



**UNIP – UNIVERSIDADE PAULISTA PROJETO
INTEGRADO MULTIDISCIPLINAR
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADO MULTIDISCIPLINAR – PIM III
SOFTWARE DE GERENCIAMENTO DE RECURSOS HUMANOS**

**SÃO PAULO
2023**

**UNIP – UNIVERSIDADE PAULISTA PROJETO
INTEGRADO MULTIDISCIPLINAR
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**CAUÃ BELEJ REIGOTA PRADO – RA: G45EBB-6
GABRIEL BELISSIMO SOARES DE LIMA – RA: G420EG-3
JOÃO PEDRO SANTOS DO NASCIMENTO – RA: G4923A-0
MARLON AKIO TOCHIRO – RA: T72740-5**

Projeto Integrado Multidisciplinar
apresentado à Universidade
Paulista - UNIP, curso de Análise
e Desenvolvimento de Sistemas,
como parte dos requisitos
necessários para conclusão do
semestre.

**SÃO PAULO
2023**

CAUÃ BELEJ REIGOTA PRADO – RA: G45EBB-6
GABRIEL BELISSIMO SOARES DE LIMA – RA: G420EG-3
JOÃO PEDRO SANTOS DO NASCIMENTO – RA: G4923A-0
MARLON AKIO TOCHIRO – RA: T72740-5

PROJETO INTEGRADO MULTIDISCIPLINAR – PIM IV
SOFTWARE DE GERENCIAMENTO DE RECURSOS HUMANOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Aprovado em: _____ de _____ de _____.

BANCA EXAMINADORA

Orientador

Universidade Paulista – UNIP

Convidado

Universidade Paulista – UNIP

Convidado

Universidade Paulista UNIP

DEDICATÓRIA

Gostaríamos de dedicar este projeto a todos os estudantes e profissionais de TI (Tecnologia da Informação) em ascensão, para que possa servir de inspiração ou base para um futuro projeto como este.

AGRADECIMENTOS

Agradecemos aos nossos queridos professores por todos os conhecimentos e orientações passados adiante sempre de bom grado e com total dedicação aos alunos. Não conseguiríamos, sem eles, concluir mais este desafio/projeto desta etapa da vida. Somos gratos, também, aos nossos colegas de classe por todo o suporte fornecido nas disciplinas que nos descobríamos com dificuldade.

“Eu acredito que às vezes são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar.”

(Alan Turing)

RESUMO

No contexto empresarial atual, a tecnologia se mostra cada vez mais importante para garantir o crescimento e a evolução das empresas. Nesse sentido, a Dexcode, startup de desenvolvimento de software, oferece soluções inovadoras para diversas áreas do mercado. É possível compreender que as empresas buscam conveniência nos serviços, e a tecnologia é uma das oportunidades excepcionais para proporcionar formas mais eficientes e modernas de gestão, oferecendo segurança e maior controle de prazos com as automatizações. Dessa forma, otimiza-se o tempo e agrega-se valor à experiência do cliente final e aos funcionários da StartRH. O Projeto Integrado Multidisciplinar tem como base o levantamento e análise de requisitos, diagramas, testes e protótipo do sistema.

Palavras-chave: Software, Automatizações, Tecnologia, Requisitos, Diagrama, Teste, Protótipo.

ABSTRACT

In the current business context, technology is increasingly important to ensure the growth and evolution of companies. In this sense, Dexcode, a software development startup, offers innovative solutions for various areas of the market. It is possible to understand that companies seek convenience in services, and technology is one of the exceptional opportunities to provide more efficient and modern forms of management, offering security and greater control of deadlines with automations. Thus, time is optimized and value is added to the experience of the end customer and StartRH employees. The Multidisciplinary Integrated Project is based on the collection and analysis of requirements, diagrams, testing, and a prototype of the system.

Keywords: Software, Automations, Technology, Requirements, Diagram, Test, Prototype.

SUMÁRIO

1 INTRODUÇÃO	13
2 DESENVOLVIMENTO	14
2.1 Cenário	14
2.2 Requisitos.....	15
2.3 Protótipo das Telas	17
2.3.1 Login	18
2.3.2 Home 1	19
2.3.2 A Receber	20
2.3.3 Histórico Salarial.....	21
2.3.4 Em Desenvolvimento	22
2.3.5 Home 2.....	23
2.3.6 Controle Empresas.....	24
2.3.7 Controle de Funcionários	25
2.3.8 Agendar pagamento.....	26
2.3.9 Histórico	27
2.4 UML	28
2.4.1 Casos de Uso.....	29
2.4.2 Diagrama de Atividade	30
2.4.3 Diagrama de Classe.....	33
2.4.4 Diagrama de Sequência	34
2.4.5 Diagrama de Implantação	37
2.5 Banco de Dados	38
2.5.1 Entidade Relacionamento	39
2.5.2 Script de Criação do Banco	40
2.6 Planilha de Testes	41
2.7 Como instalar	43
2.7.1 Rodando o back-end (Servidor)	44
2.7.2 Rodando o front-end (Web)	44
2.8 Comparativo de Mercado	44
2.8.1 TOTVS RM	45
2.8.2 G.Infor.....	45
2.8.3 Bitti	46

3 CONCLUSÃO.....	48
------------------	----

REFERÊNCIAS BIBLIOGRÁFICAS.....	49
---------------------------------	----

LISTA DE ILUSTRAÇÕES

Figuras

Figura 1 - Tela de Login	18
Figura 2 - Tela Home (Funcionário)	19
Figura 3 - Tela de Holerite (A Receber)	20
Figura 4 - Tela para consultar o Histórico Salarial.....	21
Figura 5 - Tela para consultar o Histórico Salarial (Holerite)	21
Figura 6 - Tela Em Desenvolvimento (funcionalidades ainda não entregues)	22
Figura 7 – Tela Home (RH).....	23
Figura 8 - Tela Controle de Empresas.....	24
Figura 9 - Tela Controle de Funcionários	25
Figura 10 - Tela Controle de Funcionários (Modal).....	25
Figura 11 - Tela para Agendar Pagamento.....	26
Figura 12 - Tela para Agendar Pagamento (Modal)	26
Figura 13 - Tela Histórico de Pagamentos.....	27
Figura 14 - Tela Histórico de pagamentos (Modal).....	28
Figura 15 - Caso de Uso (RH).....	29
Figura 16 - Caso de Uso (Funcionário)	30
Figura 17 - Diagrama de Atividade (Gerenciamento de Empresas)	30
Figura 18 - Diagrama de Atividade (Gerenciamento de Funcionários).....	31
Figura 19 - Diagrama de Atividade: Agendar Pagamento	31
Figura 20 - Diagrama de Atividade: Consultar Holerite	32
Figura 21 - Diagrama de Atividade: Consulta de Histórico Salarial.....	32
Figura 22 - Diagrama de Classe	33
Figura 23 - Diagrama de Sequência: Empresa.....	34
Figura 24 - Diagrama de Sequência: Funcionário.....	35
Figura 25 - Diagrama de Sequência: Pagamento	36
Figura 26 - Diagrama de Implantação	37
Figura 27 - Como o prisma.io funciona?	38
Figura 28 - Modelo Entidade Relacionamento.....	39
Figura 29 - TOTVS RM.....	45
Figura 30 – G.Infor.....	45
Figura 31 - Bitti	46

Tabelas

Tabela 1 – Requisitos Funcionais	15
Tabela 2 - Requisitos Não Funcionais.....	16
Tabela 3 - Requisitos de Sistema.....	16
Tabela 4 - Planilha de Testes	42

1 INTRODUÇÃO

O desenvolvimento de software é uma área em constante evolução, e startups como a Dexcode estão cada vez mais presentes no mercado, oferecendo soluções inovadoras para empresas. Neste contexto, o presente trabalho tem como objetivo apresentar o processo de desenvolvimento de um software para automatização de processos de RH, contratado pela empresa StartRH. O projeto será desenvolvido em etapas, utilizando o modelo de entrega em espiral, e neste trabalho serão detalhados os requisitos, protótipo das telas, casos de uso, diagramas de classe e o desenvolvimento do DB (Data Base) com diagrama ER (Entity–Relationship) para a etapa 1 do projeto.

2 DESENVOLVIMENTO

2.1 Cenário

O cenário fornece informações relevantes sobre o contexto em que o projeto está sendo desenvolvido. Essas informações incluem a natureza da empresa cliente, suas necessidades e desafios, os sistemas existentes e suas limitações, as metas do projeto e as etapas de desenvolvimento definidas.

A startup Dexcode, uma empresa focada em desenvolvimento de software para empresas, obteve um novo cliente, uma empresa chamada StartRH que atua realizando as funções de RH para outras empresas.

Iniciamos com um JAD (Joint Application Development) para um levantamento de requisitos inicial, identificamos que a empresa possui diversos sistemas onde nenhum se conversa entre si, levando a realizar cadastro em mais de uma plataforma e alguns controles por meio de planilha de Excel, se tornando o maior motivo de atrasos e erros.

Após diversas reuniões chegamos a um acordo contratual onde optamos pelo modelo de entrega em espiral, o desenvolvimento será da seguinte forma:

- Etapa 1 - Desenvolvimento, homologação e implementação do módulo de automatização de folha de pagamento;
- Etapa 2 - Desenvolvimento, homologação e implementação do módulo de sistema de ponto eletrônico e controle de documentos, e manutenção da folha de pagamento;
- Etapa 3 - Desenvolvimento, homologação e implementação do módulo de sistema de férias e décimo terceiro, e manutenção dos demais módulos;
- Etapa 4 - Manutenção preventiva e corretiva dos módulos.

No PIM III e IV será abordado apenas a etapa 1 deste projeto, no decorrer do documento será detalhado os requisitos deste sistema, protótipo das telas, casos de uso, diagramas de classe, desenvolvimento do DB (Data Base) com diagrama ER (Entity–Relationship).

Tendo em vista este cenário, entendemos a complexidade e a importância do projeto, bem como os riscos e oportunidades envolvidos. Com isso, foi possível

desenvolver uma estratégia de projeto adequada, identificando os requisitos do sistema com precisão e definir as melhores soluções técnicas para atender às necessidades da StartRH.

