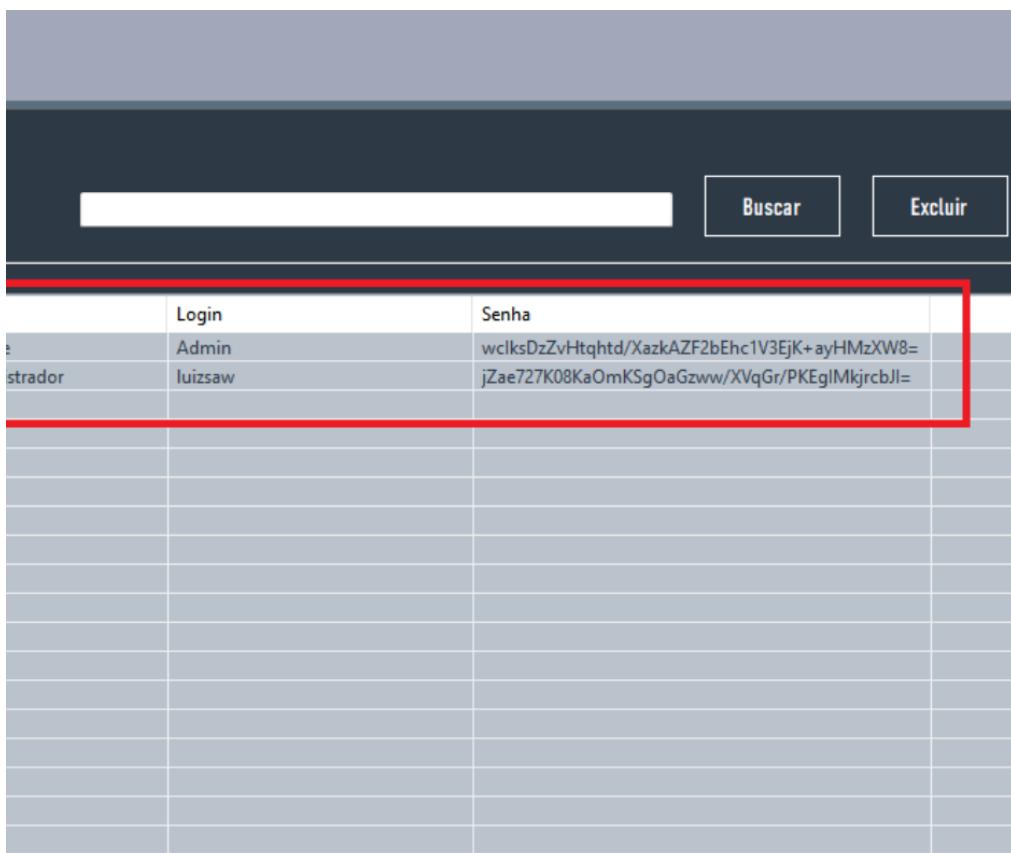
Já no sistema de folha de pagamento a criptografia é feita através de um método em C# que ao receber as credenciais do usuário, antes do armazenamento no banco de dados é criptografado, conforme as imagens a seguir:

Figura 41: Função de criptografia em C#.

```
// Método para criptografar senha
1 referência
private string CriptografarSenha(string senha)
{
    // lógica de criptografia hashing SHA256:
    using (SHA256 sha256 = SHA256.Create())
    {
        byte[] bytes = Encoding.UTF8.GetBytes(senha);
        byte[] hashBytes = sha256.ComputeHash(bytes);
        return Convert.ToString(hashBytes);
    }
}
```

Fonte: Próprios autores, 2023.

Figura 42: Lista de usuários com senha criptografadas.



The screenshot shows a user interface for managing users. At the top, there is a search bar and two buttons: 'Buscar' (Search) and 'Excluir' (Delete). Below this is a table with columns 'Login' and 'Senha'. Two rows are visible: one for 'Admin' with the password hash 'wclksDzZvHtqhtd/XazkAZF2bEhc1V3EjK+ayHMzXW8=' and another for 'luisaw' with the password hash 'jZae727K08KaOmKSgOaGzww/XVqGr/PKEglMkjrcbJl='. The row for 'luisaw' is highlighted with a red rectangular border.

	Login	Senha
strador	Admin	wclksDzZvHtqhtd/XazkAZF2bEhc1V3EjK+ayHMzXW8=
	luisaw	jZae727K08KaOmKSgOaGzww/XVqGr/PKEglMkjrcbJl=

Fonte: Próprios autores, 2023.

O método de criptografia utilizado é o SHA-256 (Secure Hash Algorithm 256 bits) que basicamente é uma função de hash criptográfica amplamente utilizada para transformar dados em uma sequência de caracteres alfanuméricos de 256 bits (ou 64 caracteres hexadecimais). É uma das variantes do algoritmo SHA-2, que foi

projeto para fornecer alta segurança e resistência a colisões (quando dois conjuntos de dados produzem o mesmo valor de hash).

Também vale ressaltar que acesso ao banco de dados é protegido com senha e seu acesso é restrito, sendo autorizado apenas pela empresa RHS.

Figura 43: API ElephantSQL.

The screenshot shows the ElephantSQL API interface. At the top, there's a navigation bar with the ElephantSQL logo, a dropdown menu set to 'RHSFolhaPagamento', and user information for 'Devcoast'. Below the header, the main area has two sections: 'Details' and 'API access'.

Details: This section displays various configuration parameters for the database instance:

Server	berry.db.elephantsql.com (berry-01)
Region	amazon-web-services:us-east-1
Created at	2023-08-28 23:11 UTC+00:00
User & Default database	kwlrltm
Password	*** (with copy and reset buttons)
URL	postgres://kwlrltm:***@berry.db.elephantsql.com/kwlrltm (with copy and refresh buttons)
Current database size	752 KB
Max database size	20 MB

To the right of the details, there's a 'Active Plan' section featuring a cartoon turtle named 'Tiny Turtle' and a 'Upgrade Instance' button.

API access: This section provides instructions for controlling the instance via the HTTP API and includes fields for generating an 'API Key' (with copy and reset buttons) and a 'Rotate key' button.

Fonte: Próprios autores, 2023.

Em resumo a Devcoast, atendeu à Lei Geral de Proteção de Dados (LGPD) por meio de uma série de medidas e práticas específicas, garantindo o cumprimento das regulamentações de privacidade de dados. Isso incluiu a implementação de políticas de privacidade transparentes e claras para os usuários, a obtenção de consentimento explícito para a coleta e processamento de dados pessoais, a anonimização e pseudonimização de informações sensíveis, a garantia de que apenas pessoal autorizado tem acesso a dados sensíveis, a criação de registros de atividades de processamento de dados, a pronta notificação de violações de dados às autoridades competentes e aos titulares dos dados afetados.

4.7 Código Fonte

Vejamos a seguir os principais códigos fonte de cada aplicação RHS Folha de Pagamento.

