



**UNIVERSIDADE PAULISTA**

**ICET - INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E  
DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADO MULTIDISCIPLINAR**

**PIM III**

**Levantamento e análise de requisitos de um sistema para  
automatização total da folha de pagamento de uma empresa**

**Nome:**

Cainã De Oliveira Silva Leite  
João Pedro William Almeida Machado  
Júlio César Batista Junior  
Marcus Vinicius Araújo Filho  
Rafaela Ribeiro de Matos

**RA:**

F34DBI3  
G006682  
G079BF0  
T612BC5  
N014083

**SÃO JOSÉ DOS CAMPOS – SP**

**JUNHO/2023**

	<b>RA</b>
Aluno: Cainã De Oliveira Silva Leite	F34DBI3
Aluno: João Pedro William Almeida Machado	G006682
Aluno: Júlio César Batista Junior	G079BF0
Aluno: Marcus Vinicius Araújo Filho	T612BC5
Aluno: Rafaela Ribeiro de Matos	N014083

## **Levantamento e análise de requisitos de um sistema para automatização total da folha de pagamento de uma empresa**

Projeto Integrado Multidisciplinar (PIM) desenvolvido como exigência parcial dos requisitos obrigatórios à aprovação semestral no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da UNIP (Universidade Paulista), orientado pelo corpo docente do curso.

**São José dos Campos – SP**

**JUNHO/2023**

## RESUMO

Durante o processo de desenvolvimento, também buscamos garantir a eficiência e a segurança do sistema, considerando a importância da gestão de dados sensíveis relacionados à folha de pagamento. Implementamos medidas de proteção e criptografia para preservar a confidencialidade e integridade das informações.

Além disso, dedicamos atenção especial à usabilidade da interface, projetando-a de forma intuitiva e amigável para facilitar a interação dos usuários com o sistema. Realizamos testes e iteramos sobre o design, buscando aprimorar a experiência do usuário e minimizar possíveis dificuldades de uso.

Após a conclusão do projeto, fornecemos treinamento e suporte aos usuários, garantindo que eles possam utilizar plenamente o novo sistema e aproveitar seus benefícios. Também estabelecemos um processo de monitoramento contínuo e atualizações periódicas para garantir a sua funcionalidade e adaptabilidade às mudanças nas políticas e regulamentações relacionadas à folha de pagamento.

Dessa forma, o projeto de automação da folha de pagamento resultou em uma solução eficaz, alinhada às necessidades do cliente, capaz de otimizar o processo de pagamento dos funcionários, garantindo precisão, segurança e eficiência em toda a operação.

Palavras-Chave: Diagrama, Dados, informação, Heurística, banco, interface, UML, usabilidade, RH

## Sumário

Palavras-Chave:.....	3
<b>1. INTRODUÇÃO.....</b>	<b>5</b>
<b>1. ENGENHARIA DE SOFTWARE:.....</b>	<b>7</b>
2.1 Crise de software.....	7
2.2 Função da engenharia do desenvolvimento de software .....	8
2.3 Exemplos de modelo e desenvolvimento .....	9
2.3.1 Modelo em cascata .....	9
2.3.2 Modelo incremental .....	9
2.3.4 Modelo baseado em componentes .....	9
2.3.5 Processo unificado .....	9
2.4 REQUISITOS FUNCIONAIS .....	10
2.5 REQUISITOS NÃO FUNCIONAIS .....	10
2.6 UML (Unified Modeling Language): .....	10
2.6.1 Quando os diagramas são necessários?.....	11
2.6.1 Usos da UML.....	12
2.7 Etapas utilizadas para desenvolver um software.....	12
<b>3. ANÁLISE DE SISTEMAS ORIENTADA A OBJETOS .....</b>	<b>13</b>
3.1 Parceiros chaves.....	13
3.2 Atividades chaves.....	13
3.3 Recursos chaves .....	14
3.4 Proposta de valor .....	14
3.5 Relações com o cliente .....	14
3.6 Canal .....	15
3.7 Segmentos de Clientes .....	15
3.8 Estrutura de Custo .....	15
3.9 Fluxo de Receita .....	15
<b>4. PROGRAMAÇÃO ORIENTADA A OBJETOS.....</b>	<b>16</b>
4.1 Encapsulamento.....	16
4.2 Herança.....	16
4.3 Polimorfismo .....	16
4.4 Abstração .....	17
<b>5.0 PROJETO DE INTERFACE DO USUÁRIO: .....</b>	<b>18</b>
<b>6. BANCO DE DADOS .....</b>	<b>21</b>

<b>6.1 Funcionamento de um banco de dados .....</b>	<b>21</b>
<b>6.2 Lei Geral de proteção de dados .....</b>	<b>21</b>
<b>7. ECONOMIA E MERCADO .....</b>	<b>23</b>
<b>8.0 GESTÃO ESTRATÉGICA DE RH .....</b>	<b>24</b>
<b>8.1 Diferença entre o RH tradicional e RH estratégico.....</b>	<b>24</b>
<b>8.2 Quando o RH deve usar uma gestão estratégica.....</b>	<b>25</b>
<b>8.3 Características do RH estratégico .....</b>	<b>26</b>
<b>8.4 Como colocar em prática a gestão estratégica de RH.....</b>	<b>26</b>
<b>8.4.1 Benefícios de implementar a gestão estratégica de RH .....</b>	<b>27</b>
<b>9.0 DESENVOLVIMENTO DO PROJETO: .....</b>	<b>27</b>
<b>9.1 Reunião.....</b>	<b>30</b>
<b>9.1.1 Levantamento de requisitos para o desenvolvimento do sistema:</b>	<b>30</b>
<b>9.1.2 Lista de requisitos não-funcionais:.....</b>	<b>31</b>
<b>9.3 ESCOLHA DA EQUIPE.....</b>	<b>35</b>
<b>9.3.1 Treinamento da equipe.....</b>	<b>35</b>
<b>9.4 ÍNICIO DO DESENVOLVIMENTO DO PROJETO .....</b>	<b>35</b>
<b>9.4.1 Escolha da linguagem .....</b>	<b>36</b>
<b>9.4.2 Escolha do sistema para programar a interface do usuário:.....</b>	<b>36</b>
<b>9.4.3 Desenvolvimento do banco de dados do sistema .....</b>	<b>39</b>

## 1. INTRODUÇÃO

No começo do século XXI poderíamos comparar o advento da internet com o que foi a máquina a vapor no século XVIII, pois esta tornou processos mais eficientes e se inseriu dentro do meio corporativo e industrial de modo completamente irreversível. [Calvete,2022]

Esse fenômeno trouxe aos dias atuais o conceito da indústria 4.0, que consiste basicamente na aplicação da internet não somente no que diz respeito ao gerenciamento dos setores e na comunicação entre os funcionários através de uma rede particular desta dada indústria, mas também permitindo o gerenciamento de sensores e atuadores de modo remoto. (Cardoso, 2016)

Paralelo a isso, tornou-se real a presença da internet em aparelhos domésticos utilizados no dia a dia como lâmpadas, geladeiras, aparelhos de Ar-Condicionado e televisores, criando o chamado “IoT” que traduzido do inglês é descrito como “Internet das Coisas”, este cria a possibilidade de objetos do cotidiano serem controlados e monitorados remotamente através de aplicativos de celular ou *Websites* por usuários comuns, ou seja, diferente da indústria 4.0 que cria a ideia de uso da tecnologia da internet por grandes empresas, o IoT traz o mesmo efeito para cidadãos comuns e torna a vida destes muito mais confortável, pois facilita processos do dia a dia como controlar a iluminação de uma residência ou até mesmo gerenciar o estoque de alimentos em uma geladeira de uma família qualquer. (Rose, 2015)

A popularização do IoT em residências comuns tornou possível o surgimento do mercado de automação residencial que consiste justamente na venda e instalação de produtos que forneçam ao cliente o total controle de sua residência em setores como iluminação, climatização, cortinas, persianas, áudio e vídeo, ilustrado na Figura 1.