2.2 Requisitos

A engenharia de requisitos engloba um conjunto de tarefas a serem executadas para gerar como produto final uma documentação de requisitos. Tudo o que estiver contido nos documentos possibilitará que o software seja criado, atualizado e reparado sempre que necessário de acordo com o que foi inicialmente estipulado.

Os requisitos são justamente todas as informações que permitem que qualquer um conheça completamente o software. Além disso, na etapa de levantamento dos requisitos funcionais e não funcionais, é possível identificar quais são os recursos viáveis e inviáveis a serem aplicados.

As planilhas a seguir são fundamentais para o desenvolvimento do software proposto pela empresa Dexcode para o cliente StartRH. Elas apresentam os requisitos funcionais, não funcionais e de sistema, que foram identificados durante o processo de levantamento de requisitos utilizando a metodologia JAD (Joint Application Development).

Tabela 1 – Requisitos Funcionais

REQUISITOS FUNCIONAIS	
RH	FUNCIONÁRIO
Controle de empresas	Consultar previsão de pagamento
Controle de funcionários	Consultar histórico de pagamento
Consultar histórico de pagamento	Mostrar horas trabalhadas
Pagamento individual personalizado	

Fonte: Elaborado pelos autores, 2023.

Tabela 2 - Requisitos Não Funcionais

REQUISITOS NÃO FUNCIONAIS	
RH	FUNCIONÁRIO
Login com senha	Login com senha
Resposta com menos de 15seg.	Resposta com menos de 15seg.
Layout intuitivo	Layout intuitivo
Resposta visual (pop-up)	Resposta visual (pop-up)
	Multiplataformas

Fonte: Elaborado pelos autores, 2023.

Tabela 3 - Requisitos de Sistema

REQUISITOS DE SISTEMA	
RH	FUNCIONÁRIO
Eficácia	Eficácia
Confiabilidade	Confiabilidade
Manutenibilidade	Manutenibilidade
Funcionabilidade	Funcionabilidade
Usabilidade	Usabilidade

Fonte: Elaborado pelos autores, 2023.

- Eficácia: Este projeto foi desenvolvido para atender plenamente ao escopo proposto pelo cliente, incluindo todos os recursos solicitados;
- Confiabilidade: O sistema estará disponível para uso 24 horas por dia, 7 dias por semana, com uma taxa de disponibilidade de 95%. Além disso, foi projetado para apresentar uma taxa de falha de apenas 2% durante consultas com alta latência do cliente;
- Manutenibilidade: O software foi dividido em quatro partes distintas (servidor, desktop, web e mobile), o que permite uma rápida identificação da parte do sistema a ser mantida ou alterada quando necessário;
- Funcionabilidade: O sistema está operando de acordo com o escopo definido previamente com o cliente;
- Usabilidade: Foi aplicado um design focado em UX/UI, tornando o sistema limpo, intuitivo e fácil de usar para os usuários finais.

A partir dessas planilhas, é possível identificar de maneira clara e objetiva os requisitos do sistema a ser desenvolvido, permitindo que a equipe de desenvolvimento da Dexcode possa direcionar seus esforços para atender às necessidades do cliente de maneira eficiente e eficaz. Com isso, espera-se que o software atenda às expectativas do cliente e possa contribuir para a melhoria dos processos de RH.

2.3 Protótipo das Telas

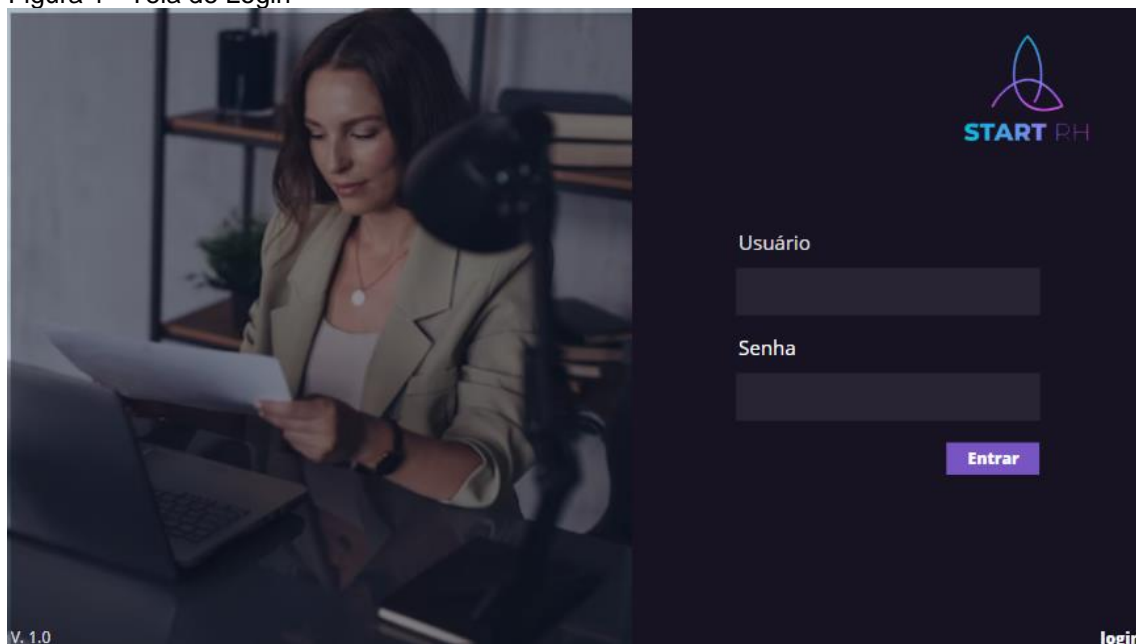
A prototipação é um processo importante no desenvolvimento de software, pois, além de servir como um primeiro rascunho de um produto ou serviço, tem como objetivo amadurecer ideias e engajar pessoas no processo de criação. Esta etapa impacta diretamente na produtividade de toda a equipe e gera valor ao cliente.

O Protótipo de Média Fidelidade demanda um pouco mais de tempo para ser elaborado e está mais próximo do que foi idealizado para o projeto. Geralmente são feitos com o auxílio de softwares e permitem que o usuário simule o comportamento do sistema. Com isso, é possível validar as interações e melhorar a UX (*user experience*). Apesar de ser uma solução mais elaborada, o custo desse tipo de protótipo continua relativamente baixo.

Com base nos requisitos funcionais e não funcionais levantados anteriormente, foram desenvolvidas diversas telas que atendem às necessidades do cliente. A seguir, serão apresentadas as telas desenvolvidas juntamente com instruções detalhadas de como será o comportamento de cada tela. É importante destacar que o protótipo das telas é uma representação visual do sistema em desenvolvimento e, portanto, não representa o produto final.

2.3.1 Login

Figura 1 - Tela de Login



Fonte: Elaborado pelos autores, 2023.

Na tela de login do StartRH, tanto os usuários responsáveis pela gestão de RH das empresas clientes quanto os funcionários dessas empresas podem acessar a plataforma. É por meio dessa tela que é possível entrar no sistema.

É necessário preencher os campos de "Usuário" e "Senha" com as informações de login fornecidas previamente pelo administrador do sistema. Após o preenchimento correto dos campos, o usuário deve clicar no botão "Login" para ter acesso às funcionalidades da plataforma.

2.3.2 Home 1

Figura 2 - Tela Home (Funcionário)



Fonte: Elaborado pelos autores, 2023.

Na tela Home do sistema StartRH, os usuários terão acesso a diversas funcionalidades que auxiliam na gestão de seus dados e informações pessoais. Entre as funcionalidades disponíveis, podemos destacar:

- A Receber, que apresenta a previsão de quanto o usuário irá receber no próximo pagamento;
- Histórico Holerite, que disponibiliza o histórico dos holerites dos meses anteriores;
- Histórico Salarial, que apresenta todas as alterações salariais e o mês em que ocorreram;
- Ponto Eletrônico, onde o usuário pode realizar a batida de ponto;
- Férias, que mostra a previsão de quando o usuário poderá tirar férias e quando foi a última vez que usufruiu deste benefício;
- Documentos, onde o usuário pode enviar seus documentos pessoais ou baixar algum documento fornecido pela empresa, caso seja necessário.

2.3.2 A Receber

Figura 3 - Tela de Holerite (A Receber)

MAIO	
Horas previstas:	180
Horas trabalhadas:	180
Salário bruto:	R\$ 1800,00
Desconto INSS:	R\$ 162,00
Desconto IR:	R\$ 135,00
Total descontos:	R\$ 297,00
Salário líquido:	R\$ 1503,00

Fonte: Elaborado pelos autores, 2023.

A tela de A Receber apresenta informações importantes para o usuário do sistema StartRH, permitindo que ele tenha um controle mais preciso sobre suas finanças. Nessa tela, são apresentados dados como a quantidade de horas previstas para o mês, a quantidade de horas trabalhadas, o valor do salário bruto que será recebido, o valor do desconto do INSS, o valor do desconto do Imposto de Renda, o cálculo com o valor total de descontos e o cálculo do salário líquido. Essas informações são extremamente úteis para que o usuário possa planejar suas finanças, sabendo com antecedência quanto irá receber e quais serão os descontos em seu salário.

2.3.3 Histórico Salarial

Figura 4 - Tela para consultar o Histórico Salarial



Fonte: Elaborado pelos autores, 2023.

Figura 5 - Tela para consultar o Histórico Salarial (Holerite)



Fonte: Elaborado pelos autores, 2023.