Figura 44: Classe FolhaPagamento desenvolvida em C#, parte 1/7.

```
C:\repos\RHS_Folha_de_Pagamento\Entity\FolhaPagamento.cs
 1  using PrjFolhaPagamento.Entity.FolhaPagamento;
 2  using RHS_Folha_de_Pagamento.Interface;
 3  using System;
 4  using System.Collections.Generic;
 5  using System.DirectoryServices.ActiveDirectory;
 6  using System.Linq;
 7  using System.Text;
 8  using System.Threading.Tasks;
 9
10 namespace PrjFolhaPagamento.Entity
11 {
12     internal class FolhaPagamento
13     {
14
15         public double cargaHoraMensal { get; set; }
16         public double horas_Trab { get; private set; }
17         public double percentualHE { get; private set; }
18         public double salario_Liquido { get; private set; }
19         public double salario_Bruto { get; set; }
20         public double valorFGTS { get; private set; }
21         public double valorSalarioFamilia { get; set; }
22
23
24         private ListViewItem _item;
25
26         //dados:Database;
27         //desconto_Inss:TabelaProgressiva;
28         //desconto_Ir:TabelaProgressiva;
29
30         //Construtores
31         public FolhaPagamento()
32         {
33
34         }
35
36         public void calcularFolha(string calculoTipo, double salario,
37             ListView lancamentos)
38         {
39             FolhaValores valoresFolha = new FolhaValores();
40             double valor = 0;
41
42             switch(calculoTipo)
43             {
44                 case "horasExtras":
45                     //Chamar form
46                     frm_HoraExtras hextras = new frm_HoraExtras();
47                     hextras.txb_salarioB = salario.ToString();
48                     hextras.txb_cargaH = cargaHoraMensal.ToString();
49                     hextras.ShowDialog();
50
51                     //Receber os dados do usuarios e converter para
52                     double
53                     this.horas_Trab = double.Parse
```

Fonte: Próprios autores, 2023.

Figura 45: Classe FolhaPagamento desenvolvida em C#, parte 2/7.

```
C:\repos\RHS_Folha_de_Pagamento\Entity\FolhaPagamento.cs 2
    52     (hextras.txb_quantH.Text);
    53         this.percentualHE = double.Parse
    54     (hextras.txb_Percent.Text);
    55
    56         // Realizar o calculo
    57         valor = horas_Trab * ((salario / cargaHoraMensal) + +
    58         ((salario / cargaHoraMensal) * (percentualHE/100)));
    59
    60             ///Adicionar os valores na classe ValoresFolha
    61             valoresFolha.tipo.Add("Remuneração");
    62             valoresFolha.descricao.Add("HORAS EXTRAS ");
    63             valoresFolha.referencia.Add(horas_Trab);
    64             valoresFolha.valor.Add(valor);
    65
    66                 //Alimentar os itens do listView
    67                 this._item = new ListViewItem
    68                 (valoresFolha.descricao[0] + percentualHE +"%");
    69                 this._item.SubItems.Add(valoresFolha.referencia
    70                 [0].ToString()+"h");
    71                 this._item.SubItems.Add(valoresFolha.valor
    72                 [0].ToString("0.00"));
    73                 this._item.SubItems.Add("");
    74                 lancamentos.Items.Add(this._item);
    75                 break;
    76             case "valeTransporte":
    77                 //Chamar form
    78                 frm_ValeTransporte vt = new frm_ValeTransporte();
    79                 vt.ShowDialog();
    80
    81                 //Receber os dados do usuarios e converter para
    82                 //double
    83                 int dias = int.Parse(vt.txb_dias.Text);
    84                 double valordiario = double.Parse
    85                 (vt.txb_valordiario.Text);
    86
    87                 // Realizar o calculo
    88                 valor = dias * valordiario;
    89
    90                 //verificar porcentagem de desconto
    91                 double limitePercentual = salario * 0.06; ////
    92                 Calcula 6% do salário
    93                 valor = (valor > limitePercentual) ?
    94                 limitePercentual : valor;
    95
    96                 //Adicionar os valores na classe ValoresFolha
    97                 valoresFolha.tipo.Add("Dedução");
    98                 valoresFolha.descricao.Add("VALE TRANSPORTE");
    99                 valoresFolha.referencia.Add(6.00);
    100                valoresFolha.valor.Add(valor);
    101
    102
    103                 //Alimentar os itens do listView
    104                 this._item = new ListViewItem
```

Fonte: Próprios autores, 2023.

Figura 46: Classe FolhaPagamento desenvolvida em C#, parte 3/7.

```
C:\repos\RHS_Folha_de_Pagamento\Entity\FolhaPagamento.cs
  3
  95      (valoresFolha.descricao[0]);
  96      this._item.SubItems.Add(valoresFolha.referencia
  97      [0].ToString()+"%");
  98      this._item.SubItems.Add("");
  99      this._item.SubItems.Add(valoresFolha.valor
 100     [0].ToString("0.00"));
 101     lancamentos.Items.Add(this._item);
 102
 103     break;
 104     case "fgts":
 105
 106         // Realizar o calculo
 107         salario_Bruto -= valorSalarioFamilia;
 108         valorFgts = (salario_Bruto * 0.08);
 109
 110         //Adicionar os valores na classe ValoresFolha
 111         valoresFolha.tipo.Add("Remuneração");
 112         valoresFolha.descricao.Add("FGTS");
 113         valoresFolha.referencia.Add(8);
 114         valoresFolha.valor.Add(valorFgts);
 115         break;
 116         case "salarioFamilia":
 117             //Chamar form
 118             frm_SalarioFamilia salarioFamilia = new
 119             frm_SalarioFamilia();
 120             salarioFamilia.set_RemunBase = salario.ToString
 121             ("0.00");
 122             salarioFamilia.ShowDialog();
 123
 124             //Receber os dados do usuarios e converter para
 125             double
 126             double quantFilho = double.Parse
 127             (salarioFamilia.get_Numfilhos.Text);
 128
 129             // Realizar o calculo
 130             if(double.Parse(salarioFamilia.get_RemunBase.Text) <=
 131             double.Parse(salarioFamilia.get_RemunMax.Text))
 132             {
 133                 valorSalarioFamilia = double.Parse
 134                 (salarioFamilia.get_ValorUnit.Text) * quantFilho;
 135
 136                 //Adicionar os valores na classe ValoresFolha
 137                 valoresFolha.tipo.Add("Remuneração");
 138                 valoresFolha.descricao.Add("SALÁRIO FAMÍLIA");
 139                 valoresFolha.referencia.Add(double.Parse
 140                 (salarioFamilia.get_Numfilhos.Text));
 141                 valoresFolha.valor.Add(valorSalarioFamilia);
 142
 143                 //Alimentar os itens do listView
 144                 this._item = new ListViewItem
 145                 (valoresFolha.descricao[0]);
 146                 this._item.SubItems.Add(valoresFolha.referencia
```

Fonte: Próprios autores, 2023.