Controle este que pode ser feito através de aplicativos de celular, comando de voz ou até sensores que tomam decisões ao detectar a mudança de alguma determinada variável, permitindo além dos comandos básicos um disparo de comandos em sequência através de um único clique para que o morador possa por exemplo preparar toda uma sala de cinema da forma que desejar em termos de temperatura local, iluminação e fonte de vídeo na televisão para uma experiência ideal de assistir um filme, sem a necessidade que o usuário entenda todo o processo por trás da configuração daquele cenário. Ou seja, essa é a chamada automação residencial, facilitar o dia a dia das pessoas em tarefas do cotidiano. (Cruz, 2022)



Figura 1: Exemplo de uso de automação residencial (Bajac, 2020)

Fonte: <https://bassautomacao.com.br/equipamentos-para-automacao-residencial/>

Baseados nestes conceitos foi proposto para este documento uma empresa fabricante de produtos para automação residencial, sendo considerados três principais setores (Suporte técnico, setor comercial e desenvolvimento) contendo um diretor de dois funcionários cada, assim como três produtos à venda, a central, um gateway IR e RF para controle dos equipamentos que contenham essa tecnologia e módulo Relé para controle da iluminação.

## 1. ENGENHARIA DE SOFTWARE:

O termo engenharia de software foi utilizado a primeira vez pelo professor Friedrich Ludwig Bauer em 1968 no meio da primeira conferência dedicada ao assunto patrocinada pelo *NATO Science Committee* (NAUR & RANDELL, 1969).

O surgimento do termo se deu em decorrência a uma análise feita na época em que as condições da indústria de software estavam em um período crítico, entrando em colapso, momento esse que ficou conhecido como crise do software. (PRESSMAN, 1995).

### 2.1 Crise de software

A *crise de software* foi uma decorrência da imaturidade do mercado e dos profissionais da computação da época, pois vinha de um período onde o desenvolvimento do software não exigia requisitos e configurações complexas, sua utilização era, em média, limitada ao ambiente em que era desenvolvido. De fato, em meados de 1965 o termo *crise de software* não havia sido usado, isto ocorreu durante a década de 1970 quando as dificuldades relacionadas ao desenvolvimento do *software* começaram a ser mais graves, principalmente no aumento das demandas e da complexidade que o *software* passava a ter que executar, frente a inexistência de técnicas adequadas para resolver tais desafios (ENGHOLM JR, 2010, p. 32-33). Foi a partir desta lacuna que os princípios da *Engenharia de Software* tomaram forma e passou-se a considerar o que apregoava Bauer, o *software* passou a partir de então a ser visto como um produto e como tal necessita ser desenvolvido a partir de critérios de produção acentuados na busca de sua qualidade, custo adequado e entregue dentro dos prazos prometidos (PRESSMAN, 1995).

A crise de software foi uma decorrência de imaturidade do mercado e dos profissionais da computação da época, pois vinha de um período em que o desenvolvimento do software não exigia requisitos e configurações complexas a utilização do software era limitada ao ambiente em que era desenvolvido. Na realidade, a crise de software não tinha esse nome em 1965, o termo foi ganhando força em 1970 quando lidar com o software no dia a dia foi se tornando mais difícil, as dificuldades foram se tornando ainda mais frequentes e graves, principalmente a complexidade em que o software passava a ter que executar diante das inexistências de técnicas adequadas para resolver os desafios (ENGHOLM JR, 2010, p. 32-33).



Foi a partir desta lacuna que os princípios da *Engenharia de Software* tomaram forma e passou-se a considerar o que apregoava Bauer, o *software* passou a partir de então a ser visto como um produto e como tal necessita ser desenvolvido a partir de critérios de produção acentuados na busca de sua qualidade, custo adequado e entregue dentro dos prazos prometidos (PRESSMAN, 1995).

## **2.2 Função da engenharia do desenvolvimento de software**

A *Engenharia de Software* foca sua preocupação em manter o controle sobre todas as fases do processo de desenvolvimento do *software* por meio de métricas voltadas ao controle produtivo dessas aplicações. Esse controle é calcado sob a ótica de uso de diversos paradigmas (modelos metodológicos) de controle e acompanhamento. Cada paradigma, em particular, possui características relacionadas ao tipo de *software* desenvolvido ou ao tamanho que certa aplicação deverá possuir para atender as necessidades de seus clientes. Essas preocupações começaram a levar os analistas de sistemas de simples gerentes de projetos que pensavam e olhavam apenas a aplicação das regras computacionais em seus projetos para que o *software* atendesse o mínimo da necessidade dos clientes a um nível superior de ordem onde se soma as preocupações tradicionais outras como o custo da produção do *software* e a qualidade efetiva, além de ter que se preocupar com os métodos de desenvolvimento e suas ferramentas de aplicação. Frente a isso a definição sobre *Engenharia de Software* apregoada por Bauer toma força como mostra (PRESSMAN, 1995).

A engenharia de software foca a preocupação em manter o controle sobre todas as fases do processo de desenvolvimento de software por meio das métricas voltadas ao controle produtivo dessas aplicações. O controle é feito através do uso de diversos modelos metodológicos de controle e acompanhamento. Cada metodologia em particular tem as características definidas a partir do tipo de software e o tamanho do sistema a ser desenvolvido, levando em consideração que necessita atender as necessidades dos clientes. Ao enxergar essas maiores necessidades, os analistas e desenvolvedores passaram a detalhar mais ao pensar no desenvolvimento do projeto, como os métodos e as ferramentas de aplicação. Após isso, a definição de engenharia de software ganhou força (PRESSMAN, 1995).

As metodologias de desenvolvimento de software são propostas para colocar ordem no momento de desenvolver um software. São os processos metodológicos que acomodam atividades genéricas, mas também tem como função guardar a particularidade que cada método necessita para controlar o projeto a que é submetido. (PRESSMAN, 2006). O

objetivo principal do uso de tais métodos é garantir uma melhor qualidade processual para o desenvolvimento do software. devido a esta característica operacional presume-se serem esses métodos ferramentas que auxiliam a obtenção da qualidade final no processo de desenvolvimento, pois é melhor uma ferramenta simples que garanta um mínimo de mensuração da qualidade do que não ter absolutamente nada com que medir e acompanhar. (PRESSMAN, 1995).

## **2.3 Exemplos de modelo e desenvolvimento**

### **2.3.1 Modelo em cascata**

proposta de acompanhamento sequencial e linear no processo de desenvolvimento sendo útil quando os requisitos do sistema são estáveis e bem definidos, usados em projetos de pequeno porte com baixa margem de manutenção;

### **2.3.2 Modelo incremental**

proposta de produção de software a partir de diversas versões e incrementos, sendo adequado para a produção de projetos maiores em espaço de tempo limitado;

### **2.3.3 Modelo evolucionário**

proposta usada para acomodar modificações que sejam constantes na fase de desenvolvimento do projeto, sendo esta que melhor usa a natureza iterativa da maioria dos projetos de engenharia de software adequada para projetos maiores;

### **2.3.4 Modelo baseado em componentes**

proposta baseada na reutilização de componentes prontos de outros projetos, desde que possam ser conectados ao projeto em desenvolvimento, adequado para projetos maiores;

### **2.3.5 Processo unificado**

proposta de projeto baseada na orientação de casos de uso centrado na arquitetura incremental e iterativa, adequado para projetos maiores.

## **2.4 REQUISITOS FUNCIONAIS**

Requisitos Funcionais, como o nome sugere, descrevem as funções do sistema a ser projetado. É uma descrição do que o sistema será e como ele funcionará para satisfazer as necessidades do usuário. Eles fornecem uma descrição clara de como o sistema deve responder a um comando específico, os recursos e o que os usuários esperam. São eles que garantem que a equipe inteira entenda a forma como deve ser desenvolvido o sistema e assim garantindo o sucesso do projeto.

## **2.5 REQUISITOS NÃO FUNCIONAIS**

Os requisitos não funcionais explicam as limitações e restrições do sistema a ser projetado. Esses requisitos não têm nenhum impacto na funcionalidade do aplicativo. Além disso, existe uma prática comum de classificar os requisitos não funcionais em várias categorias, como:

- Interface de Usuário
- Confiabilidade
- Segurança
- atuação
- Manutenção
- Standards

## **2.6 UML (Unified Modeling Language):**

É uma linguagem de notação destinada à modelação e documentação das fases de desenvolvimento de softwares orientados a objetos.