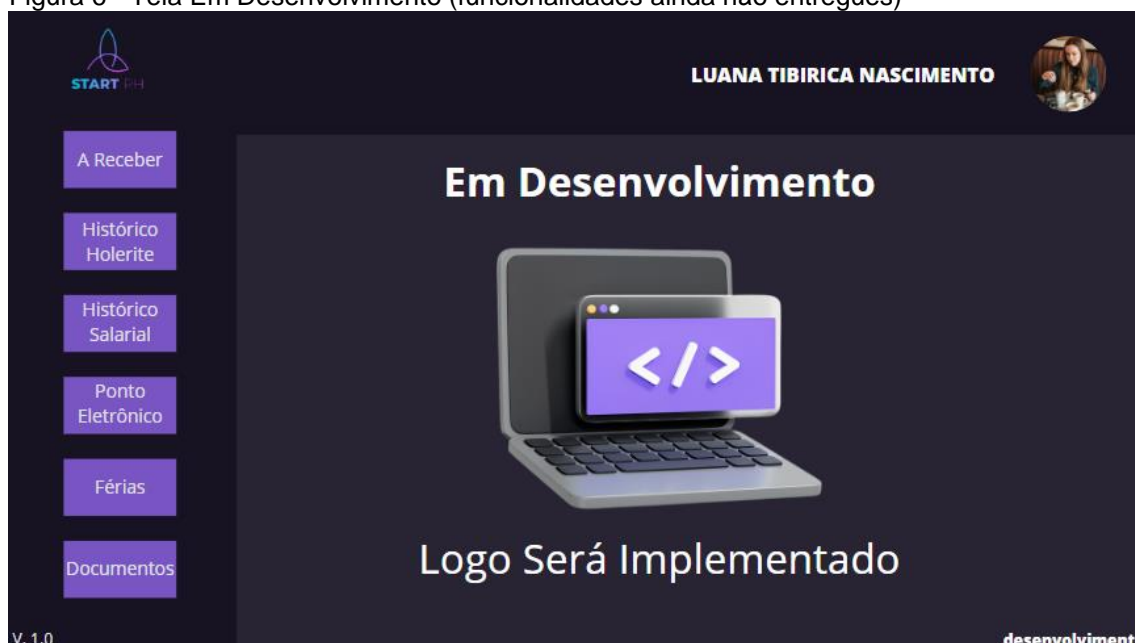
A tela de Histórico Salarial apresenta ao usuário um registro completo de todos os pagamentos salariais realizadas até então. Cada registro é identificado por meio do mês correspondente e o valor recebido pelo funcionário naquele período.

Ao clicar no mês desejado, é possível visualizar o holerite completo do referido mês, com informações detalhadas sobre os descontos e os adicionais que compõem a remuneração do funcionário, tais como horas trabalhadas, descontos, entre outros.

Essa funcionalidade é extremamente útil tanto para o funcionário, que pode conferir seus pagamentos passados, quanto para o departamento de recursos humanos da empresa, que pode consultar os dados do colaborador em questão a qualquer momento.

2.3.4 Em Desenvolvimento

Figura 6 - Tela Em Desenvolvimento (funcionalidades ainda não entregues)



Fonte: Elaborado pelos autores, 2023.

No processo de desenvolvimento contínuo do StartRH, novas funcionalidades estão sendo projetadas e implementadas seguindo o modelo em espiral. Por isso, algumas funções ainda não estão disponíveis na plataforma e estarão sendo lançadas nas próximas entregas. Ao tentar acessar essas funções, o usuário irá se deparar com uma tela informando que a funcionalidade está em desenvolvimento e em breve estará disponível. O processo de desenvolvimento em espiral permite uma abordagem flexível, evolutiva e iterativa, proporcionando melhorias contínuas na plataforma para atender às necessidades dos usuários e empresas clientes/parceiras do StartRH.

2.3.5 Home 2

Figura 7 – Tela Home (RH)



Fonte: Elaborado pelos autores, 2023.

Na tela Home do sistema os usuários terão acesso a diversas funcionalidades que auxiliam na gestão de suas empresas e funcionários. Entre as funcionalidades disponíveis, podemos destacar:

- Gestão de Funcionários, que permite o controle dos funcionários de cada empresa cliente;
- Gestão de Empresas, que possibilita o controle das empresas parceiras;
- Agendar Pagamento, que realiza o controle individual de lançamento de pagamento de cada funcionário;
- Histórico de Pagamento, que apresenta o histórico dos holerites dos meses anteriores de cada funcionário;
- Ponto Eletrônico, que realiza a gestão das batidas de ponto de cada funcionário;
- Gestão de Férias, que permite a gestão de férias de cada empresa.

2.3.6 Controle Empresas

Figura 8 - Tela Controle de Empresas



Fonte: Elaborado pelos autores, 2023.

A tela de Controle de Empresas permite que o usuário cadastre e realize alterações nas empresas clientes. Essa tela contém informações importantes, como o ID da empresa, sua razão social, e-mail e telefone de contato. Esses dados são essenciais para que a empresa possa ser identificada e contatada facilmente quando necessário.

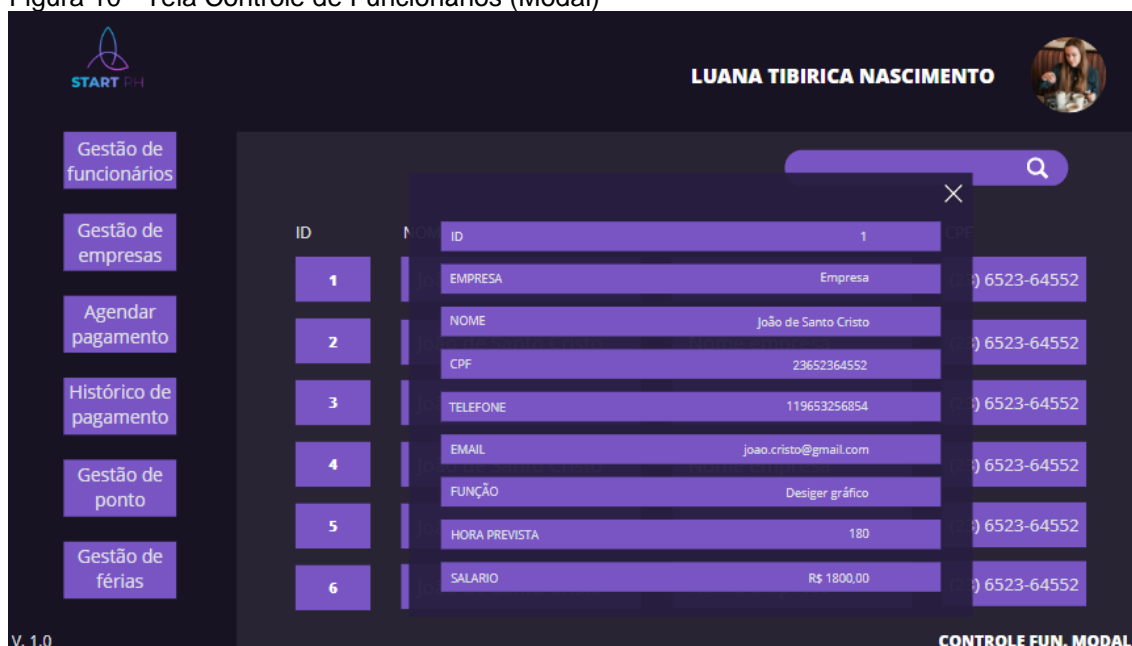
2.3.7 Controle de Funcionários

Figura 9 - Tela Controle de Funcionários



Fonte: Elaborado pelos autores, 2023.

Figura 10 - Tela Controle de Funcionários (Modal)



Fonte: Elaborado pelos autores, 2023.

A tela de controle de funcionários possibilita aos usuários cadastrar e atualizar dados referentes aos funcionários de cada empresa cliente. Essa funcionalidade é essencial para manter a base de dados atualizada e organizada, além de possibilitar o gerenciamento de informações relevantes sobre os funcionários.

Na tela, é possível visualizar informações importantes como o ID, nome do funcionário, razão social e CPF. Ao clicar em um funcionário específico, são apresentadas mais informações, como telefone, e-mail de contato, função registrada em carteira, horas mensais previstas e o valor do salário bruto.

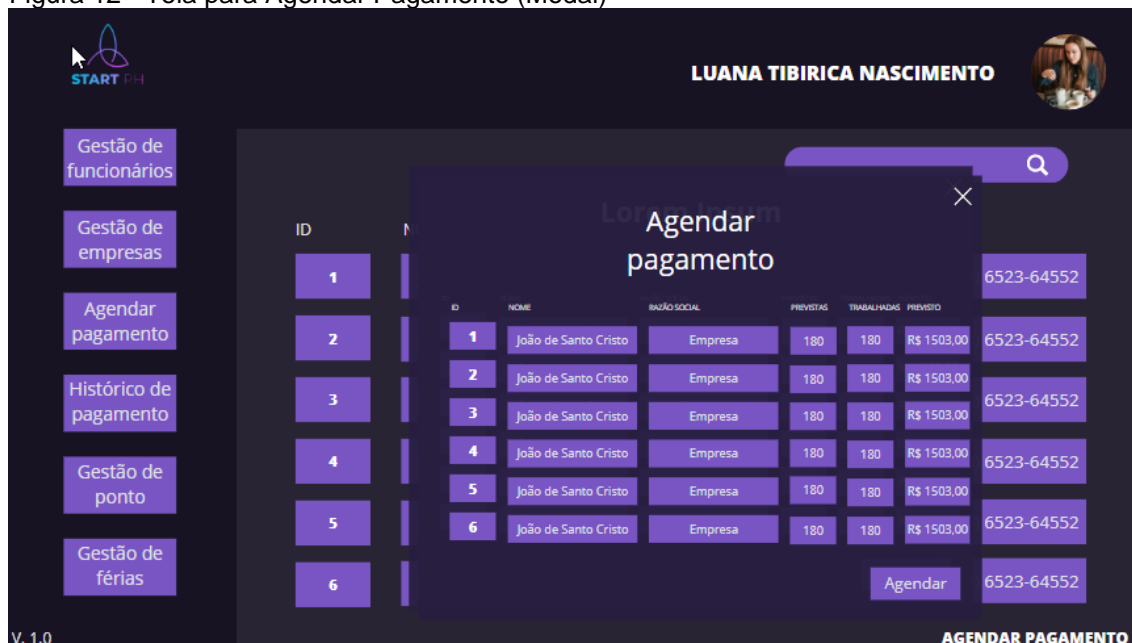
2.3.8 Agendar pagamento

Figura 11 - Tela para Agendar Pagamento



Fonte: Elaborado pelos autores, 2023.

Figura 12 - Tela para Agendar Pagamento (Modal)



Fonte: Elaborado pelos autores, 2023.

Na tela de agendar pagamento é possível visualizar uma lista com todos os funcionários e suas respectivas empresas. Ao clicar em um funcionário, é possível acessar um campo que exibe as horas trabalhadas previstas e as horas trabalhadas naquele mês. Além disso, apresenta o valor do salário líquido que o funcionário irá receber naquele mês.