Figura 47: Classe FolhaPagamento desenvolvido em C#, parte 4/7.

```
C:\repos\RHS_Folha_de_Pagamento\Entity\FolhaPagamento.cs
137     [0].ToString());
138         this._item.SubItems.Add(valoresFolha.valor
139         [0].ToString("0.00"));
140         this._item.SubItems.Add("");
141         lancamentos.Items.Add(this._item);
142     }
143     else
144     {
145         MessageBox.Show("Remuneração Base não entra na
146 cota do Salário Família");
147     }
148     break;
149     case "faltas":
150         //Chamar form
151         frm_Faltas faltas = new frm_Faltas();
152         faltas.set_RemunBase = salario.ToString("0.00");
153         faltas.ShowDialog();
154
155         //Receber os dados do usuarios e converter para
156         double
157         double quantDias = double.Parse
158         (faltas.get_Dias.Text);
159         double quantFaltas = double.Parse
160         (faltas.get_Faltas.Text);
161
162         //Realizar o calculo
163
164         double descontoFalta = (salario_Bruto / quantDias)
165         * quantFaltas;
166
167         //Alimentar os itens do listView
168         this._item = new ListViewItem("DESC. FALTAS");
169         this._item.SubItems.Add(faltas.get_Faltas.Text);
170         this._item.SubItems.Add("");
171         this._item.SubItems.Add(descontoFalta.ToString
172         ("0.00"));
173         lancamentos.Items.Add(this._item);
174
175         break;
176     case "atrasos":
177         //Chamar form
178         frm_Atrasos atrasos = new frm_Atrasos();
179         atrasos.set_RemunBase = salario.ToString("0.00");
180         atrasos.set_cargaHoraria = cargaHoraMensal.ToString
181         ();
182         atrasos.ShowDialog();
183
184         //Receber os dados do usuarios e converter para
185         double
186         double quantHoras = double.Parse
187         (atrasos.get_Atrasos.Text);
188
189         //Realizar o calculo
190
191         double descontoAtraso = (salario_Bruto / quantHoras)
192         * quantHoras;
193
194         //Alimentar os itens do listView
195         this._item = new ListViewItem("DESC. ATASOS");
196         this._item.SubItems.Add(atrasos.get_Atrasos.Text);
197         this._item.SubItems.Add("");
198         this._item.SubItems.Add(descontoAtraso.ToString
199         ("0.00"));
200         lancamentos.Items.Add(this._item);
201
202         break;
203     default:
204         MessageBox.Show("Opção Inválida");
205     }
206 }
```

Fonte: Próprios autores, 2023.

Figura 48: Classe FolhaPagamento desenvolvida em C#, parte 5/7.

```
C:\repos\RHS_Folha_de_Pagamento\Entity\FolhaPagamento.cs 5
179         // Realizar o calculo
180
181         double descontoAtrasos = (salario_Bruto /
cargaHoraMensal) * quantHoras;
182
183
184         //Alimentar os itens do listView
185         this._item = new ListViewItem("DESC. HORAS
INJUSTIFICADAS");
186         this._item.SubItems.Add(atrasos.get_Atrasos.Text);
187         this._item.SubItems.Add("");
188         this._item.SubItems.Add(descontoAtrasos.ToString
("0.00"));
189         lancamentos.Items.Add(this._item);
190         break;
191     case "planosaude":
192         //Chamar form
193         frm_PlanoSaude ps = new frm_PlanoSaude(salario);
194         ps.ShowDialog();
195
196         //Receber os dados do usuarios e converter para
double
197         double valorPlano = double.Parse
(ps.get_valor.Text);
198
199         //Alimentar os itens do listView
200         this._item = new ListViewItem("DESC. PLANO DE
SAÚDE");
201         this._item.SubItems.Add("");
202         this._item.SubItems.Add("");
203         this._item.SubItems.Add(valorPlano.ToString
("0.00"));
204         lancamentos.Items.Add(this._item);
205         break;
206     case "outrosded":
207         //Chamar form
208         frm_Otras outrasDeducoes = new frm_Otras();
209         outrasDeducoes.ShowDialog();
210
211         //Receber os dados do usuarios e converter para
double
212         double outrosValores = double.Parse
(outrasDeducoes.get_valor.Text);
213
214         //Alimentar os itens do listView
215         this._item = new ListViewItem
(outrasDeducoes.get_descricao.Text.ToUpper());
216         this._item.SubItems.Add("");
217         this._item.SubItems.Add("");
218         this._item.SubItems.Add(outrosValores.ToString
("0.00"));
219         lancamentos.Items.Add(this._item);
220         break;
```

Fonte: Próprios autores, 2023.

Figura 49: Classe FolhaPagamento desenvolvida em C#, parte 6/7.

```
C:\repos\RHS_Folha_de_Pagamento\Entity\FolhaPagamento.cs
221      case "outrosrem":
222          //Chamar form
223          frm_Outras outrasRemuneracoes = new frm_Outras();
224          outrasRemuneracoes.ShowDialog();
225
226          //Receber os dados do usuarios e converter para
227          double
228          double outrosValoresRem = double.Parse
229          (outrasRemuneracoes.get_valor.Text);
230
231          //Alimentar os itens do listView
232          this._item = new ListViewItem
233          (outrasRemuneracoes.get_descricao.Text.ToUpper());
234          this._item.SubItems.Add("");
235          this._item.SubItems.Add(outrosValoresRem.ToString("0.00"));
236          this._item.SubItems.Add("");
237          lancamentos.Items.Add(this._item);
238          break;
239      case "adicionais":
240          //Chamar form
241          frm_Adicionais adicionais = new frm_Adicionais();
242          adicionais.set_RemunBase = salario.ToString
243          ("0.00");
244          adicionais.set_cargaHoraria =
245          cargaHoraMensal.ToString();
246          adicionais.ShowDialog();
247
248          double valorInsalubre = 0;
249          double valorNoturno = 0;
250
251          //calculo insalubridade
252
253          if (adicionais.get_valor.Text != string.Empty)
254          {
255              valorInsalubre = salario * ((double.Parse
256              (adicionais.get_valor.Text) / 100));
257          }
258
259          if(valorInsalubre > 0)
260          {
261              //Alimentar os itens do listView
262              this._item = new ListViewItem("INSALUBRIDADE "
263              + adicionais.get_valor.Text + "%");
264              this._item.SubItems.Add
265              (adicionais.get_valor.Text + "%");
266              this._item.SubItems.Add(valorInsalubre.ToString
267              ("0.00"));
268              this._item.SubItems.Add("");
269              lancamentos.Items.Add(this._item);
270          }
271
272          //calculo adicional noturno
273
```

Fonte: Próprios autores, 2023.

Figura 50: Classe FolhaPagamento desenvolvida em C#, parte 7/7.

```
C:\repos\RHS_Folha_de_Pagamento\Entity\FolhaPagamento.cs
264         if ((adicionais.get_cargaHoraria.Text !=      ↵
265             string.Empty) &&           ↵
266                 (adicionais.get_horas.Text != string.Empty) &&    ↵
267                     (adicionais.get_percentual.Text !=      ↵
268                         string.Empty) &&           ↵
269                             (adicionais.get_horas.Text != string.Empty))   ↵
270     {
271         this.horas_Trab = double.Parse            ↵
272             (adicionais.get_cargaHoraria.Text);    ↵
273         this.percentualHE = double.Parse        ↵
274             (adicionais.get_percentual.Text);       ↵
275
276         valorNoturno = (((salario / horas_Trab) *      ↵
277             (percentualHE / 1000)) + (salario / horas_Trab)) *    ↵
278             double.Parse(adicionais.get_horas.Text));          ↵
279     }
280
281     if (valorNoturno > 0)
282     {
283         //Alimentar os itens do listView
284         this._item = new ListViewItem("ADICIONAL.      ↵
285             NOTURNO " + percentualHE + "%");
286         this._item.SubItems.Add
287             (adicionais.get_horas.Text + "h");
288         this._item.SubItems.Add(valorNoturno.ToString      ↵
289             ("0.00"));
290         this._item.SubItems.Add("");
291         lancamentos.Items.Add(this._item);
292     }
293
294     break;
295 }
296 }
297 }
```

Fonte: Próprios autores, 2023.