Utilizando uma série de elementos gráficos, como retângulos, setas e linhas, a UML consegue criar diagramas para representar as áreas de um software, suas interações e mudanças. Em outras palavras, ela fornece uma espécie de “desenho” para auxiliar a equipe do projeto a visualizar os aspectos do programa e facilitar a construção.

Assim, os desenvolvedores podem ter uma visão geral do trabalho em diagramas padronizados, evitando problemas comuns no desenvolvimento, como erros nas fases de implementação ou na comunicação entre os envolvidos. Afinal, como se trata de uma linguagem padrão, objetiva e eficiente, ela pode ser facilmente entendida.

Tipos de diagramas UML: os diagramas são divididos em dois grandes grupos: estruturais e comportamentais. Veja abaixo mais detalhes sobre eles:

- Estruturais: os diagramas estruturais conseguem modelar os aspectos estáticos do software, como métodos, interfaces, serviços, arquitetura e classes, por exemplo;
- Comportamentais: os diagramas comportamentais especificam detalhes do comportamento do software para exemplificar como determinadas funções deverão funcionar, incluindo os possíveis métodos de interação dos usuários.

### 2.6.1 Quando os diagramas são necessários

Apesar de a UML ser uma linguagem importante para agilizar o processo de desenvolvimento de um software, ela não é necessariamente obrigatória. Na verdade, seus diagramas costumam ter um uso padrão.

Na maioria dos casos, eles são inseridos em projetos onde os desenvolvedores precisam de uma visão padronizada do trabalho e das especificações; para documentar e visualizar o funcionamento do software ou auxiliar na fase inicial de implementação dos principais recursos.

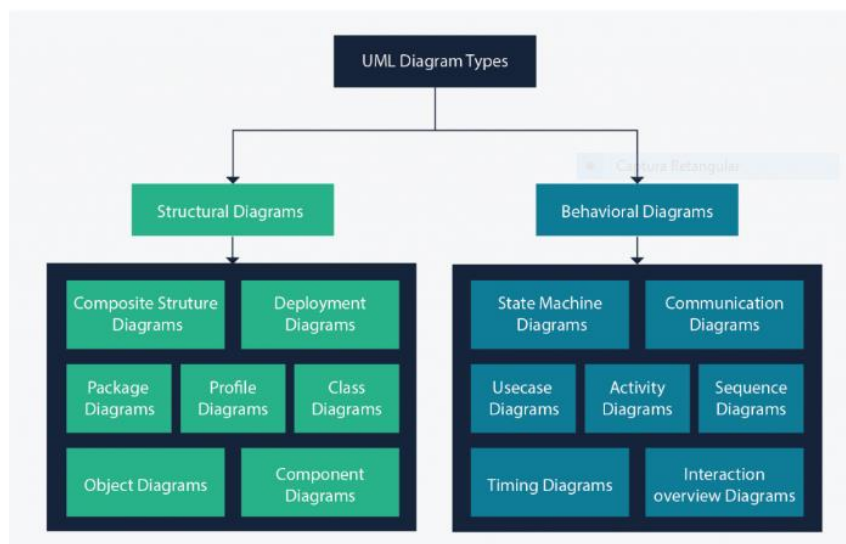


Figura 2: tipos de diagrama de UML

Fonte: <https://tecnoblog.net/responde/o-que-e-uml/>

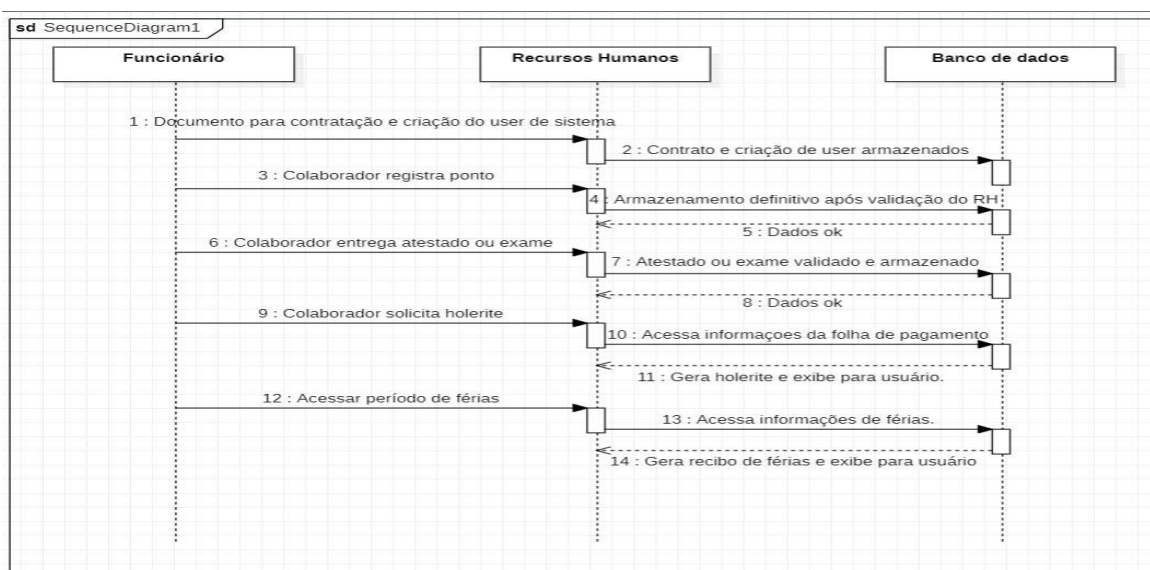
## 2.6.1 Usos da UML

Geralmente, a UML é aplicada em sistemas complexos de softwares, atuando em áreas como:

- Sistemas corporativos;
- Serviços bancários e financeiros;
- Telecomunicações;
- Transportes;
- Defesa/espaço aéreo;
- Vendas de varejo;
- Sistemas médicos;
- Serviços de distribuição

Modelagem UML – Diagrama de Caso de Uso A UML (Unified Modeling Language) é uma linguagem visual padrão utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos. Para a criação do Diagrama de Caso de Uso, foi utilizada a ferramenta livre JUDE UML. A Figura 3 apresenta o Caso de Uso do Módulo Recursos Humanos e Financeiro. O Diagrama de Caso de Uso possui as seguintes características: PRINCIPAIS ATORES: Cliente, Fornecedor, Funcionário, Funcionário RH e Gerente PRINCIPAIS CASOS DE USO: Gerar Nota Fiscal, Gerar Folha de Pagamento, Consultar Funcionário e Gerar Relatório.

## 2.7 DIAGRAMA DE SEQUÊNCIA



### 3. ANÁLISE DE SISTEMAS ORIENTADA A OBJETOS

Um dos principais nomes quando falamos em modelo de negócio é o do Alexander Osterwalder, ele criou o modelo Canvas em 2004, na Universidade de Lusanne, na Suíça, durante sua formação em doutorado em Sistema de Informação Gerencial. Esse modelo passou a ser a principal ferramenta a ser utilizada. De acordo com o livro "Business Model Generation - Inovação em Modelos de Negócios" existem nove elementos-chave que formam um modelo de negócios bem-sucedido: Proposta de Valor, Segmentos de Clientes, Canais de Distribuição, Relacionamento com Clientes, Fontes de Receita, Recursos Principais, Atividades-Chave, Parcerias Principais e Estrutura de Custos".