Após preencher todos os campos necessários, basta clicar no botão "Agendar" para confirmar o agendamento do pagamento. Essa funcionalidade ajuda na organização e no controle de pagamentos de cada funcionário, tornando o processo mais eficiente e preciso.

2.3.9 Histórico

Figura 13 - Tela Histórico de Pagamentos



Fonte: Elaborado pelos autores, 2023.

Figura 14 - Tela Histórico de pagamentos (Modal)



Fonte: Elaborado pelos autores, 2023.

A tela de histórico permite a visualização dos dados históricos dos funcionários de cada empresa cliente. Nessa tela, é listado todos os meses e, ao clicar em um determinado mês, serão apresentados todos os funcionários correspondentes àquela data, sua respectiva empresa, previsão de horas, horas trabalhadas e o valor do salário líquido recebido. Além disso, permite a realização de análises comparativas entre os meses e o planejamento estratégico para os próximos períodos.

As instruções detalhadas de cada tela têm como objetivo fornecer um guia claro de como o usuário deve interagir com o sistema, além de orientar o desenvolvedor no processo de implementação das funcionalidades e comportamentos de cada tela.

Ao apresentar o protótipo das telas juntamente com as instruções detalhadas, busca-se garantir que o sistema será desenvolvido de forma clara e precisa, atendendo às necessidades do cliente e proporcionando uma boa experiência de uso para o usuário final.

2.4 UML

Um acrônimo para a expressão “Unified Modeling Language” ou simplesmente “Linguagem de Modelagem Unificada”, o UML é uma linguagem de notação para uso em projetos de sistema.

Os diagramas da UML fornecem um “desenho” do sistema que se pretende desenvolver, centralizando nos diagramas um determinado conceito de fácil entendimento aos envolvidos no projeto.

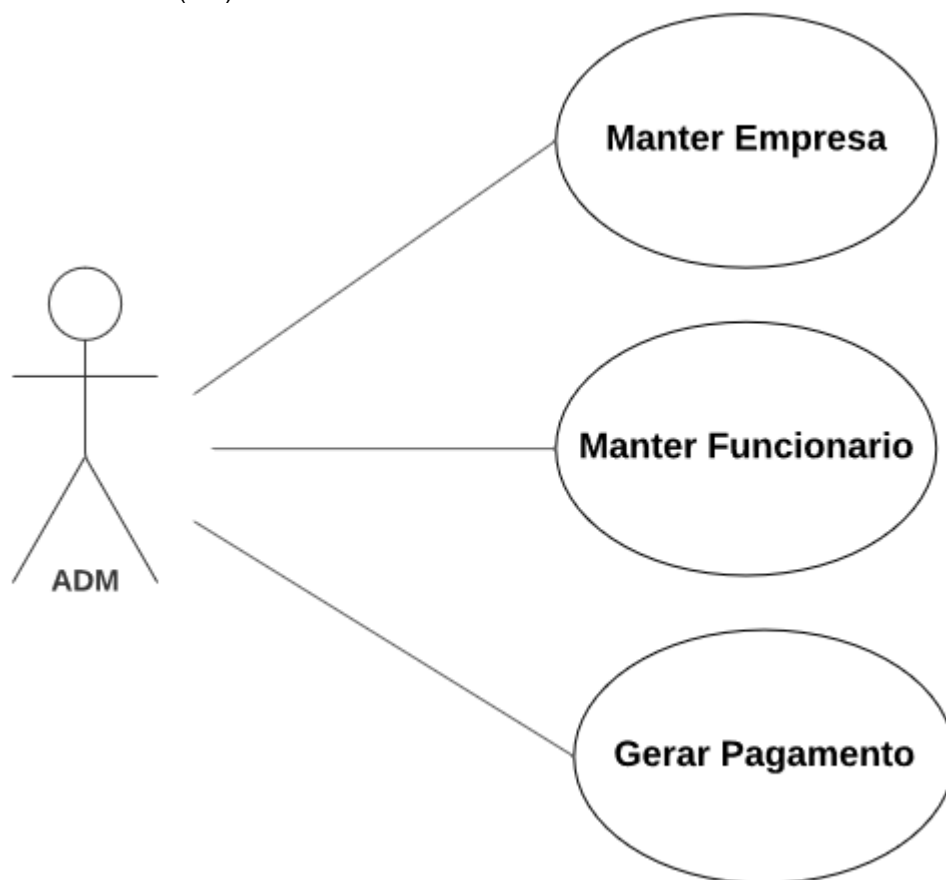
Composto por diferentes tipos de diagramas, permite que os desenvolvedores possam ter uma visão geral do seu trabalho em diagramas padronizados. Estão divididos em dois grandes grupos: Os diagramas estruturais e os diagramas comportamentais.

2.4.1 Casos de Uso

O diagrama de caso de uso resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema.

Caso de Uso: Administrador Mantém Empresa, Mantém Funcionário e Gera pagamento.

Figura 15 - Caso de Uso (RH)



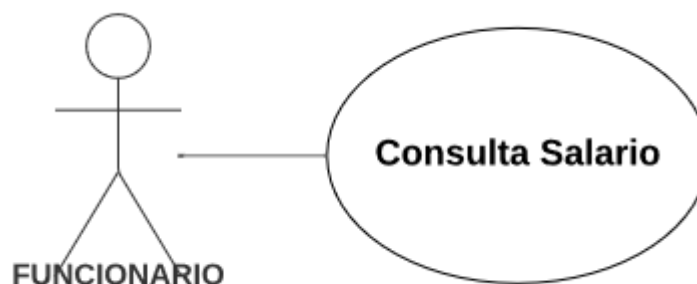
Fonte: Elaborado pelos autores, 2023.

Ator Principal: Administrador.

Descrição: O administrador realiza operações de manutenção de informações sobre empresas no sistema; O administrador realiza operações de manutenção de informações sobre funcionários no sistema; O administrador gera os pagamentos dos funcionários no sistema.

Caso de Uso: Funcionário Consulta Salário.

Figura 16 - Caso de Uso (Funcionário)



Fonte: Elaborado pelos autores, 2023.

Ator Principal: Funcionário.

Descrição: O funcionário consulta o seu salário no sistema.

2.4.2 Diagrama de Atividade

Um diagrama de atividade é essencialmente um fluxograma que mostra as atividades executadas por um sistema.

Diagrama de Atividade: Gerenciamento de Empresa.

Figura 17 - Diagrama de Atividade (Gerenciamento de Empresas)



Fonte: Elaborado pelos autores, 2023.

Descrição: Nesse diagrama, o ator principal é o "ADM" (Administrador), que executa quatro atividades sequenciais. Primeiro, o ADM seleciona a opção "gerenciamento de empresas". Em seguida, ocorre a atividade "Consulta Empresas", onde o ADM consulta informações sobre as empresas existentes. Após a consulta, o ADM seleciona uma empresa específica. Por fim, é realizada a atividade "Realiza

Atualização da Empresa", onde o ADM atualiza (inclui, altera e/ou exclui) os dados da empresa selecionada.

Diagrama de Atividade: Gerenciamento de Funcionário.

Figura 18 - Diagrama de Atividade (Gerenciamento de Funcionários)

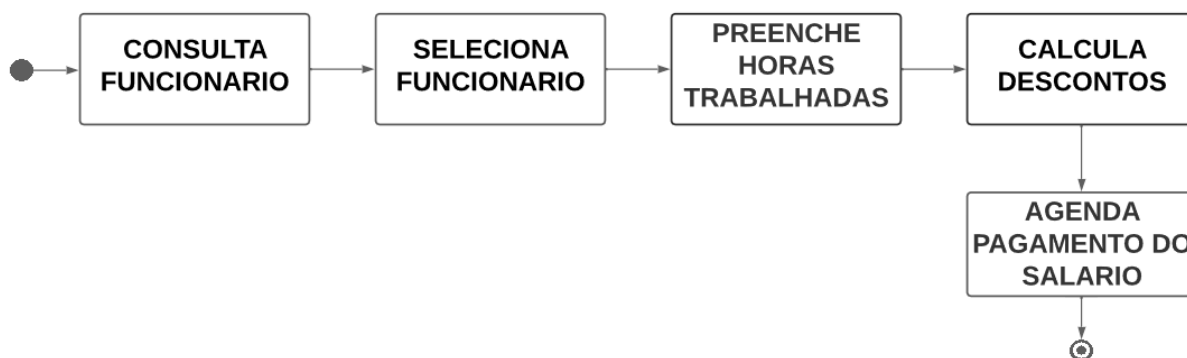


Fonte: Elaborado pelos autores, 2023.

Descrição: Nesse diagrama, o ator principal é o "ADM" (Administrador), que executa quatro atividades sequenciais. Primeiro, o ADM seleciona a opção "empresa". Em seguida, ocorre a atividade "Consulta Funcionário", onde o ADM consulta informações sobre os funcionários da empresa selecionada. Após a consulta, o ADM seleciona um funcionário específico. Por fim, é realizada a atividade "Realiza Atualização do Funcionário", onde o ADM atualiza os dados do funcionário selecionado.

Diagrama de Atividade: Agendar Pagamento

Figura 19 - Diagrama de Atividade: Agendar Pagamento



Fonte: Elaborado pelos autores, 2023.

Descrição: Nesse diagrama, o ator principal é o "ADM" (Administrador), que executa cinco atividades sequenciais. Primeiro, o ADM realiza a consulta dos funcionários. Em seguida, ocorre a atividade "Seleciona Funcionário", onde o ADM escolhe um funcionário específico. Após a seleção, o ADM preenche as horas trabalhadas pelo funcionário. Em seguida, ocorre a atividade "Calcula Descontos", onde são realizados os cálculos necessários para determinar os descontos no salário. Por fim, o ADM agenda o pagamento do salário do funcionário.

Diagrama de Atividade: Consultar Holerite.

Figura 20 - Diagrama de Atividade: Consultar Holerite



Fonte: Elaborado pelos autores, 2023.

Descrição: Nesse diagrama, temos um único participante, o "Funcionário", que realiza duas atividades sequenciais. A primeira atividade é "seleciona holerite", onde o funcionário seleciona o holerite específico que deseja consultar. Em seguida, ocorre a atividade "Consulta Holerite", onde o funcionário realiza a consulta do holerite selecionado.