O código apresentado acima, representa a classe chamada "FolhaPagamento" que estar envolvida no cálculo de detalhes salariais e de benefícios para funcionários. Ela tem várias propriedades relacionadas a informações de pagamento, como carga horária mensal, horas trabalhadas, salário líquido, salário bruto, valor do FGTS e valor do Salário Família.

A classe possui um método chamado "calcularFolha" que aceita um parâmetro "calcultoTipo" para determinar que tipo de cálculo de pagamento ou dedução deve ser realizado. Dependendo do valor desse parâmetro, a função realiza cálculos específicos relacionados a horas extras, vale-transporte, FGTS, salário família, faltas, atrasos, plano de saúde e outras deduções e remunerações adicionais.

Essa classe faz parte do sistema de folha de pagamento, código também interage com janelas de diálogo ("frm_HoraExtras", "frm_ValeTransporte", etc.) para receber informações dos usuários e, em seguida, realiza os cálculos com base nesses dados para gerar detalhes da folha de pagamento.

A seguir falaremos sobre a classe HomePage em linguagem Dart, no framework Flutter que usamos para desenvolvimento do App Registro de Ponto.

Figura 51: Classe HomePage desenvolvida em Dart no Flutter, parte 1/11.

```

1  import 'package:firebase_auth/firebase_auth.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter/services.dart';
4  import 'package:folha_de_ponto_rhs/pages/banco_bados.dart';
5  import 'package:folha_de_ponto_rhs/widgets/auth_check.dart';
6  import 'package:folha_de_ponto_rhs/widgets/relogio.dart';
7  import 'package:folha_de_ponto_rhs/widgets/usuario_logado.dart';
8
9  class HomePage extends StatelessWidget {
10    const HomePage({super.key});
11
12    @override
13    Widget build(BuildContext context) {
14      return MaterialApp(
15        debugShowCheckedModeBanner: false,
16        title: "Registro de Ponto",
17        theme: ThemeData(
18          primaryColor: Colors.blue,
19        ), // ThemeData
20        home: const WidgetHome(),
21      ); // MaterialApp
22    }
23  }
24
25  class WidgetHome extends StatefulWidget {
26    const WidgetHome({super.key});
27
28    @override
29    State<WidgetHome> createState() => _WidgetLoginState();
30  }
31
32  class _WidgetLoginState extends State<WidgetHome> {
33    //controle dos botões
34    bool _entradaButtonDisabled = false;
35    bool _saidaButtonDisabled = false;
36    bool _pausaButtonDisabled = false;
37    bool _retornoButtonDisabled = false;
38
39    final FirebaseAuth _auth = FirebaseAuth.instance;
40

```

Fonte: Próprios autores, 2023.

Figura 52: Classe HomePage desenvolvido em Dart no Flutter, parte 2/11.

```

41 //acessando banco de dados
42 final postgresService = PostgreSQService(
43   host: 'berry.db.elephantsql.com',
44   port: 5432, // Porta padrão do PostgreSQL
45   databaseName: 'kwlrlitm',
46   username: 'kwlrlitm',
47   password: 'XSvdSkoe1d8aCwESg7DQsJ7ay6ADyEz5',
48 );
49 //logout
50 Future<void> _signOut(BuildContext context) async {
51   await _auth.signOut();
52   // ignore: use_build_context_synchronously
53   Navigator.of(context).pushReplacement(
54     MaterialPageRoute(builder: (context) => const AuthCheck()),
55   );
56 }
57
58 TimeOfDay obterHoraAtual() {
59   final agora = DateTime.now();
60   return TimeOfDay(hour: agora.hour, minute: agora.minute);
61 }
62
63 void registrarPonto(String tipo) async {
64   //metodo pega o id na tabela funcionario no banco de dados,
65   //atraves do email do usuario logado, e passa para outro metodo
66   //que insere as informacoes
67   //do registro de ponto na tabela registro de ponto
68   // Crie uma instância da classe UserEmailWidget
69   // Crie uma instância da classe UserEmailWidget
70   DateTime dataAtual = DateTime.now();
71   int ano = dataAtual.year;
72   int mes = dataAtual.month;
73   int dia = dataAtual.day;
74   String data = '$dia/$mes/$ano';
75
76   String? emailDoUsuario;

```

Fonte: Próprios autores, 2023.

Figura 53: Classe HomePage desenvolvido em Dart no Flutter, parte 3/11.

```

78 // Verifica se o usuário está autenticado e pega o email logado
79 User? usuario = FirebaseAuth.instance.currentUser;
80 if (usuario != null) {
81   emailDoUsuario = usuario.email;
82 } else {
83   emailDoUsuario = null;
84 }
85 try {
86   int? idFuncionario =
87     await postgresService.getIdFuncionario(emailDoUsuario.toString());
88   if (idFuncionario != null) {
89     await postgresService.inserirRegistroPonto(
90       idFuncionario, data, obterHoraAtual(), tipo);
91   } else {
92     print('Email não encontrado no banco de dados.');
93   }
94 } catch (e) {
95   print('Erro ao se conectar no banco de dados: $e');
96 }
97
98 void _handleMenuItemClick(String value) {
99   // Implemente a lógica para cada opção do menu aqui
100  if (value == 'Sair') {
101    SystemNavigator.pop();
102  } else if (value == 'Logout') {
103    _signOut(context);
104  } else if (value == 'Sobre') {
105    _showVersionNotesDialog(context);
106  }
107 }
108

```

Fonte: Próprios autores, 2023.

Figura 54: Classe HomePage desenvolvido em Dart no Flutter, parte 4/11.

```

110 void _showVersionNotesDialog(BuildContext context) {
111   showDialog(
112     context: context,
113     builder: (BuildContext context) {
114       return AlertDialog(
115         title: const Text('Notas da Versão 1.0.0'),
116         content: const SingleChildScrollView(
117           child: Text(
118             'Este aplicativo foi desenvolvido pela Devcoast.',
119           ),
120         ),
121         actions: <Widget>[
122           TextButton(
123             onPressed: () {
124               Navigator.of(context).pop();
125             },
126             child: const Text('Fechar'),
127           ),
128         ],
129       );
130     },
131   );
132 }
133
134 void _showSnackBar(BuildContext context, String message) {
135   ScaffoldMessenger.of(context).showSnackBar(
136     SnackBar(
137       content: Text(message),
138       duration: const Duration(seconds: 2), // Defina a duração desejada
139     ),
140   );
141 }
```

Fonte: Próprios autores, 2023.