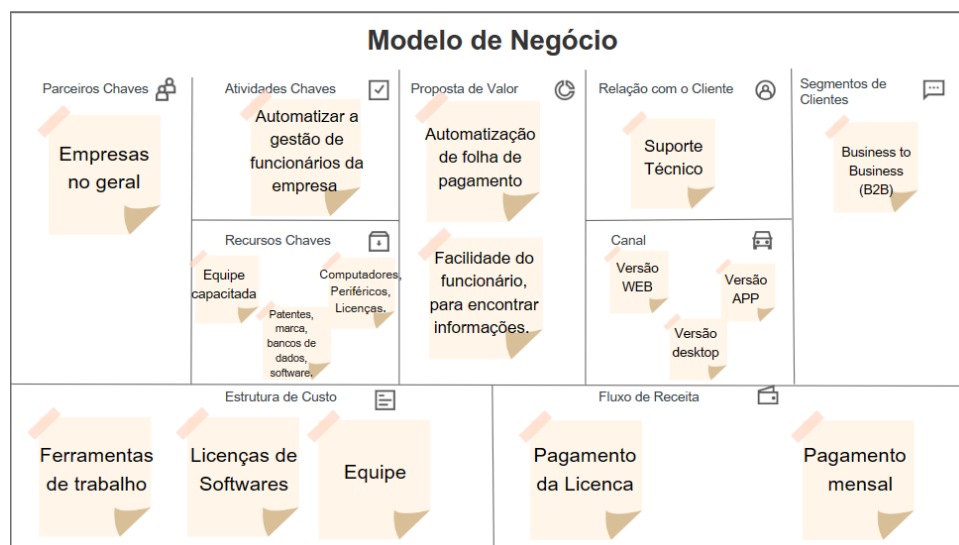


Figura 3: modelo de negócios

Fonte: os autores.

#### 3.1 Parceiros chaves

Empresas no geral: Nosso software abrange todo tipo de *software*, ele pode ser adaptado para qualquer empresa.

#### 3.2 Atividades chaves

Automatizar a gestão de funcionários da empresa: O intuito do nosso sistema é facilitar a gestão da folha de pagamento da empresa, tornando o processo mais automatizado.

### **3.3 Recursos chaves**

Equipe capacitada: ter uma equipe tecnicamente capacitada e organizada no projeto é essencial para o seu sucesso.

Computadores, Periféricos, Licenças: os computadores são o meio no qual vamos realizar as atividades referentes ao software, então é a principal ferramenta de trabalho. Os periféricos servem para auxiliar nesse quesito, facilitando a rotina do funcionário e as licenças seriam o acesso as ferramentas para o desenvolvimento do *software*.

Patentes, marcas, bancos de dados e *software*: as patentes vão ser usadas para o registro do nome do software, a marca vai ser a identidade visual do nosso software, banco de dados vai ser onde iremos armazenar os dados que forem registrados no nosso software, e o software vai ser o sistema em si, que será implementado nas empresas.

### **3.4 Proposta de valor**

Automatização de folha de pagamento: nossa proposta é automatizar o processo de folha de pagamento da empresa, facilitando a sua gestão. Exemplo: seria no demonstrativo de pagamento, ficaria acessível no nosso software sem a necessidade do funcionário de RH ter que separar um por um na época de declarar o imposto de renda.

Facilidade do funcionário, para encontrar informações: o funcionário teria as informações de que ele precisaria na palma da mão dele, ou em alguns cliques, sem precisar solicitar para o departamento do RH.

### **3.5 Relações com o cliente**

Suporte técnico: prestaremos um suporte para o cliente, auxiliando em possíveis erros, bugs e em novas atualizações. Também o instruindo, com treinamentos de como usar o sistema.

### **3.6 Canal**

Versão WEB: vai ser a versão acessada através dos navegadores.

Versão APP: vai ser a versão criada para smartphones.

Versão desktop: vai ser a versão criada para computadores.

### **3.7 Segmentos de Clientes**

*Business to Business (B2B)* - O segmento que nossa empresa vai atuar, tem como intuito auxiliar na gestão de outras empresas, sendo assim, podemos classificar o modelo de negócio escolhido como "B2B".

### **3.8 Estrutura de Custo**

Ferramentas de trabalho: um dos pilares, que temos como principal na nossa empresa, são as ferramentas de trabalho dos novos funcionários, para que tenham condições de desempenhar o máximo do seu potencial.

Licenças de Softwares: esse custo é essencial, para que nossos colaboradores tenham acesso as melhores ferramentas do mercado.

Equipe: ter uma boa equipe é essencial para uma empresa de sucesso, pois contribui para a eficiência na realização de projetos, na produtividade referente as atividades e nas inovações, temos como ponto positivo também a resolução de problemas, melhora do ambiente de trabalho e compartilhamento de conhecimento entre os colaboradores.

### **3.9 Fluxo de Receita**

Pagamento da Licença: para que o cliente possa utilizar nosso sistema, ele terá de custo inicial, o pagamento de uma taxa que será a de licença de uso do software.

Pagamento mensal: depois de adquirir a licença, será cobrado mensalmente uma taxa referente às atualizações e suporte.



## 4. PROGRAMAÇÃO ORIENTADA A OBJETOS

Uma classe tem como objetivo dar características e comportamentos ao objeto. Então podemos dizer que o objeto é uma instância de uma classe. Como características, temos dois tipos de variáveis, que são "atributos" ou "propriedades", são nelas que ficam armazenados os dados associados ao objeto. Já os comportamentos, são representados por "métodos", que são as funções que definem qual será esse comportamento.

A Programação Orientada a Objetos, segue alguns princípios fundamentais, que são:

### 4.1 Encapsulamento

Esse princípio tem como objetivo, proteger os dados de uma classe, para que eles não sofram alterações indevidas, sendo assim, ele pode ser acessado, somente por métodos específicos que seriam o "get" e "set". Eles têm como objetivo, utilizar os dados da classe de maneira controlada. O "get" tem como intuito, retornar o valor de um atributo da classe, já o "set", tem como objetivo alterá-lo.

### 4.2 Herança

Uma das principais técnicas usadas na Programação orientada a objetos, é a herança, esse conceito, permite criar classes a partir de classes existentes.

A Herança permite criar hierarquias de classes e organizar o código de forma mais estruturada.

A classe base é chamada de classe pai ou superclasse, já a classe derivada dela é chamada de classe filha ou subclasse. A classe filha herda todos os atributos e métodos da classe pai, permitindo adicionar ou sobrescrever os mesmos.

### 4.3 Polimorfismo

O Polimorfismo é uma das principais vantagens da programação orientada a objetos, porque ele nos permite, tornar o código mais flexível e adaptável em diferentes situações.

Ele é um conceito que permite usar um mesmo método em diferentes tipos de objetos, permitindo que objetos de diferentes classes possam ser tratados de maneira semelhante.

Existem dois tipos de polimorfismo, Polimorfismo de sobreposição (*override*) e Polimorfismo de Sobrecarga (ou *Overload*).

O polimorfismo de sobreposição, ocorre quando uma subclasse reescreve o método de uma superclasse, realizando-o de forma diferente.

Já o Polimorfismo de Sobrecarga é quando o método possui o mesmo nome, porém diferentes parâmetros em uma mesma classe ou em classes diferentes.

#### 4.4 Abstração

A abstração é utilizada para simplificar um objeto, destacando as características principais e ignorando informações irrelevantes para seu funcionamento. Ela é utilizada para criar classes do mundo real ou conceitos abstratos.

Ela também promove a reutilização do código, tornando a interação dele mais simples e eficiente, pelo fato de ignorar informações irrelevantes.