Diagrama de Atividade: Consulta Histórico Salarial.

Figura 21 - Diagrama de Atividade: Consulta de Histórico Salarial



Fonte: Elaborado pelos autores, 2023.

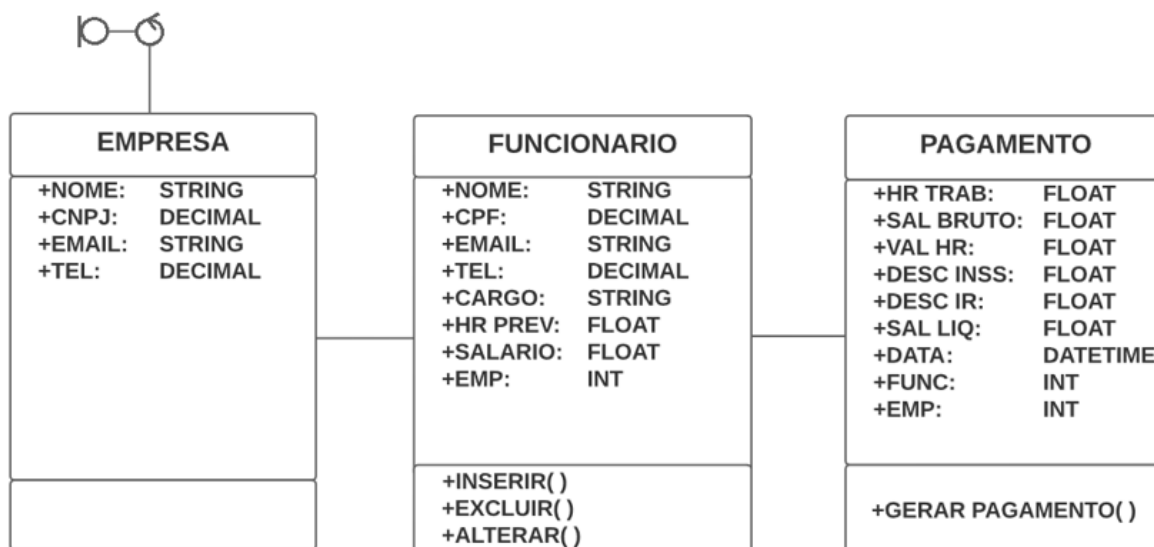
Descrição: Nesse diagrama de atividade, o ator principal pode ser o "ADM" (Administrador) ou o "Funcionário" realizam três atividades sequenciais. O processo começa com a atividade "Seleciona Histórico Salarial", onde o ator escolhe essa opção no sistema. Em seguida, o ator seleciona o mês desejado na atividade "Seleciona Mês". Após a seleção do mês, o ator realiza a consulta do holerite na atividade "Consulta Holerite". A consulta retorna as informações detalhadas do holerite do salário para o mês selecionado. Esse processo encerra após a consulta do holerite. É importante ressaltar que o fluxo desse diagrama pode ser seguido tanto pelo ADM quanto pelo Funcionário, dependendo de quem estiver interagindo com o sistema.

2.4.3 Diagrama de Classe

Diagramas de classes estão entre os tipos mais úteis de diagramas UML pois mapeiam de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos, operações e relações entre objetos.

Diagrama de Classe

Figura 22 - Diagrama de Classe



Fonte: Elaborado pelos autores, 2023.

Descrição: Nesse diagrama, temos as três classes principais: "Empresa", "Funcionário" e "Pagamento".

- Classe Empresa: A classe "Empresa" possui os atributos nome, CNPJ, email e telefone, além dos métodos de inserir, excluir e alterar para manipular os dados da empresa;
- Classe Funcionario: A classe "Funcionário" contém os atributos nome, CPF, email, telefone, cargo, horas previstas, salário e uma referência à empresa em que trabalha. Também possui os métodos de inserir, excluir e alterar para manipular os dados do funcionário;
- Classe Pagamento: A classe "Pagamento" possui os atributos relacionados a informações do pagamento, como horas trabalhadas, salário bruto, valor hora, descontos (INSS e IR), salário líquido, data do pagamento e referências ao funcionário e empresa relacionados. Além dos métodos de inserir, excluir e alterar, a classe "Pagamento" possui o método "gerarPagamento()" para executar o processo de geração do pagamento com base nas informações

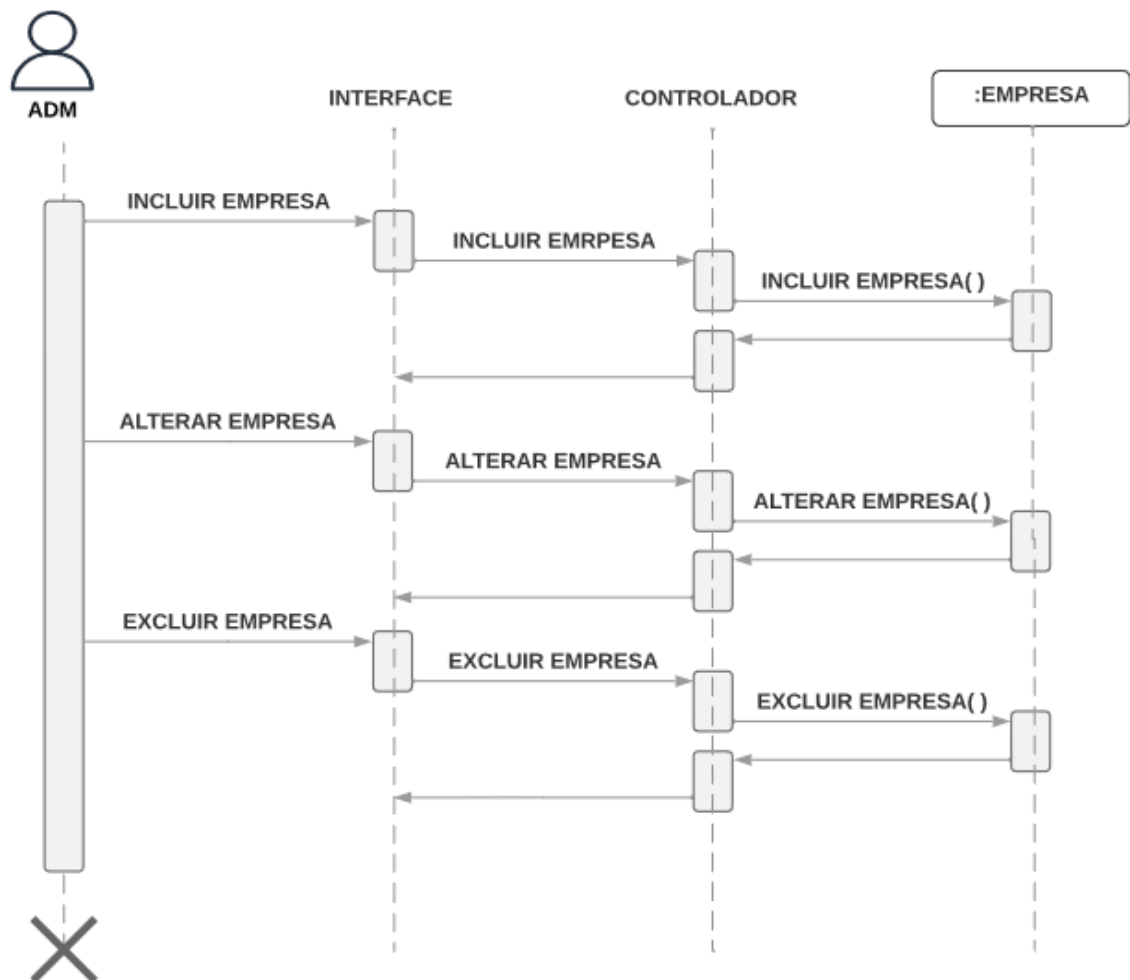
fornecidas.

2.4.4 Diagrama de Sequência

O diagrama de sequência é uma solução dinâmica de modelagem em UML bastante usada porque incide especificamente sobre linhas da vida, ou os processos e objetos que vivem simultaneamente, e as mensagens trocadas entre eles para desempenhar uma função antes do término da linha da vida.

Diagrama de Sequência: Empresa

Figura 23 - Diagrama de Sequência: Empresa



Fonte: Elaborado pelos autores, 2023.

Descrição: Nesse diagrama, temos o ator "ADM" interagindo com a classe "Empresa". A sequência de interações é a seguinte:

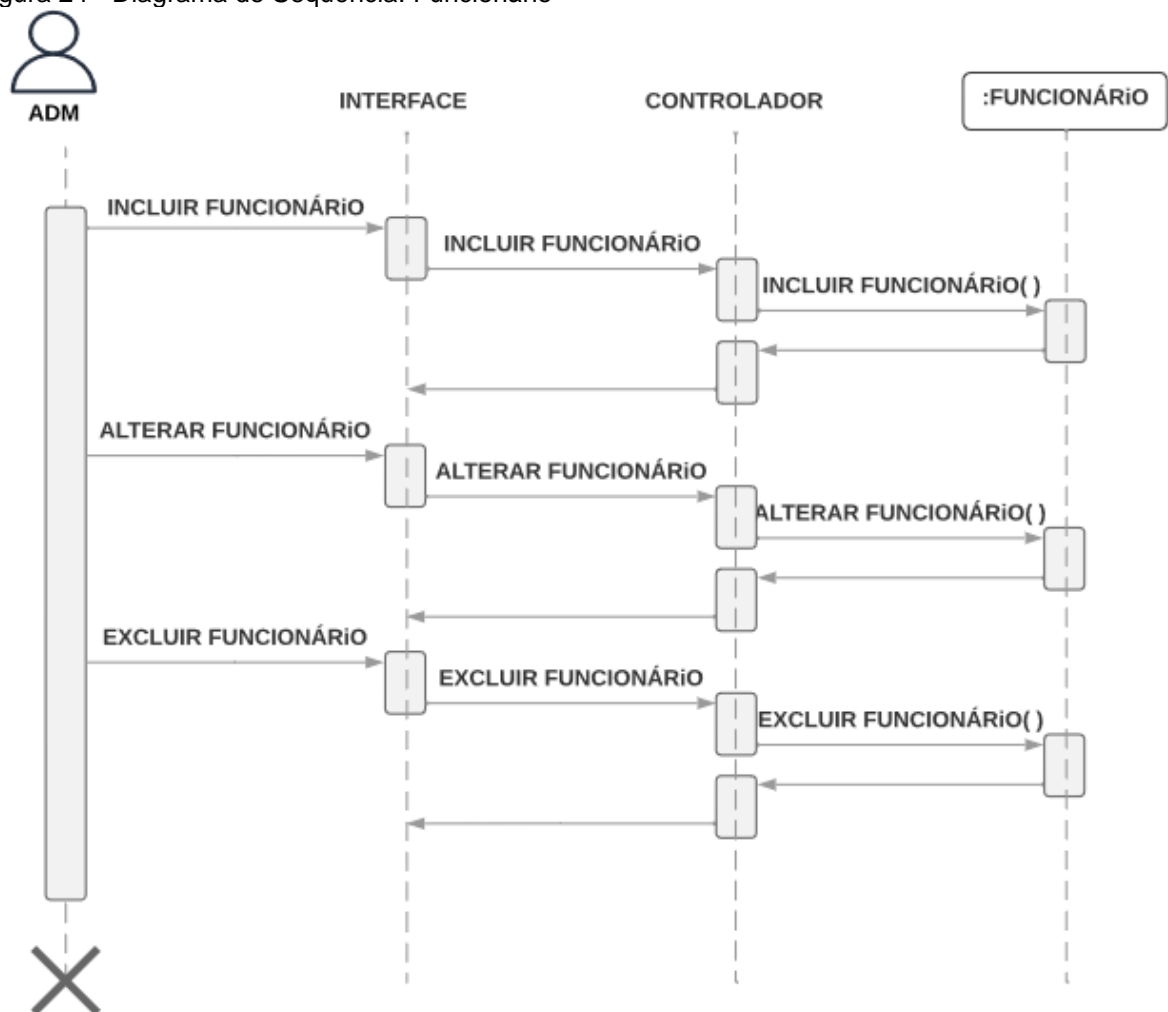
1. O ator "ADM" invoca o método "incluir()" da classe "Empresa" para incluir uma nova empresa.

2. A classe "Empresa" processa a solicitação e retorna uma resposta ao ator "ADM".
3. O ator "ADM" invoca o método "alterar()" da classe "Empresa" para alterar uma empresa existente.
4. A classe "Empresa" processa a solicitação e retorna uma resposta ao ator "ADM".
5. O ator "ADM" invoca o método "excluir()" da classe "Empresa" para excluir uma empresa existente.
6. A classe "Empresa" processa a solicitação e retorna uma resposta ao ator "ADM".

Esse diagrama de sequência ilustra a interação entre o ator "ADM" e a classe "Empresa" para realizar as operações de inclusão, alteração e exclusão de empresas.

Diagrama de Sequência: Funcionário

Figura 24 - Diagrama de Sequência: Funcionário



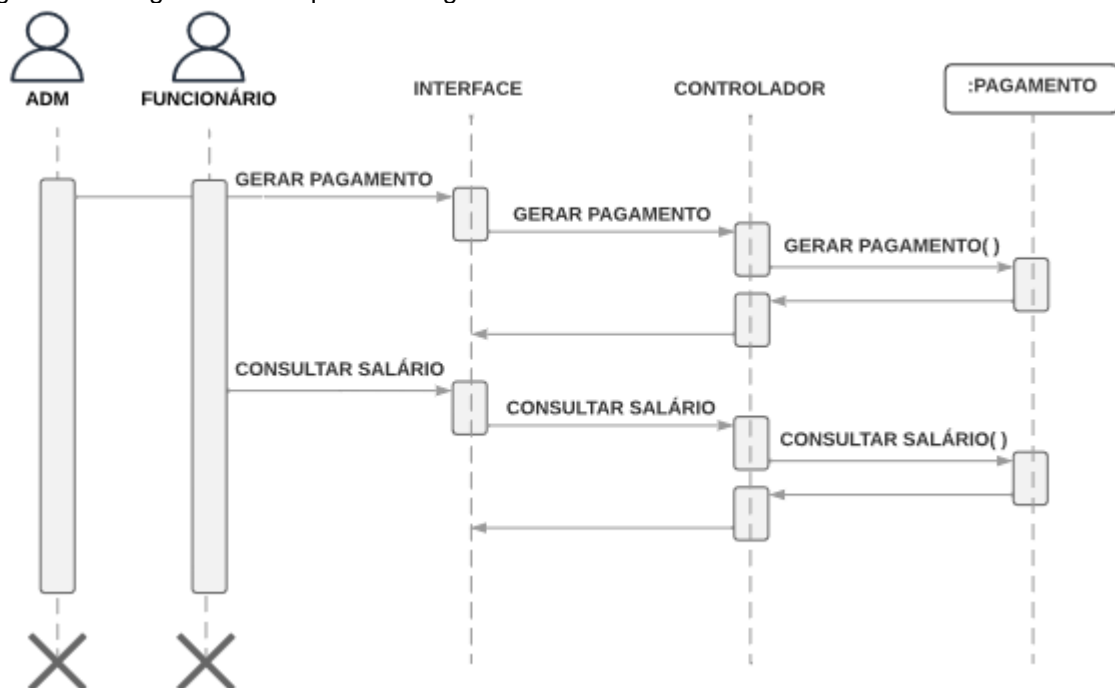
Descrição: Neste diagrama, temos o ator "ADM" interagindo com a classe "Funcionário". A sequência de interações é a seguinte:

1. O ator "ADM" invoca o método "incluir()" da classe "Funcionário" para incluir um novo funcionário.
2. A classe "Funcionário" processa a solicitação e retorna uma resposta ao ator "ADM".
3. O ator "ADM" invoca o método "alterar()" da classe "Funcionário" para alterar um funcionário existente.
4. A classe "Funcionário" processa a solicitação e retorna uma resposta ao ator "ADM".
5. O ator "ADM" invoca o método "excluir()" da classe "Funcionário" para excluir um funcionário existente.
6. A classe "Funcionário" processa a solicitação e retorna uma resposta ao ator "ADM".

Este diagrama de sequência ilustra a interação entre o ator "ADM" e a classe "Funcionário" para realizar as operações de inclusão, alteração e exclusão de funcionários.

Diagrama de Sequência: Pagamento

Figura 25 - Diagrama de Sequência: Pagamento



Fonte: Elaborado pelos autores, 2023.

Descrição: Neste diagrama, temos o ator "ADM" interagindo com a classe "Pagamento" para gerar o pagamento, e o ator "Funcionario" consultando o salário através da mesma classe. A sequência de interações é a seguinte:

1. O ator "ADM" invoca o método "geraPagamento()" da classe "Pagamento" para gerar o pagamento.
2. A classe "Pagamento" processa a solicitação e retorna uma resposta ao ator "ADM".
3. O ator "Funcionario" invoca o método "consultaSalario()" da classe "Pagamento" para consultar o salário.
4. A classe "Pagamento" processa a solicitação e retorna o valor do salário ao ator "Funcionario".

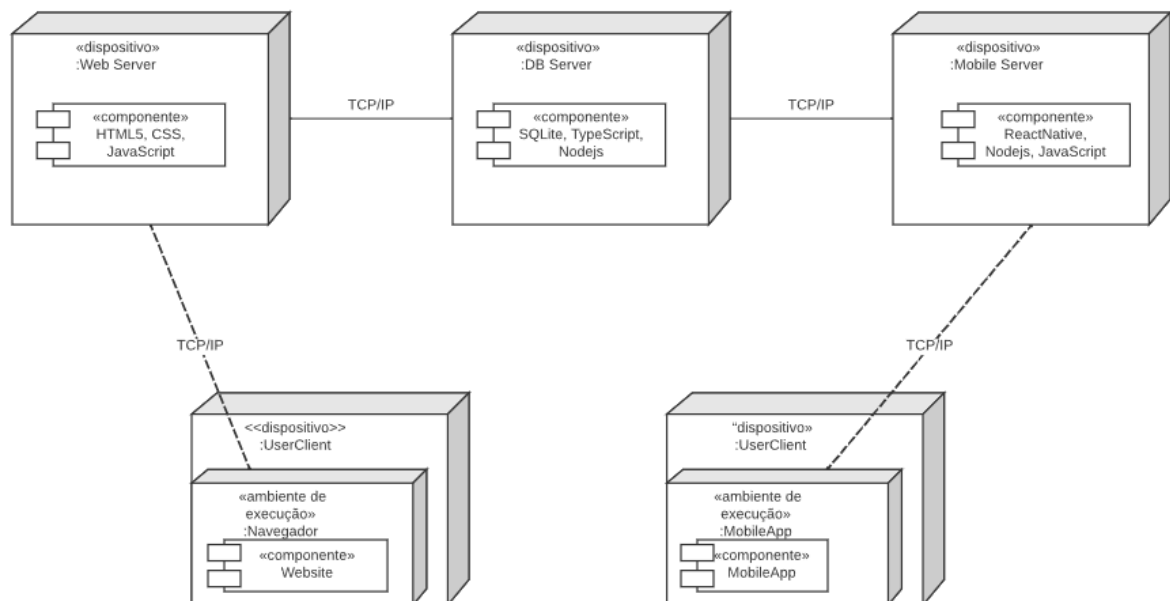
Este diagrama de sequência ilustra a interação entre o ator "ADM", o ator "Funcionario" e a classe "Pagamento" para realizar as operações de geração de pagamento e consulta de salário.

2.4.5 Diagrama de Implantação

O diagrama de implementação descreve a implementação física de informações geradas pelo programa de software em componentes de hardware.

Diagrama de Implantação.

Figura 26 - Diagrama de Implantação



Fonte: Elaborado pelos autores, 2023.

Descrição: O web server, responsável por hospedar o website e fornecer os arquivos HTML, CSS e JavaScript para o navegador; o DB server, que contém o banco de dados SQLite e utiliza TypeScript e NodeJS para gerenciar as operações de banco de dados; e o mobile server, que hospeda o aplicativo mobile desenvolvido com React Native, NodeJS e JavaScript.