Figura 55: Classe HomePage desenvolvido em Dart no Flutter, parte 5/11.

```

143 @override
144 Widget build(BuildContext context) {
145   return Scaffold(
146     appBar: AppBar(
147       title: const Text(
148         'Registro de Ponto RHS',
149       ),
150       actions: <Widget>[
151         PopupMenuButton<String>(
152           onSelected: _handleMenuItemClick,
153           itemBuilder: (BuildContext context) {
154             return {'Sair', 'Logout', 'Sobre'}.map((String choice) {
155               return PopupMenuItem<String>(
156                 value: choice,
157                 child: Text(choice),
158               );
159             }).toList();
160           },
161         ),
162       ],
163     ),
164     body: Column(
165       mainAxisAlignment:
166         MainAxisAlignment.start, // Alinha os elementos no topo.
167       children: <Widget>[
168         const SizedBox(height: 50.0),
169         Row(
170           mainAxisAlignment: MainAxisAlignment.center,
171           children: <Widget>[
172             UserEmailWidget(),
173             const SizedBox(
174               height: 50.0), // Espaçamento entre o texto e o DigitalClock. // SizedBox
175           ],
176         ),
177         Center(
178           child: DigitalClock(),
179         ),
180         const SizedBox(
181           height: 50.0), // Espaçamento entre o texto e o GridView. // SizedBox
182       ],
183     ),
184   );
185 }
```

Fonte: Próprios autores, 2023.

Figura 56: Classe HomePage desenvolvido em Dart no Flutter, parte 6/11.

```

182     ↴ Expanded(
183       ↳ child: GridView.count(
184         padding: const EdgeInsets.all(10.0),
185         crossAxisCount: 2,
186         mainAxisSpacing: 10.0,
187         crossAxisSpacing: 10.0,
188         children: <Widget>[
189           //botão Entrada
190           ElevatedButton(
191             onPressed: _entradabuttonDisabled
192               ? null
193               : () {
194                 setState(() {
195                   // Desativar o botão
196                   _entradabuttonDisabled = true;
197                 });
198
199                 registrarPonto('Entrada');
200                 _showSnackBar(context, 'Entrada Registrada.');
201               },
202             style: ElevatedButton.styleFrom(
203               foregroundColor: Colors.white,
204               backgroundColor:
205                 Colors.green, // Define a cor do texto do botão
206             ),
207             child: _entradabuttonDisabled
208               ? Row(
209                 mainAxisSize: MainAxisSize.min,
210                 children: [
211                   Image.asset(
212                     "imagens/check_icon.png",
213                     fit: BoxFit.contain,
214                   ), // Image.asset
215                   const SizedBox(width: 8),
216                   const Text(
217                     'Entrada\n Registrada',
218                     textAlign: TextAlign.center,
219                     style: TextStyle(
220                       color: Color.fromARGB(255, 127, 127,
221                         129), // Defina a cor desejada aqui // Color.fromARGB
222                         fontSize: 15, // Tamanho da fonte
223                         fontWeight:
224                           FontWeight.bold, // Peso da fonte (opcional)
225                     ), // TextStyle
226                   ), // Text
227               ],
228             );
229           ),
230         ],
231       ),
232     );
233   );
234 
```

Fonte: Próprios autores, 2023.

Figura 57: Classe HomePage desenvolvido em Dart no Flutter, parte 7/11.

```

228   ↳ ),
229   ↳ : Row(
230     ↳ children: [
231       ↳ const Text(
232         'Entrada',
233         style: TextStyle(
234           color: Colors.white,
235           fontSize: 15.0,
236         ), // TextStyle
237         ),
238         ↳ const SizedBox(width: 20.0),
239         ↳ Image.asset([
240           "imagens/entrada_icon.png",
241             fit: BoxFit.contain,
242           ],
243         ),
244       ), // Row
245     ), // ElevatedButton
246     //botão Pausa
247     ElevatedButton(
248       onPressed: _pausabuttonDisabled
249         ? null
250         : () {
251           setState(() {
252             // Desativar o botão
253             _pausabuttonDisabled = true;
254           });
255
256           registrarPonto('Pausa');
257           _showSnackBar(context, 'Pausa Registrada.');
258         },
259       style: ElevatedButton.styleFrom(
260         foregroundColor: Colors.white,
261         backgroundColor:
262           Colors.red, // Define a cor do texto do botão
263       ),
264     );
265   );
266 
```

Fonte: Próprios autores, 2023.

Figura 58: Classe HomePage desenvolvido em Dart no Flutter, parte 8/11.

```

  263 ),
  264 child: _pausaButtonDisabled
  265   : Row(
  266     mainAxisSize: MainAxisSize.min,
  267     children: [
  268       Image.asset(
  269         "imagens/check_icon.png",
  270         fit: BoxFit.contain,
  271       ), // Image.asset
  272       const SizedBox(width: 8),
  273       const Text(
  274         'Pausa\n Registrada',
  275         textAlign: TextAlign.center,
  276         style: TextStyle(
  277           color: Color.fromARGB(255, 127, 129,
  278             129), // Defina a cor desejada aqui // Color.fromRGB
  279           fontSize: 15, // Tamanho da fonte
  280           fontWeight:
  281             FontWeight.bold, // Peso da fonte (opcional)
  282         ), // TextStyle
  283       ],
  284     ), // Row
  285   : Row(
  286     children: [
  287       const Text(
  288         'Pausa ',
  289         style: TextStyle(
  290           color: Colors.white,
  291           fontSize: 15.0,
  292         ), // TextStyle
  293       ),
  294       const SizedBox(width: 20.0),
  295       Image.asset(
  296         "imagens/pausa_icon.png",
  297         fit: BoxFit.contain,
  298       ), // Image.asset
  299     ],
  300   ), // Row
  301 ), // ElevatedButton
  302

```

Fonte: Próprios autores, 2023.

Figura 59: Classe HomePage desenvolvido em Dart no Flutter, parte 9/11.

```

 304 //botão Retorno
 305 ElevatedButton(
 306   onPressed: _retornoButtonDisabled
 307     ? null
 308     : () {
 309       setState(() {
 310         // Desativar o botão
 311         _retornoButtonDisabled = true;
 312       });
 313
 314       registrarPonto('Retorno');
 315       _showSnackBar(context, 'Retorno Registrado.');
 316     },
 317     style: ElevatedButton.styleFrom(
 318       foregroundColor: Colors.white,
 319       backgroundColor:
 320         Colors.yellow[800], // Define a cor do texto do botão
 321     ),
 322     child: _retornoButtonDisabled
 323     ? Row(
 324       mainAxisSize: MainAxisSize.min,
 325       children: [
 326         Image.asset(
 327           "imagens/check_icon.png",
 328           fit: BoxFit.contain,
 329         ), // Image.asset
 330         const SizedBox(width: 8),
 331         const Text(
 332           'Retorno\n Registrado',
 333           textAlign: TextAlign.center,
 334           style: TextStyle(
 335             color: Color.fromARGB(255, 127, 129,
 336               129), // Defina a cor desejada aqui // Color.fromRGB
 337             fontSize: 15, // Tamanho da fonte
 338             fontWeight:
 339               FontWeight.bold, // Peso da fonte (opcional)
 340             ), // TextStyle
 341           ],
 342         ) // Row
 343

```

Fonte: Próprios autores, 2023.