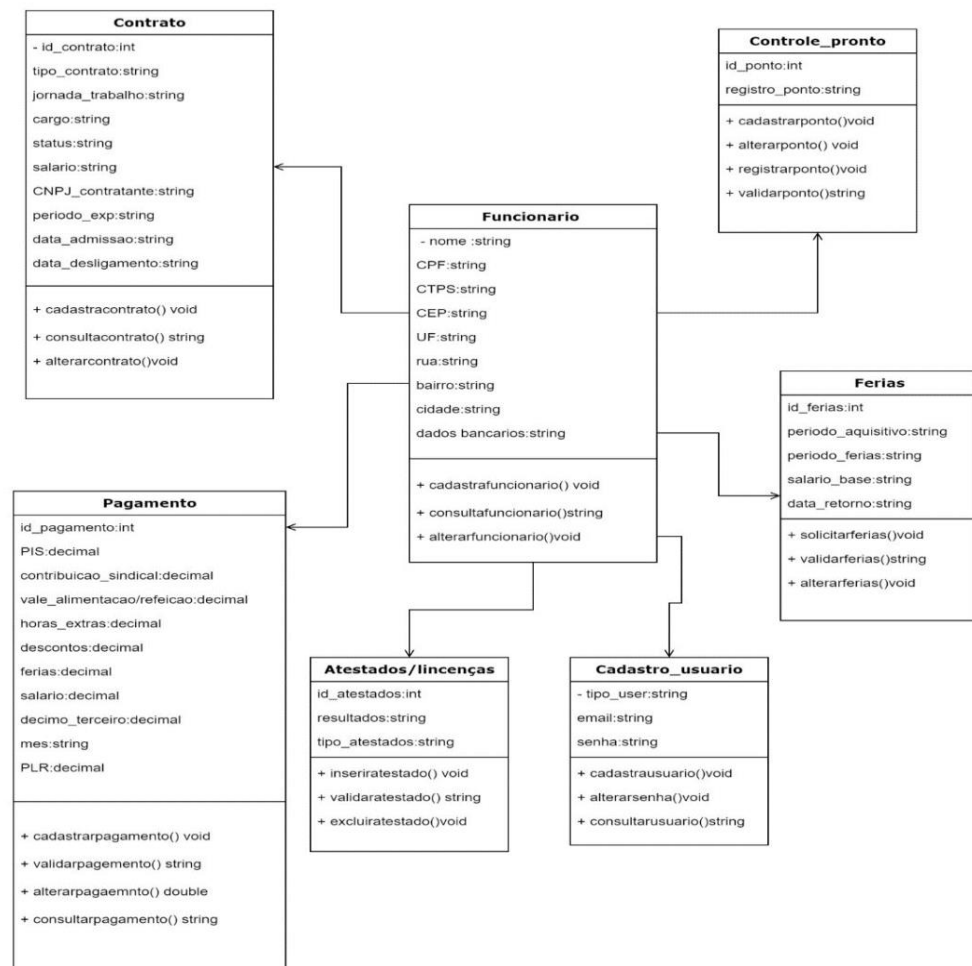


Figura 4: diagrama de classes.

Fonte: o autor

## 5.0 PROJETO DE INTERFACE DO USUÁRIO:

No processo de criação e desenvolvimento de um software a importância do design em todo o projeto de interface possui e tem um papel essencial para atender as necessidades, expectativas e a capacidade de facilitar a compreensão, simplificar o objetivo e propósito do usuário e tornar intuitivo, moderno, transformador e sem deixar de lado a beleza atrativa do software. Todo esse processo de implementação visual e funcional seguindo a linha de raciocínio de negócio do projeto e deixando o software atraente, é possível aumentar o tráfego de usuários em um determinado local. (Norman, 2006).

O chamado “intuitivo” é um conhecimento já adquirido ao longo da vida portanto, é de fácil entendimento ao se deparar com tal informação “... as pessoas aprendem melhor e sentem-se mais à vontade quando o conhecimento exigido para a execução de uma tarefa está disponível externamente: explícito no mundo ou prontamente inferível para ser recebido por meio de coerções. Mas o conhecimento no mundo só é útil se houver um relacionamento natural, interpretável com facilidade, entre esse conhecimento e as informações que ele pretende transmitir sobre possíveis ações e resultados” (Norman, 2006).

“O design deve fazer uso das características inerentes naturais das pessoas e do mundo, deve explorar os relacionamentos naturais e as coerções naturais. Tanto quanto possível, deve operar sem instruções ou rótulos. Qualquer instrução ou treinamento necessário só deve precisar ser dado uma única vez; a cada explicação, a pessoa deve poder dizer: "É claro" ou "Sim, compreendo" Uma explicação simples basta se houver racionalidade no design, se tudo tiver seu lugar e sua função, e se os resultados das ações forem visíveis. Se a explicação induzir a pessoa a pensar ou a dizer "Como vou conseguir me lembrar disso?", o design terá falhado.” (Norman, 2006)

Jacob Nielsen, conhecido como o guru da usabilidade, é um cientista da computação e especialista em interface para usuário, é o autor das famosas 10 Heurísticas de Nielsen, criadas no intuito de alcançar a melhor experiência e usabilidade possível. São elas as seguintes:

“Visibilidade do Status do sistema. O sistema deve sempre manter os usuários informados sobre o que está acontecendo, em tempo real, por meio de feedbacks instantâneos dentro de um prazo razoável, que servirão para orientar o usuário sobre o que está acontecendo na tela em que ele se encontra.

Correspondência entre o sistema e o mundo real. O sistema deve ter a mesma linguagem que seu usuário utiliza no dia a dia, portanto utilizar palavras, frases, imagens e conceitos que sejam familiares ao usuário. Também é recomendável a utilização de ícones que representem uma ação para realizar essa correspondência.

Controle e liberdade para o usuário. No sistema é importante que o usuário tenha liberdade para realizar ações que ele deseja, porém muitas vezes essas ações são realizadas por engano, nessas situações deve haver uma “saída de emergência” de fácil localização, permitindo que o usuário saia daquela janela indesejada ou retorne ao ponto anterior.

Consistência e padronização. Não devemos deixar o usuário em dúvida se palavras, situações ou ações diferentes tem o mesmo significado, para isso a interface deve manter uma consistência, permitindo que o usuário identifique os padrões de estética, interação e informação existentes nesta.

Prevenção de erros. Essa heurística foca em criar uma plataforma que busca prevenir que problemas ocorram, eliminando condições mais propensas ao erro, isso pode ser feito sinalizando ações como quando o usuário não finalizou uma tarefa e pede para sair da página ou oferecendo a ele uma opção de confirmação antes de completar uma ação.

Reconhecimento em vez de recordação. Devemos minimizar a quantidade de informações que o usuário precisa memorizar, para isso, objetos, ações e opções importantes devem ficar visíveis, isso faz com que o cérebro perceba as ações que são similares, reconhecendo assim padrões.

Eficiência e flexibilidade de uso. A interface deve atender tanto as necessidades dos usuários leigos quanto a dos experientes, os leigos precisam ter as informações bem detalhadas para conseguirem realizar uma tarefa, mas a medida que vão conhecendo a interface precisam conseguir interagir de forma mais rápida, portanto a interface deve permitir que os usuários adaptem ações frequentes de acordo com o nível em que estão. Exemplos de atalhos de teclado que o Windows oferece

Estética e design minimalista. Quanto maior a quantidade de informações, mais tempo o usuário vai levar para analisá-las e poder tomar uma decisão, aumentando assim as chances de ele abandonar a aplicação/site por achar confuso demais.

Ajude os usuários a reconhecer, diagnosticar e recuperar erros. Caso dê algo de errado, é importante mostrar ao usuário qual foi o erro e como se recuperar dele, para isso, as mensagens de erro devem ser claras e objetivas, em linguagem simples e próximas da ação que causou o erro.

Ajuda e documentação. Nunca sabemos quando um usuário vai precisar de um auxílio, embora as áreas de documentação e ajuda sejam as menos acessadas, elas devem estar lá, principalmente em interfaces que possuem muitas possibilidades, pois podem ajudar o usuário a resolver um problema sozinho.” (Pinhata, Gabriela, 2018)

O sistema como um todo segue a característica de ser responsivo, que nos dias de hoje é ser sinônimo de inovação e modernidade, trazendo sua flexibilidade. As versões para web, desktop e mobile possuem características similares e seguem os mesmos princípios estéticos e funcionais. Isso garante uma experiência consistente ao usuário, independentemente do dispositivo ou plataforma que ele esteja utilizando. As alterações realizadas e versões otimizadas podem ser feitas facilmente e aplicadas umas às outras sem esforço e se mantendo coerentes. (Faria, 2016)

## 6. BANCO DE DADOS

### 6.1 Funcionamento de um banco de dados

Um banco de dados permite o armazenamento, gerenciamento e recuperação de informações de maneira estruturada. Sua principal funcionalidade é organizar uma quantidade significativa de dados de forma eficaz, isso permite que os usuários tenham acesso rápido e seguro a esses dados. Os bancos de dados são formados por tabelas, que são dados ordenados em linhas e colunas. Cada registro representa uma entidade e cada campo contém um atributo específico dessa entidade. Os dados são armazenados em formato tabular, o que facilita a consulta e manipulação dos dados.