Os clientes são o navegador e o aplicativo mobile, que se comunicam com os servidores via protocolo TCP/IP, como indicado na descrição. O navegador acessa o web server para obter as páginas HTML, CSS e JavaScript e renderizá-las. O aplicativo mobile se conecta ao mobile server para obter as informações e recursos necessários para funcionar.

2.5 Banco de Dados

Um banco de dados é uma coleção organizada de informações (ou dados) estruturadas, normalmente armazenadas eletronicamente em um sistema de computador.

Prisma é uma biblioteca do lado do servidor que ajuda os desenvolvedores a ler e gravar dados no banco de dados de maneira intuitiva, eficiente e segura.

Figura 27 - Como o prisma.io funciona?



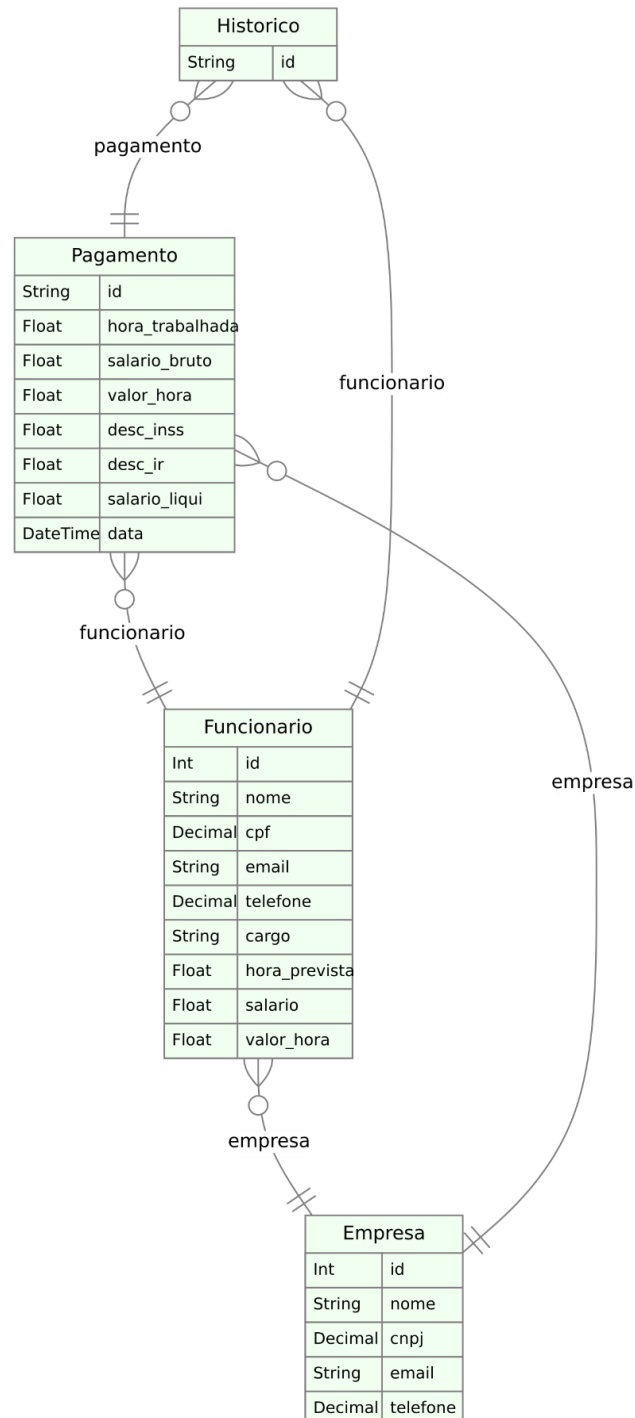
Fonte: Elaborado por prisma.io, 2023; <https://www.prisma.io/>.

O Prisma.io é uma ferramenta ORM (*Object Relational Mapper*) de gerenciamento de banco de dados moderna e eficiente que permite a realização de consultas de maneira rápida e eficaz. Ao utilizá-lo para gerir e realizar as queries no banco, podemos garantir uma baixa latência para a comunicação com o Banco de Dados. Isso é possível porque o Prisma é otimizado para trabalhar com o banco de dados e oferece uma performance similar a realizar uma query convencional do SQL.

2.5.1 Entidade Relacionamento

Um diagrama entidade relacionamento (ER) é um tipo de fluxograma que ilustra como “entidades se relacionam entre si dentro de um sistema. Diagramas ER são mais utilizados para projetar ou depurar bancos de dados relacionais nas áreas de engenharia de software.

Figura 28 - Modelo Entidade Relacionamento



Fonte: Elaborado pelos autores, 2023.

Na imagem do diagrama ER apresentado, é possível identificar a presença de quatro bancos distintos: Histórico, Pagamento, Funcionário e Empresa. Cada banco representa um conjunto de informações que são armazenadas no sistema, sendo que os relacionamentos entre eles são indicados pelas linhas que conectam as entidades. A partir do diagrama ER, é possível entender como as informações se relacionam e como elas são armazenadas no sistema.

2.5.2 Script de Criação do Banco

A criação do banco de dados e suas tabelas é, muitas vezes, um dos primeiros passos da parte prática do desenvolvimento de um novo projeto.

Com o auxílio do Prisma.io, foi possível criar o banco de dados necessário para o desenvolvimento do sistema utilizando o seguinte script:

```
1  ```Prisma
2  generator client {
3    provider = "prisma-client-js"
4  }
5
6  datasource db {
7    provider = "sqlite"
8    url       = env("DATABASE_URL")
9  }
10
11 model Empresa {
12   id      Int      @id @default(autoincrement())
13   nome    String
14   cnpj    Decimal @unique
15   email   String
16   telefone Decimal
17
18   funcionario Funcionario[]
19   pagamento   Pagamento[]
20
21   @@map("empresas")
22 }
23
24 model Funcionario {
25   id      Int      @id @default(autoincrement())
26   nome    String
27   cpf     Decimal  @unique
28   email   String?
29   telefone Decimal?
30   empresa_id Int
31   cargo   String
32   hora_prevista Float
33   salario  Float
34   valor_hora Float
35
36   empresa Empresa @relation(fields: [empresa_id], references: [id])
```

```
37
38 historico Historico[]
39 pagamento Pagamento[]
40
41 @@map("funcionarios")
42 }
43
44 model Pagamento {
45   id          String    @id @default(uuid())
46   funcionario_id Int
47   empresa_id   Int
48   hora_trabalhada Float
49   salario_bruto Float
50   valor_hora    Float
51   desc_inss     Float
52   desc_ir       Float
53   salario_liqui  Float
54   data          DateTime @default(now())
55
56   historico Historico[]
57
58   funcionario Funcionario @relation(fields: [funcionario_id], references: [id])
59   empresa      Empresa    @relation(fields: [empresa_id], references: [id])
60
61   @@unique([funcionario_id, data])
62   @@map("pagamentos")
63 }
64
65 model Historico {
66   id          String    @id @default(uuid())
67   funcionario_id Int
68   pagamento_id String
69
70   pagamento Pagamento @relation(fields: [pagamento_id], references: [id])
71   funcionario Funcionario @relation(fields: [funcionario_id], references: [id])
72
73   @@map("historico")
74 }
75 ````
```

2.6 Planilha de Testes

Teste de software é uma das atividades do processo de desenvolvimento de sistema de software que visa executar um programa de modo sistemático com o objetivo de encontrar falhas. Perceba que isto requer verificação e validação de software. Nesse sentido, definir quando as atividades de verificação e validação iniciam e terminam, como os atributos de qualidade serão avaliados e como os releases do software serão controlados, são questões que devem ser acompanhadas ao longo do processo de software.

Tabela 4 - Planilha de Testes

PLANILHA DE TESTES					
STEP ID	DESCRIÇÃO DO PASSO	RESULTADOS ESPERADOS	RESULTADOS REAIS	PASSE / FALHA	NOTAS ADICIONAIS
1	Realizar login	Preencher os campos corretos de login e senha	Login realizado com sucesso	Sucesso	Avança para a página home
1	Realizar login	Não preencher os campos de login e/ou senha	Login não realizado	Falha	Alerta na tela ("Preencha os campos")
1	Realizar login	Preencher os campos com tipo dados incorretos	Login não realizado	Falha	Alerta na tela ("Login incorreto")
2	Cadastrar empresa	Preencher os campos solicitados corretamente	Empresa cadastrada no sistema	Sucesso	Apresenta a nova empresa na tela
2	Cadastrar empresa	Não preencher os campos	Empresa não cadastrada no sistema	Falha	Alerta na tela ("Preencha todos os campos")
2	Cadastrar empresa	Preencher os campos com tipo dados incorretos	Empresa não cadastrada no sistema	Falha	Alerta na tela ("Tipo de dado inválido")
3	Cadastrar funcionário	Preencher os campos solicitados corretamente	Funcionário cadastrada no sistema	Sucesso	Apresenta o novo funcionário na tela

3	Cadastrar funcionário	Não preencher os campos	Funcionário não cadastrado no sistema	Falha	Alerta na tela ("Preencha todos os campos")
3	Cadastrar funcionário	Preencher os campos com tipo dados incorretos	Funcionário não cadastrado no sistema	Falha	Alerta na tela ("Tipo de dado inválido")
4	Agendar pagamento	Preencher o número de horas trabalhadas	Pagamento agendado	Sucesso	Gera um agendamento de pagamento do salário
4	Agendar pagamento	Não preencher os campos	Não é gerado o pagamento	Falha	Alerta na tela ("Preencha todos os campos")
4	Agendar pagamento	Preencher o campo com tipo dado incorreto	Não é gerado o pagamento	Falha	Alerta na tela ("Tipo de dado inválido")
5	Consultar histórico de pagamento	Pesquisar por um mês com pagamento válido	Apresenta o holerite	Sucesso	Apresenta o holerite do mês solicitado
5	Consultar histórico de pagamento	Pesquisar por um mês onde não houve pagamento	Não apresenta holerite	Falha	Alerta na tela ("Mês pesquisado não possui pagamento")

Fonte: Elaborado pelos autores, 2023.

2.7 Como instalar

Antes de iniciar a instalação, é necessário verificar se as seguintes ferramentas estão instaladas na máquina: Git, Node.js, e Expo para dispositivos móveis (Android ou iOS). Recomenda-se também o uso de um editor de código-fonte como o VSCode

para trabalhar no código. Certifique-se de que todas as ferramentas estão corretamente instaladas antes de prosseguir com a instalação.