Figura 60: Classe HomePage desenvolvido em Dart no Flutter, parte 10/11.

```

344   : Row(
345     children: [
346       const Text(
347         'Retorno',
348         style: TextStyle(
349           color: Colors.white,
350           fontSize: 15.0,
351         ), // TextStyle
352       ), // Text
353       const SizedBox(width: 20.0),
354       Image.asset(
355         "imagens/retorno_icon.png",
356         fit: BoxFit.contain,
357       ), // Image.asset
358     ],
359   ), // Row
360 ), // ElevatedButton
361 //botão Saida
362 ElevatedButton(
363   onPressed: _saidaButtonDisabled
364     ? null
365     : () {
366       setState(() {
367         // Desativar o botão
368         _saidaButtonDisabled = true;
369       });
370
371       registrarPonto('Saida');
372       _showSnackBar(context, 'Saida Registrada.');
373     },
374   style: ElevatedButton.styleFrom(
375     foregroundColor: Colors.white,
376     backgroundColor:
377       Colors.lightBlue[400], // Define a cor do texto do botão
378   ),
379   child: _saidaButtonDisabled
380     ? Row(
381       mainAxisSize: MainAxisSize.min,
382       children: [
383         Image.asset(
384           "imagens/check_icon.png",
385           fit: BoxFit.contain,
386         ), // Image.asset
387         const SizedBox(width: 8),
388       ],
389     ) // Row
390   ), // ElevatedButton
391 );
392 
```

Fonte: Próprios autores, 2023.

Figura 61: Classe HomePage desenvolvido em Dart no Flutter, parte 11/11.

```

388   const Text(
389     'Saida\n Registrada',
390     textAlign: TextAlign.center,
391     style: TextStyle(
392       color: Color.fromARGB(255, 127, 129,
393         129), // Defina a cor desejada aqui // Color.fromARGB
394       fontSize: 15, // Tamanho da fonte
395       fontWeight:
396         FontWeight.bold, // Peso da fonte (opcional)
397     ), // TextStyle
398   ], // Text
399 ), // Row
400 Row(
401   children: [
402     const Text(
403       'Saida ',
404       style: TextStyle(
405         color: Colors.white,
406         fontSize: 15.0,
407       ), // TextStyle
408       const SizedBox(width: 20.0),
409       Image.asset(
410         "imagens/saida_icon.png",
411         fit: BoxFit.contain,
412       ), // Image.asset
413     ],
414   ), // Row
415 ), // ElevatedButton
416 ], // <Widget>[]
417 ), // GridView.count
418 ), // Expanded
419 ], // <Widget>[]
420 ), // Column
421 ); // Scaffold
422 }
423 
```

Fonte: Próprios autores, 2023.

Os códigos apresentados anteriormente:

- Importa bibliotecas necessárias, incluindo "firebase_auth" para autenticação, "material.dart" para widgets do Material Design, "folha_de_ponto_rhs/pages/banco_bados.dart" para acesso a um banco de dados, e vários widgets personalizados;
- Define a classe "HomePage" como um estado inalterável ("StatelessWidget") e configura a aparência do aplicativo, incluindo o título e a aparência visual, como a cor principal;
- Dentro da classe "HomePage", define a classe "WidgetHome" como um estado mutável (" StatefulWidget"). Esta classe é responsável por criar a interface do usuário principal da página inicial;
- Cria botões para "Entrada", "Pausa", "Retorno" e "Saída" no layout da página, com a possibilidade de registrar eventos de ponto, como entrada e saída. Os botões desabilitam-se após o primeiro registro bem-sucedido para evitar registros duplicados;
- Implementa funcionalidades de registro de ponto, que envolvem a obtenção da hora atual, o acesso a informações do usuário logado, o registro de ponto no banco de dados e o tratamento de erros de conexão com o banco;
- Implementa um menu de contexto que permite ao usuário sair do aplicativo, fazer logout e acessar informações "Sobre" em um diálogo;
- Define a exibição de informações como o Email do usuário logado e um relógio digital;
- Configura a aparência visual da página, incluindo a barra de aplicativos, a exibição de botões, e a exibição de informações.

Em resumo, este código cria a interface do aplicativo de registro de ponto que permite aos usuários registrar entradas, saídas e outros eventos relacionados ao trabalho, com a capacidade de fazer logout. Além disso, o código interage com um banco de dados para registrar as informações de ponto dos funcionários.

E por fim os códigos a seguir fazem parte de um componente React chamado "Home" que faz parte do sistema de autenticação, que iremos explicar o que ele faz em detalhes:

Importações:

- O código importa vários módulos e pacotes, incluindo useState, Link, auth de um arquivo de conexão Firebase, signInWithEmailAndPassword do FirebaseAuth, useNavigate e alguns recursos de estilo.

Componente "Home":

- Define um componente de função chamado "Home" que renderiza uma página de login;
- Inicializa dois estados (email e password) usando o hook useState para armazenar o email e a senha digitados pelo usuário;
- Obtém a função navigate do React Router para navegar entre as páginas.

Função handleLogin:

Uma função assíncrona que lida com a submissão do formulário de login quando o usuário pressiona o botão "Logar".

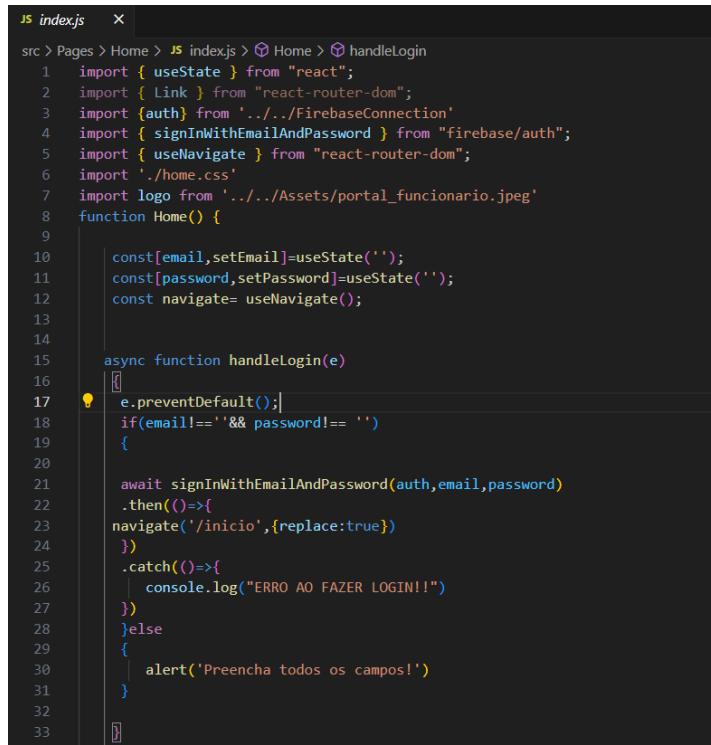
- Ele verifica se o email e a senha não estão vazios;
- Se ambos os campos estiverem preenchidos, ele usa o signInWithEmailAndPassword do Firebase para autenticar o usuário com o email e a senha fornecidos;
- Se a autenticação for bem-sucedida, ele redireciona o usuário para a página "/início";
- Se a autenticação falhar, exibe uma mensagem de erro no console.