Figura 5: ilustração de banco de dados

Fonte: <https://www.cortex-intelligence.com/blog/inteligencia-de-mercado/banco-de-dados-tecnologia>

### 6.2 Lei Geral de proteção de dados

A construção de um software que envolve um grande número de partes interessadas que inclui usuários que são pessoas físicas, a segurança da informação deve ser primordial e bem executada, sendo contempladas pela lei de nº 13.709 em 14 de agosto de 2018, Lei Geral de Proteção de Dados Pessoais (LGPD), sancionada pelo presidente da República Michel Temer. Em seu primeiro artigo possui em seu propósito:

Art. 1º Esta Lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natural. (LGPD,2018)

Sobre o tratamento dos dados, a empresa deverá seguir normas e requisitos dispostos no artigo 7 e 8, mediante ao consentimento do usuário no ato da contratação:

Art. 7º O tratamento de dados pessoais somente poderá ser realizado nas seguintes hipóteses: I - mediante o fornecimento de consentimento pelo titular; II - para o cumprimento de obrigação legal ou regulatória pelo controlador. (LGPD, 2018)

Entre termos e disposições necessárias de forma coerente e explícita, conforme também:

Art. 8º O consentimento previsto no inciso I do art. 7º desta Lei deverá ser fornecido por escrito ou por outro meio que demonstre a manifestação de vontade do titular; § 1º Caso o consentimento seja fornecido por escrito, esse deverá constar de cláusula destacada das demais cláusulas contratuais. (LGPD, 2018)

Medidas de cautelas são apresentadas nos artigos 46,47,48, no qual se diz respeito ao cumprimento da empresa para os deveres de organização e proteção:

Art. 46. Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito. Art. 47. Os agentes de tratamento ou qualquer outra pessoa que intervenha em uma das fases do tratamento obriga-se a garantir a segurança da informação prevista nesta Lei em relação aos dados pessoais, mesmo após o seu término. Art. 48. O controlador deverá comunicar à autoridade nacional e ao titular a ocorrência de incidente de segurança que possa acarretar risco ou dano relevante aos titulares. (LGPD, 2018)

Em base o artigo 50, será criada uma política interna de boas práticas e governança por parte da empresa.

Art. 50. Os controladores e operadores, no âmbito de suas competências, pelo tratamento de dados pessoais, individualmente ou por meio de associações, poderão formular regras de boas práticas e de governança que estabeleçam as condições de organização, o regime de funcionamento, os procedimentos, incluindo reclamações e petições de titulares, as normas de segurança, os padrões técnicos, as obrigações específicas para os diversos envolvidos no tratamento, as ações educativas, os mecanismos internos de supervisão e de mitigação de riscos e outros aspectos relacionados ao tratamento de dados pessoais. (LGPD, 2018)

Para o cumprimento da lei, anonimato, segurança e proteção em geral em nossos bancos de dados e o software como um todo, a empresa se dedicará aos métodos de autenticação de usuário, para que somente pessoas autorizadas tenham acesso as informações de dados pessoais, sejam eles sensíveis ou não, terá uma política de backup, e definirá quais dados serão armazenados para um local seguro, serão feitas cópias de segurança para possibilitar a recuperação de dados em caso de necessidade e, por fim, também será dedicada atenção ao método de encriptação e criptografia que significa: “Encriptação significa a transformação de dados pelo uso de criptografia para produzir dados ininteligíveis (dados encriptados) para garantir sua confidencialidade. Criptografia significa a disciplina que incorpora princípios, meios e métodos para a transformação de dados a fim de ocultar seu conteúdo informativo, estabelecer sua autenticidade, impedir a sua modificação não detectada, impedir o seu repúdio e/ou impedir o seu uso não autorizado.” (SCHULZ; HOBOKEN, 2021) todos os dados passarão para o processo de encriptação e serão totalmente criptografados.

## **7. ECONOMIA E MERCADO**

Nosso modelo de monetização, é através de vendas de licenças de uso e taxa mensal para atualizações e suporte técnico.

O valor não será fixo e varia de acordo com o faturamento da empresa, para que o sistema seja personalizado de acordo com as necessidades do negócio.

Pensando em mercado, nossa empresa agrega valor para o cliente das seguintes formas:

- Aumento da eficiência operacional: Visando o aumento da eficiência operacional, o software poderá ser uma boa alternativa para empresas que buscam otimizar processos repetitivos e manuais, visto que tem o intuito de automatizar tarefas e buscar aumentar a produtividade. Um bom exemplo, seria o informe de rendimentos que as empresas precisam disponibilizar todos os anos, para a declaração do imposto de renda. Nesse caso, os usuários teriam acesso a um clique, facilitando a vida do usuário e também do time de RH.



- **Melhoria da qualidade e satisfação do cliente:** Conforme comentado no tópico acima, o cliente terá acesso à informação importantes para o seu dia a dia, com poucos cliques, podendo ser através do seu smartphone, ou através de computadores, o que tornaria sua vida, muito mais prática. Com a diminuição de tarefas manuais e repetitivas do departamento de RH, resultará em um aumento da produtividade, e também eficiência na utilização de recursos e redução de custos operacionais, aumentando a satisfação do cliente e de consequência melhorando a qualidade de diversas áreas do seu processo.

- **Agilidade e flexibilidade:** O software terá a flexibilidade de se adequar de acordo com as necessidades da empresa, essa personalização permitirá que ele se adeque aos padrões do mercado. Através da automação de processos conforme citamos nos tópicos acima, algumas demandas serão realizadas mais rapidamente, o que tornará os processos mais ágeis e eficientes.

## **8.0 GESTÃO ESTRATÉGICA DE RH**

A gestão estratégica de RH vem como uma solução que é implementada nas empresas para não apenas resolver problemas, mas sim como reconhecer um problema e solucioná-lo antes que ele venha a aparecer.

Essa prática afeta a organização, ela tem como prioridade melhora as rotinas e introduz uma nova forma de enxergar problemas e focar na solução. Como exemplo, a gestão estratégica de recursos humanos atua, principalmente, no dia a dia mais saudável do colaborar alinhando assim objetivos e necessidades. O intuito desse novo formato de gerir uma corporação é cultivar um ambiente de trabalho sadio e salubre e consequentemente mais produtivo. Com colaboradores mais engajados, maiores são os benefícios a corporação. (LAFZ, 2018)

### **8.1 Diferença entre o RH estratégico entre em RH tradicional e estratégico**

O RH tradicional, aquele mais utilizado tem muito pouco para agregar nos ambientes corporativos, ele está voltado unicamente para o crescimento e desenvolvimento do negócio. Diferente do RH tradicional, o RH com gestão estratégica tem como função ser um parceiro não gestão da organização por completo. O foco do RH estratégico é em pessoas, ele é direcionado para o desenvolvimento de ideias para melhorar o desempenho dos colaboradores e da empresa no geral. Ele atua ao lado dos demais gestores com o

objetivo de desenvolver ferramentas e soluções que possam beneficiar o crescimento da empresa. (G4 EDUCAÇÃO, 2021)

## **8.2 Quando o RH deve usar uma gestão estratégica**

Não há uma regra quanto ao uso da gestão estratégica, pois mesmo quando não há nenhum problema aparente para ser solucionado, uma estratégia bem estruturada pode melhorar os resultados. É por esse motivo que a mudança do chamado RH burocrático para o RH estratégico é evidente em boa parte das empresas. (PONTOTEL, 2023)

### **8.2.1 Casos em que o método é utilizado**

Baixo engajamento de um time: se uma equipe está apresentando sinais de baixo engajamento, é provável que algo esteja desestimulando aqueles profissionais. Os motivos podem ser diversos, desde baixa remuneração até abuso de liderança. Para solucionar a situação, apostar na gestão estratégica pode ser o caminho. (PONTOTEL, 2023)

Para isso, o RH precisa fazer um diagnóstico para entender o que está ocasionando os baixos índices de engajamento. A partir disso, traçar o planejamento e utilizar as ferramentas de gestão para acompanhar e analisar os resultados. (PONTOTEL, 2023)

Produtividade de uma equipe afetada: se a produtividade de uma equipe está baixa, provavelmente, algo não está certo. Porém, nem apenas a falta de engajamento pode ser a causa do problema. Pode ser que falte recursos para a realização do trabalho, algum líder que não esteja atuando corretamente ou ainda a alta demanda que está atrapalhando a organização dos colaboradores.