Antes de executar as aplicações frontend e mobile, é necessário verificar o endereço IP da máquina onde o backend será executado (presente na pasta "server") e configurá-lo no arquivo de configuração do Axios, localizado na pasta "lib" dos projetos correspondentes.

2.7.1 Rodando o back-end (Servidor)

```
1 `` `Bash
2 # Clone este repositório.
3 $ git clone git@github.com:marlonakio/PIM_IV_ADS_UNIP.git
4
5 # Acesse a pasta do projeto no terminal.
6 $ cd PIM_IV_ADS_UNIP
7
8 # Vá para a pasta server.
9 $ cd server
10
11 # Instale as dependências.
12 $ npm install
13
14 # Execute a aplicação em modo de desenvolvimento.
15 $ npm run dev
16
17
18 # O servidor iniciará na porta 3333 - acesse <http://localhost:3333>
19 `` `
```

2.7.2 Rodando o front-end (Web)

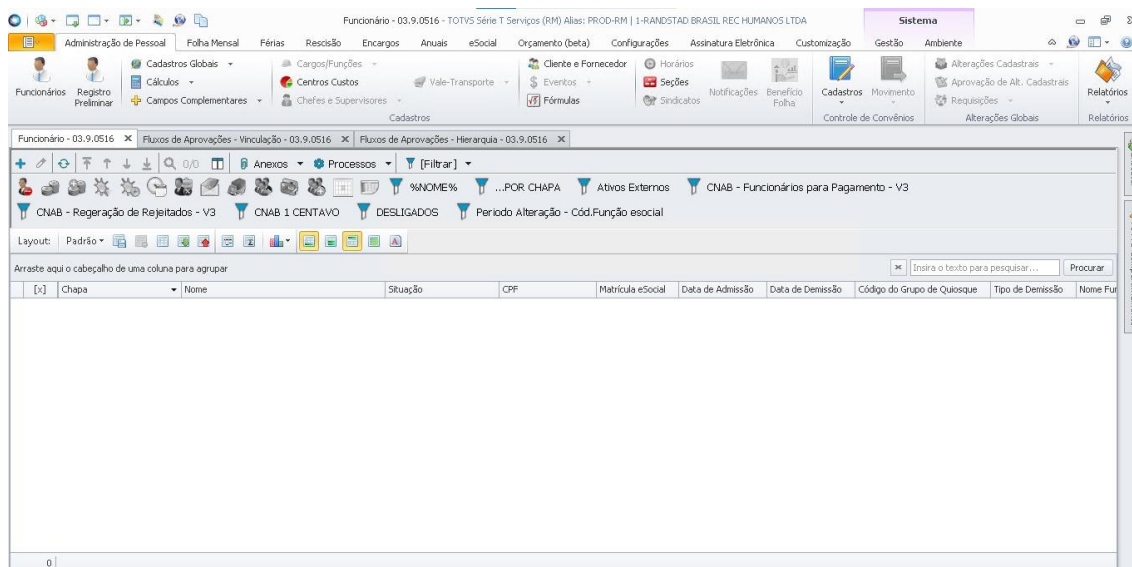
```
1 `` `Bash
2 # O servidor web deve estar rodando na porta 80 - acesse
3 <http://localhost/PIM/WEB/>
4 `` `
```

2.8 Comparativo de Mercado

Método Comparativo é aquele que define o valor através da comparação com dados de mercado assemelhados quanto às características, identificando o valor de mercado do bem por meio de tratamento técnico dos atributos e características dos elementos comparáveis, constituintes da amostra pesquisada.

2.8.1 TOTVS RM

Figura 29 - TOTVS RM



Fonte: TOTVS RM, 2023; <https://www.totvs.com/>.

TOTVS RH linha RM é um sistema completo de gestão de Recursos Humanos desenvolvido pela TOTVS, que inclui processos de recrutamento e seleção, cargos e salários, controle de ponto eletrônico, medicina e segurança do trabalho, folha de pagamento, avaliação de desempenho, treinamento e desenvolvimento.

2.8.2 G.Infor

Figura 30 – G.Infor



Fonte: G.Infor, 2023; <https://www.gi.com.br/>.

O G.Infor é um sistema de RH que oferece soluções para gerenciamento de diversos processos, incluindo cargos e salários, faturamento e folha de pagamento, entre outras funcionalidades.

2.8.3 Bitti

Figura 31 - Bitti



Fonte: Bitti, 2023; <http://www.bitti.com.br/solucoes>.

O sistema Bitti é uma solução de RH que oferece diversos módulos para gerenciar processos, incluindo folha de pagamento com faturamento acoplado. Além disso, a plataforma integra-se com o financeiro, permitindo o controle de contas a pagar, contas a receber e bancos, bem como a contabilidade.

Ao comparar estes três sistemas já existentes no mercado, podemos notar que cada um deles possui suas próprias funcionalidades e características únicas. Entretanto, a plataforma do StartRH se destaca por algumas razões específicas.

Uma das principais vantagens da plataforma do StartRH é o controle de RH personalizado para cada empresa cliente, levando em consideração suas necessidades específicas. Isso permite que a plataforma seja adaptada para atender às demandas de cada cliente, garantindo que suas necessidades sejam atendidas de maneira eficiente e eficaz.

Além disso, a plataforma do StartRH oferece a automatização de processos como batidas de ponto, processos seletivos e pagamento de salários e décimo terceiro, o que aumenta a eficiência e reduz o tempo gasto em tarefas manuais e

repetitivas. Isso também resulta em uma redução de erros e uma melhoria na precisão dos dados.

Outra vantagem da plataforma do StartRH é o acesso via Web e Mobile, o que permite que os funcionários acessem a plataforma de qualquer lugar e a qualquer hora, trazendo maior flexibilidade e conveniência. Isso aumenta a satisfação dos funcionários e melhorar a comunicação entre a empresa e seus colaboradores.

A plataforma do StartRH também permite o controle de férias previstas, o que pode ajudar na gestão e planejamento do RH, garantindo que as férias sejam distribuídas de maneira justa e adequada.

Por fim, a integração de funcionalidades da plataforma do StartRH com outras funcionalidades como batidas de ponto e controle de férias pode torná-lo mais eficiente do que outros sistemas que não oferecem essa integração. Isso permite uma melhor comunicação e fluxo de dados entre as diversas funcionalidades, aumentando a eficiência geral do sistema.

3 CONCLUSÃO

Diante do exposto, o desenvolvimento do software para automatização de processos de RH se mostrou um desafio complexo e exigiu um amplo levantamento de requisitos, negociações e decisões estratégicas entre a Dexcode e a empresa cliente StartRH. Com a escolha do modelo de entrega em espiral, é possível desenvolver o sistema de forma gradual, garantindo maior eficiência e qualidade na entrega dos módulos.

O presente trabalho apresentou uma visão detalhada do processo de desenvolvimento da primeira etapa do projeto, incluindo a descrição dos requisitos, protótipos das telas, casos de uso, diagramas de classe e o desenvolvimento do DB com diagrama ER. Através da elaboração desse material, foi possível estabelecer um plano de ação claro e objetivo, que permitiu aos programadores trabalharem de forma mais organizada e eficiente.

Por fim, acreditamos que o software desenvolvido pela Dexcode irá proporcionar uma maior eficiência aos processos de RH da empresa StartRH, reduzindo a necessidade de cadastros em múltiplas plataformas e aumentando o controle sobre a folha de pagamento, ponto eletrônico, férias e décimo terceiro. O trabalho também demonstrou a importância do modelo de entrega em espiral no desenvolvimento de projetos de software, permitindo a construção de um sistema de forma mais estruturada e eficiente.

REFERÊNCIAS BIBLIOGRÁFICAS

CUNHA, Fernando. **Requisitos funcionais e não funcionais: o que são?** [2022]. Disponível em: <https://www.mestresdaweb.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>. Acesso em: 12 mar. 2023.

PRISMA DATA, INC. (Berlin, Berlin). **Next-generation Node.js and TypeScript ORM.** [20--]. Disponível em: <https://www.prisma.io/>. Acesso em: 18 mar. 2023.

ORACLE (Austin, Texas). **O que é um Banco de Dados?** [20--]. Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 18 mar. 2023.

RONIERE. **Criando banco de dados e tabelas no SQL Server.** [2012]. Disponível em: <https://www.devmedia.com.br/criando-banco-de-dados-e-tabelas-no-sql-server/24299>. Acesso em: 18 mar. 2023.

JULIANO, Rui. **Método Comparativo Direto de Dados de Mercado.** [2021]. Disponível em: <https://cursoavalia.com/metodo-comparativo/metodo-comparativo-direto-de-dados-de-mercado/>. Acesso em: 08 abr. 2023.

SOARES, João Paulo. **Como funciona a prototipação no desenvolvimento de software?** [2020]. Disponível em: <https://www.treinaweb.com.br/blog/como-funciona-a-prototipacao-no-desenvolvimento-de-software>. Acesso em: 29 abr. 2023.

ANTONIO. **Plano de Teste - Um Mapa Essencial para Teste de Software.** [2009]. Disponível em: <https://www.devmedia.com.br/plano-de-teste-um-mapa-essencial-para-teste-de-software/13824>. Acesso em: 07 maio 2023.

ANDRADE, Ana Paula de. **O que é UML?** 2019. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-uml>. Acesso em: 20 maio 2023.

LUCIDCHART (Utah, EUA). **Diagrama de caso de uso UML: O que é, como fazer e exemplos.** [20--]. Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>. Acesso em: 20 maio 2023.

LUCIDCHART (Utah, EUA). **O que é diagrama de atividades UML?** [20--]. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml>. Acesso em: 20 maio 2023.

LUCIDCHART (Utah, EUA). **O que é um diagrama de classe UML?**. [20--]. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>. Acesso em: 20 maio 2023.

LUCIDCHART (Utah, EUA). **O que é um diagrama de sequência UML?**. [20--]. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml>. Acesso em: 21 maio 2023.

LUCIDCHART (Utah, EUA). **O que é um diagrama de implementação?**. [20--]. Disponível em: https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-implementacao-uml#discovery_top. Acesso em: 21 maio 2023.