Renderização:

- Renderiza um formulário com campos de entrada para email e senha;
- Quando o usuário insere texto nos campos de entrada, os estados email e password são atualizados usando o onChange para acompanhar as mudanças;
- Um botão "Logar" para enviar o formulário.

Em resumo, este código cria a página de login do Portal do Funcionário, que permite aos usuários inserirem seu Email e senha para autenticação. Se os campos não estiverem vazios e a autenticação for bem-sucedida, o usuário é redirecionado para a página "/inicio". Caso contrário, uma mensagem de erro é exibida no console.

Figura 62: Index.js do componente Home, desenvolvido em JavaScript, parte 1/2.



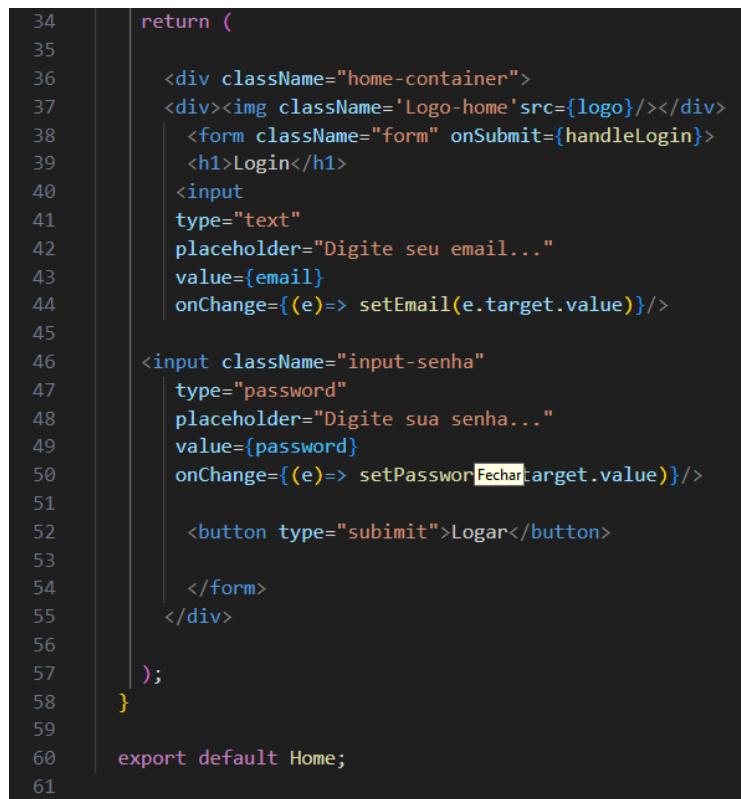
```

JS index.js ✘
src > Pages > Home > JS index.js > ⚡ Home > ⚡ handleLogin
1 import { useState } from "react";
2 import { Link } from "react-router-dom";
3 import { auth } from "../../FirebaseConnection"
4 import { signInWithEmailAndPassword } from "firebase/auth";
5 import { useNavigate } from "react-router-dom";
6 import './home.css'
7 import logo from '../../../../../Assets/portal_funcionario.jpeg'
8 function Home() {
9
10    const [email, setEmail] = useState('');
11    const [password, setPassword] = useState('');
12    const navigate = useNavigate();
13
14
15    async function handleLogin(e)
16    [
17        e.preventDefault();
18        if(email !== '' && password !== '')
19        {
20
21            await signInWithEmailAndPassword(auth, email, password)
22            .then(() =>
23                navigate('/inicio', {replace: true})
24            )
25            .catch(() =>
26                console.log("ERRO AO FAZER LOGIN!!")
27            )
28        }
29        else
30        {
31            alert('Preencha todos os campos!')
32        }
33    ]

```

Fonte: Próprios autores, 2023.

Figura 63: Index.js do componente Home, desenvolvido em JavaScript, parte 2/2.



```

34    return (
35
36        <div className="home-container">
37            <div><img className='Logo-home' src={logo}/></div>
38            <form className="form" onSubmit={handleLogin}>
39                <h1>Login</h1>
40                <input
41                    type="text"
42                    placeholder="Digite seu email..."
43                    value={email}
44                    onChange={(e) => setEmail(e.target.value)} />
45
46                <input className="input-senha"
47                    type="password"
48                    placeholder="Digite sua senha..."
49                    value={password}
50                    onChange={(e) => setPassword(e.target.value)} />
51
52                <button type="submit">Logar</button>
53
54            </form>
55        </div>
56    );
57}
58
59
60 export default Home;
61

```

Fonte: Próprios autores, 2023.

5 CONCLUSÃO

O desafio de desenvolver um sistema integrado de folha de pagamento foi realizada com notável eficiência, pois desenvolvemos um sistema que automatiza e simplifica o processo de cálculo da folha de pagamento, garantindo conformidade legal e precisão dos dados.

Em resumo, a parceria estratégica entre a RHS e a Devcoast se revelou fundamental para a aprimoração dos processos internos da empresa. Ao alinhar o projeto à aplicação prática dos conhecimentos adquiridos em sala de aula, este trabalho propôs uma abordagem holística, incorporando não apenas a expertise técnica, mas também uma compreensão profunda dos processos e das necessidades específicas do modelo de negócio da empresa.

Em última análise, este trabalho não se limita a ser uma resposta eficaz aos desafios específicos enfrentados pela RHS; ele também contribui de maneira significativa para o avanço do campo de tecnologia aplicada à gestão de Recursos Humanos. Sua implementação promove eficiência, integração e precisão nos processos organizacionais, consolidando-se como uma peça-chave no progresso contínuo da empresa no cenário tecnológico.

REFERÊNCIAS

Editorial do Indeed. 2023. Guia de Carreira. *Indeed*. [Online] 19 de março de 2023. [Citado em: 27 de maio de 2023.] <https://br.indeed.com/conselho-de-carreira/encontrando-emprego/definicao-de-recursos-humanos>.

Equipe TOTVS. 2023. Folha de pagamento: Guia completo sobre o tema! *Eleve TOTVS*. [Online] 12 de janeiro de 2023. [Citado em: 27 de maio de 2023.] <https://elevessuavendas.com.br/blog/financeiro/folha-de-pagamento>.

Furtado, Marcelo. 2022. Quais são os descontos em folha de pagamento? Conheça os 7 principais. *Covenia*. [Online] 2022. [Citado em: 27 de maio de 2023.] <https://blog.convenia.com.br/descontos-da-folha-de-pagamento/>.

Guimarães, Bruna. 2023. Departamento pessoal (DP): veja rotinas, funções e como funciona. *Gupy*. [Online] 02 de maio de 2023. [Citado em: 27 de maio de 2023.] <https://www.gupy.io/blog/departamento-pessoal#:~:text=O%20Departamento%20Pessoal%20%C3%A9%20a,pagamento%2C%20benef%C3%ADcios%2C%20entre%20outros>.

Marques, José Carlos. 2016. *Gestão de Recursos Humanos*. São Paulo: Cengage Learning Edições Ltda. , 2016. ISBN 978-85-221-2359-9.

Morgana Perini, Caroline Lorensi da Silva. 2023. Folha de pagamento: o que é, seu cálculo e desoneração. *Metadados RH Em Sintonia*. [Online] 15 de fevereiro de 2023. [Citado em: 27 de maio de 2023.] <https://www.metadados.com.br/blog/folha-de-pagamento-o-que-e-seu-calcular-e-desoneracao>.