Por isso, fazer uma avaliação estratégica e um plano de ação para resolver os itens que estão impactando é essencial para a melhoria da produtividade. (PONTOTEL, 2023)

#### **Processos desatualizados e desestruturados na gestão**

Estamos vivendo uma era em que as tecnologias e os processos mudam a todo momento. Isso serve também para a maneira como as áreas atuam. Dependendo de como a gestão trabalhou até o momento, é possível que o setor de RH ainda aplique processos e/ou ferramentas que já não são mais utilizados no restante do mercado de trabalho. (PONTOTEL, 2023)

Entretanto, para fazer a transição entre o modelo desatualizado para um mais moderno, fazer uma gestão estratégica é essencial. Assim, tanto os profissionais do setor que terão sua rotina transformada completamente quanto os demais colaboradores que serão impactados indiretamente saberão cada passo e qual o objetivo da mudança. (PONTOTEL, 2023)

### **8.3 Características do RH estratégico**

Para estabelecer essa metodologia nas empresas são necessários quatro pilares para obter os melhores resultados. São eles: comunicação, visão, liderança e tecnologia. (REDAÇÃO ONZE, 2023)

- **Comunicação:** É preciso estabelecer um canal de comunicação direto com os colaboradores, não apenas para transmitir mensagens importantes, mas também para promover a interação e aumentar o sentimento de identificação. (REDAÇÃO ONZE, 2023)
- **Visão:** A gestão estratégica de RH deve ter uma visão aprofundada da empresa. Sua missão, sua visão e seus valores precisam ser conhecidos para que seja possível alinhar objetivos e traçar as melhores táticas para alcançá-los. (REDAÇÃO ONZE, 2023)
- **Liderança:** Toda equipe de trabalho precisa de gestores que sejam responsáveis por motivar os colaboradores e que sirvam de exemplo positivo para eles. Ou seja, as lideranças devem conscientizar os profissionais em busca das metas comuns. (REDAÇÃO ONZE, 2023)
- **Tecnologia:** Para se estabelecer uma gestão integrada de RH, a tecnologia é fundamental. Por meio do *People Analytics*, por exemplo, é possível criar um banco de dados e utilizar essas informações para fazer escolhas mais conscientes. (REDAÇÃO ONZE, 2023)

### **8.4 Como colocar em prática a gestão estratégica de RH.**

No dia a dia são desenvolvidas ações para atrair talentos para trabalhar na corporação e principalmente reter, manter os talentos. Para isso, é necessário criar um ambiente de trabalho onde exista boas práticas de humanização, capacitar novas

lideranças e atuarem e concordância com a cultura organizacional da empresa, entre outros. (REDAÇÃO ONZE, 2023)

#### **8.4.1 Benefícios de implementar a gestão estratégica de RH**

A gestão estratégica de Recursos Humanos oferece benefícios tanto para a empresa quanto para os colaboradores. Listamos algumas das vantagens mais importantes a seguir: (REDAÇÃO ONZE, 2023)

- Antecipação de problemas com foco na solução;
- Aumento da produtividade;
- Clima organizacional mais saudável;
- Alinhamento de objetivos e clareza na sua divulgação;
- Equilíbrio na delegação de tarefas e no compartilhamento de responsabilidades;
- Tomada de decisões mais acertadas;
- Melhoria contínua;
- Aprimoramento da comunicação interna;
- Diminuição no índice de turnover; e
- Engajamento dos colaboradores

#### **8.5 Manual do usuário:**

O login e senha para utilização do sistema será fornecido pelo RH

1- Acessar o site [WWW.BABBOB.COM.BR](http://WWW.BABBOB.COM.BR)

1.1- Coloque seu login com o nome de usuário e senha para a validação.

1.2- Caso esqueça a senha, selecionar a opção “Esqueci minha senha”.

1.3- Caso esqueça o nome de usuário, contatar o RH

Após o acesso aceito, será direcionado ao Menu Principal, onde se encontra os principais avisos e recados importantes, a esquerda se encontra a barra de menu, com as principais funcionalidades do sistema como: Perfil do Usuário; Folha de Pagamento; Controle de Ponto; Exames Médicos; Contrato de Trabalho, Férias e Ajuda.

2- Para acessar o atalho de menu na esquerda, clique na seta junto a coluna e aparecerá a descrição de cada funcionalidade.

2.1- Para acessar o seu perfil, clique em sua foto ou em seu nome localizadas na barra de menu.

2.2- Para acessar a folha de pagamento e ter acesso ao seu holerite e informe de rendimentos, clique em “Folha de Pagamento”.

2.3- Para acessar o controle de ponto e ter acesso, visualizar e registrar os seus horários, clique em “Controle de Ponto”.

2.4- Para acessar os exames médicos e ter acesso aos seus atestados, saber seu dia de retorno, mandar atestados e se informar sobre o exame periódico da empresa, clique em “Exames”.

2.5- Para acessar seu contrato de trabalho e poder ler o que consta em seu documento, clique em “Contrato de Trabalho”.

2.6- Para acessar férias e saber sobre seu período de aquisição de férias, retorno e pedir alguma alteração e informação sobre, clique em “Férias”.

2.7- Para acessar ajuda e saber mais sobre as funcionalidades, algum tipo de erro, alguma dúvida ou problema, tutoriais, FAQs (perguntas frequentes) ou querer contatar alguém do SAC (Serviço de Atendimento ao Consumidor) sobre o uso do site, clique em “Ajuda”.

Na página do Perfil do Usuário, onde é possível alterar sua foto de apresentação e é exibido as principais informações do usuário como: seu departamento, a que gestor ou superior responde se tiver, cargo ocupado na empresa, de onde é, contato, rede social profissional entre outras. Essa sessão também dá acesso a visualizações de seus benefícios mostrando o tipo de benefício, quando será o dia creditado, o valor recebido e seus descontos.

3- Para alteração de foto de seu perfil, clicar em cima de sua foto e ou do espaço indicado de foto, e ele dará opção de adicionar sua foto direcionando para a pasta de fotos e arquivos de seu dispositivo. Será aceito foto no formato PNG ou JPEG.

Em Folha de Pagamento, é possível gerar seu holerite e informe de rendimentos.

4- Para gerar o informe de rendimentos do período trabalhado na empresa, clique na seta do campo indicado e depois clique no ano desejado.

4.1- Para gerar o holerite do mês que deseja, clique na seta do campo indicado e depois clique no mês interessado.

Folha de Ponto, é onde é exibido, feito alterações e o registro de ponto do funcionário.

5- Para solicitar correção de ponto, selecione o dia desejado e informe o erro, correção ou verificação do mesmo.

5.1- Para gerar a visualização de ponto em formato PDF, clique na seta do campo indicado e depois clique no mês interessado.

Em Contrato de Trabalho, é onde é localizado a documentação de contrato que o funcionário é ativo.

6- Para ver o documento, clique em “Documento de Contrato de Trabalho” e será aberto um arquivo em formato PDF que possibilita download.

Em Férias, é evidenciado na tela o mês de data de início e retorno em que o funcionário irá usufruir de suas férias e gera a documentação.

7- Para solicitar mudanças ou outras questões relacionadas as férias, usar o campo “solicitação de mudanças ou dúvidas”.

7.1- Para gerar recibo, clique “Gerar recibo de férias”.

A página Ajuda, é em qual local o usuário pode tirar dúvidas sobre as funcionalidades, erros e outras questionamentos sobre o sistema.

8- Selecione o tópico de dúvida desejado para ler a solução e o direcionamento de resposta.

## 9.0 DESENVOLVIMENTO DO PROJETO:

Após a reunião com o cliente, foi feito um planejamento com todas as etapas a serem realizadas para o desenvolvimento do projeto.