NBR 9241-11. 2002. NBR 9241-11. *Requisitos Ergonômicos para Trabalho de Escritórios com Computadores Parte 11 – Orientações sobre Usabilidade*. [Documento]. Rio de Janeiro, Rio de Janeiro, Brasil: ABNT–Associação Brasileira de Normas Técnicas, Agosto de 2002. p. 21.

Senior. 2022. Rotinas do departamento pessoal: descubra quais são e como otimizá-las! *Senior Blog*. [Online] 09 de dezembro de 2022. [Citado em: 27 de maio de 2023.] <https://www.senior.com.br/blog/rotinas-do-departamento-pessoal-descubra-quais-sao-e-como-otimiza-las>.

Agência Senado. 2022. Lei Geral de Proteção de Dados entra em vigor
Fonte: Agência Senado. [Online] 24 de outubro de 2023. [Citado em: 24 de outubro de 2023.] <https://www.senior.com.br/blog/rotinas-do-departamento-pessoal-descubra-quais-sao-e-como-otimiza-las>.

ANEXO A - TERMO DE CONSENTIMENTO PARA TRATAMENTO DE DADOS PESSOAIS LEI GERAL DE PROTEÇÃO DE DADOS PESSOAIS – LGPD

Este documento visa registrar a manifestação livre, informada e inequívoca pela qual o Titular concorda com o tratamento de seus dados pessoais para finalidade específica, em conformidade com a Lei nº 13.709 – Lei Geral de Proteção de Dados Pessoais (LGPD).

Ao manifestar sua aceitação para com o presente termo, o Titular consente e concorda que a empresa DEVCOAST, CNPJ nº xx.xxx.xxx/xxxx-xx, doravante denominada Controladora, tome decisões referentes ao tratamento de seus dados pessoais, dados referentes as empresas em que atuem os usuários ou dados necessários ao usufruto de serviços ofertados pela DEVCOAST, bem como realize o tratamento de tais dados, envolvendo operações como as que se referem a coleta, produção, recepção, classificação, utilização, acesso, reprodução, transmissão, distribuição, processamento, arquivamento, armazenamento, eliminação, avaliação ou controle da informação, modificação, comunicação, transferência, difusão ou extração.

Dados Pessoais

A Controladora fica autorizada a tomar decisões referentes ao tratamento e a realizar o tratamento dos seguintes dados pessoais do Titular:

- Nome completo;
- Nome empresarial;
- Data de nascimento;
- Número e imagem da Carteira de Identidade (RG);
- Número e imagem do Cadastro de Pessoas Físicas (CPF);
- Número e imagem da Carteira Nacional de Habilitação (CNH);
- Número do Cadastro Nacional de Pessoas Jurídicas (CNPJ);
- Endereço completo;
- Números de telefone, WhatsApp e endereços de e-mail;
- Banco, agência e número de contas bancárias;
- Nome de usuário e senha específicos para uso dos serviços do Controlador;
- Comunicação, verbal e escrita, mantida entre o Titular e o Controlador.

Finalidades do Tratamento dos Dados

O tratamento dos dados pessoais listados neste termo tem as seguintes finalidades:

- Possibilitar que a Controladora identifique e entre em contato com o Titular para fins de relacionamento comercial;
- Possibilitar que a Controladora elabore contratos comerciais e emita cobranças contra o Titular;
- Possibilitar que a Controladora envie ou forneça ao Titular seus produtos e serviços, de forma remunerada ou gratuita;
- Possibilitar que a Controladora estruture, teste, promova e faça propaganda de produtos e serviços, personalizados ou não ao perfil do Titular.

- Possibilitar que a Controladora utilize tais dados em Pesquisas de Mercado;
- Possibilitar que a Controladora utilize tais dados na inscrição, divulgação, premiação dos interessados participantes de Eventos, Prêmios ou Concursos;
- Possibilitar que a Controladora utilize tais dados na elaboração de catálogos e na Curadoria de autores;
- Possibilitar que a Controladora utilize tais dados na elaboração de relatórios e emissão de produtos e serviços;
- Possibilitar que a Controladora utilize tais dados para suas peças de Comunicação;
- Possibilitar que a Controladora utilize tais dados emissão de Notas Fiscais e documentos financeiros correlatos;
- Possibilitar que a Controladora utilize tais dados para facilitar a prestação de serviços diversos além dos primariamente contratados, desde que o cliente também demonstre interesse em contratar novos serviços;
- Possibilitar que a Controladora utilize tais dados para manter banco de dados de profissionais do mercado para facilitar o contato em futuros convites para eventos.

Compartilhamento de Dados

A Controladora fica autorizada a compartilhar os dados pessoais do Titular com outros agentes de tratamento de dados, caso seja necessário para as finalidades listadas neste termo, observados os princípios e as garantias estabelecidas pela Lei nº 13.709.

Segurança dos Dados

A Controladora responsabiliza-se pela manutenção de medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações accidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito.

Em conformidade ao art. 48 da Lei nº 13.709, o Controlador comunicará ao Titular e à Autoridade Nacional de Proteção de Dados (ANPD) a ocorrência de incidente de segurança que possa acarretar risco ou dano relevante ao Titular.

Término do Tratamento dos Dados

A Controladora poderá manter e tratar os dados pessoais do Titular durante todo o período em que eles forem pertinentes ao alcance das finalidades listadas neste termo. Dados pessoais anonimizados, sem possibilidade de associação ao indivíduo, poderão ser mantidos por período indefinido.

O Titular poderá solicitar via e-mail ou correspondência ao Controlador, a qualquer momento, que sejam eliminados os dados pessoais não anonimizados do Titular. O Titular fica ciente de que poderá ser inviável ao Controlador continuar o fornecimento de produtos ou serviços ao Titular a partir da eliminação dos dados pessoais.

Direitos do Titular

O Titular tem direito a obter da Controladora, em relação aos dados por ele tratados, a qualquer momento e mediante requisição: I - confirmação da existência de tratamento; II - acesso aos dados; III - correção de dados incompletos, inexatos ou desatualizados; IV - anonimização, bloqueio ou eliminação de dados desnecessários, excessivos ou tratados em desconformidade com o disposto na Lei nº 13.709; V - portabilidade dos dados a outro fornecedor de serviço ou produto, mediante requisição expressa e observados os segredos comercial e industrial, de acordo com a regulamentação do órgão controlador; VI - portabilidade dos dados a outro fornecedor de serviço ou produto, mediante requisição expressa, de acordo com a regulamentação da autoridade nacional, observados os segredos comercial e industrial; VII - informação das entidades públicas e privadas com as quais o controlador realizou uso compartilhado de dados; VIII - informação sobre a possibilidade de não fornecer consentimento e sobre as consequências da negativa; IX - revogação do consentimento, nos termos do § 5º do art. 8º da Lei nº 13.709.

Direito de Revogação do Consentimento

Este consentimento poderá ser revogado pelo Titular, a qualquer momento, mediante solicitação via e-mail ou correspondência ao Controlador.

Titular: _____ data: _____, de 2023.

ANEXO B – MANUAL DO USUÁRIO