TAREFAS	JANEIRO	FEVEREIRO	MARÇO	ABRIL	MAIO	JUNHO	JULHO	AGOSTO	SETEMBRO	OUTUBRO	NOVEMBRO	DEZEMBRO
REUNIÃO	x											
LEVANTAMENTO DE REQUISITOS	x	x										
ESCOLHA DA EQUIPE		x	x									
TREINAMENTO DA EQUIPE			x									
FORMAÇÃO DOS GRUPOS			x	x								
INICIO DO DESENVOLVIMENTO				x	x	x						
CONTROLE DAS ETAPAS						x						
TESTE DE PROTÓTIPO						x	x					
CONTROLE DE MELHORIAS							x	x				
DESENVOLVIMENTO								x	x	x		
II TESTE DE PROTÓTIPO										x	x	
CONTROLE DE MELHORIAS											x	x
DESENVOLVIMENTO FINAL												x
TESTE DE USABILIDADE (INTERNO)												x
TESTE DE USABILIDADE USUÁRIO												x

Figura 6: *planner do projeto*  
Fonte: os autores.

## **9.1 Reunião**

Na reunião com os clientes, foram levantados os requisitos funcionais e não funcionais para o desenvolvimento do software e foi criado um documento com os requisitos.

### **9.1.1 Levantamento de requisitos funcionais para o desenvolvimento do sistema:**

1. Cadastro de funcionários;
2. Dados pessoais dos funcionários. Exemplos: nome, endereço, data de nascimento;
3. Registro de admissão de funcionários;
4. Registro de demissão dos funcionários;
5. Registro de férias dos funcionários;
6. Registro de afastamentos. Exemplos: licenças médicas, licença maternidade, licença paternidade, licença por luto;
7. Cálculo de salário bruto;
8. Cálculo de horas trabalhadas;
9. Controle de jornada de trabalho;
10. Registro de horas extras;
11. Registro de benefícios. Exemplos: vale-transporte, vale-refeição, PLR;
12. Dedução de impostos e contribuições obrigatórias;
13. Cálculo de INSS (Instituto Nacional do Seguro Social);
14. Cálculo de FGTS (Fundo de Garantia do Tempo de Serviço);
15. Cálculo de IRRF (Imposto de Renda Retido na Fonte);
16. Cálculo de PIS (Programa de Integração Social);
17. Cálculo de Cofins (Contribuição para o Financiamento da Seguridade Social);
18. Cálculo de salário líquido;
19. Controle de descontos e deduções da folha de pagamento;
20. Controle de contribuições sindicais;
21. Registro de ausência. Exemplos: faltas, atrasos, horas não trabalhadas;
22. Controle de banco de horas;
23. Pagamento de banco de horas (3 meses o tempo máximo para efetuar o pagamento das horas trabalhadas ou folga ao funcionário)

24. Geração de recibos de pagamento;
25. Emissão de holerites;
26. Controle de vale-transporte (desconto de 6%);
27. Controle de vale-refeição/alimentação;
28. Cálculo de horas de trabalho após ou anterior ao período trabalhado hora extra;
29. Controle de aviso prévio;
30. Cálculo do décimo terceiro salário;
31. Cálculo de férias (incluindo o terço constitucional);
32. Controle de licença maternidade/paternidade;
33. Controle de licença médica;
34. Cálculo de horas de trabalho em home office;
35. Cálculo de horas de trabalho em jornadas reduzidas;
36. Cálculo de horas de trabalho em contrato de experiência;
37. Geração de relatórios para órgãos governamentais.

#### **9.1.2 Lista de requisitos não-funcionais:**

1. Precisão: A folha de pagamento deve calcular com precisão os cumprimentos, benefícios e deduções dos funcionários.
2. Segurança: O sistema deve garantir a segurança dos dados pessoais e financeiros dos funcionários.
3. Conformidade legal: O software deve cumprir todas as leis e regulamentações fiscais e trabalhistas.
4. Escalabilidade: O sistema deve ser capaz de lidar com o aumento do número de funcionários sem comprometer o desempenho.
5. Flexibilidade: Deve ser possível configurar diferentes tipos de declarações, benefícios e deduções para atender às necessidades específicas da empresa.
6. Integração: O software deve se integrar facilmente com outros sistemas, como o sistema de recursos humanos e o sistema de contabilidade.
7. Acessibilidade: O sistema deve ser acessível a partir de diferentes dispositivos, como computadores, tablets e smartphones.
8. Usabilidade: A interface do usuário deve ser intuitiva e fácil de usar, mesmo para usuários não técnicos.
9. Disponibilidade: O sistema deve estar disponível para uso durante o período necessário, com o mínimo de tempo de inatividade.



10. Desempenho: O software deve ter um tempo de resposta rápido para processar grandes volumes de dados de pagamento.

11. Confiabilidade: O sistema deve ser confiável, evitando erros e garantindo que os dados não sejam perdidos.

12. Auditoria: O software deve permitir a rastreabilidade das operações realizadas, registrando informações de alterações e acessos.

13. Gerenciamento de características: O sistema deve lidar com situações instrutivas, como painéis variáveis ou controles específicos.

14. Customização: Deve ser possível personalizar relatórios e documentos de pagamento de acordo com as necessidades da empresa.

15. Controle de acesso: O sistema deve ter diferentes níveis de acesso para garantir que apenas pessoas autorizadas possam fazer alterações nos dados de pagamento.

16. Backup e recuperação: O software deve realizar backups regularmente e permitir a recuperação dos dados em caso de falhas ou desastres.

17. Controle de versão: O sistema deve manter o controle de versões anteriores dos dados de pagamento para fins de histórico e auditorias.

18. Multilíngue: O software deve suportar diferentes idiomas para atender às necessidades de empresas com funcionários internacionais.

19. Suporte técnico: Deve haver suporte técnico disponível para resolver problemas e fornecer orientação aos usuários.

20. Notificações automáticas: O sistema deve enviar notificações automáticas aos funcionários sobre pagamentos, alterações ou alterações salariais.

21. Cálculo de férias: O software deve calcular e registrar com precisão os pagamentos de férias dos funcionários.

22. Controle de horas extras: O sistema deve registrar e calcular as horas extras trabalhadas pelos funcionários.

23. Cálculo de impostos: O software deve calcular corretamente os impostos a serem retidos dos trabalhadores dos funcionários.

24. Gestão de benefícios: O sistema deve permitir a administração de benefícios, como plano de saúde e vale-refeição.

25. Automatização de tarefas: O software deve automatizar tarefas repetitivas, atendendo a carga de trabalho administrativo.

26. Relatórios personalizados: O sistema deve gerar relatórios personalizados com informações relevantes sobre a folha de pagamento.

27. Contabilização: O software deve gerar informações contábeis precisas e

fornecer integração com sistemas contábeis.

28. Acesso remoto: O sistema deve permitir o acesso remoto para processamento da folha de pagamento em locais diferentes.

29. Suporte a diferentes tipos de contratos: O software deve suportar diferentes tipos de contratos de trabalho, como tempo integral, meio período, temporário e etc.

30. Cálculo de encargos trabalhistas: O sistema deve calcular corretamente os encargos trabalhistas, como INSS e FGTS.

31. Controle de subsídios: O software deve controlar as concessões concedidas aos funcionários e calcular as deduções corretas.

32. Integração com instituições financeiras: O sistema deve se integrar com bancos e instituições financeiras para facilitar o pagamento dos internamentos.

33. Gestão de recibos: O software deve gerar recibos de pagamento precisos e mantê-los disponíveis para consulta.

34. Portal do funcionário: Deve haver um portal do funcionário onde os funcionários possam acessar seus contracheques e informações pessoais.

35. Controle de faltas e atrasos: O sistema deve registrar e calcular faltas e atrasos dos funcionários para aplicar as devidas deduções.

36. Controle de benefícios flexíveis: O software deve permitir a administração de benefícios flexíveis, onde os funcionários podem escolher entre diferentes opções.

37. Gestão de afastamentos: O sistema deve lidar com afastamentos temporários, como licença médica, licença maternidade, entre outros.

38. Regras de negócio personalizáveis: O software deve permitir a configuração de regras de negócio específicas da empresa.

39. Histórico de pagamentos: O sistema deve manter um histórico completo de pagamentos para referência futura.

40. Notas fiscais eletrônicas: O software deve gerar notas fiscais eletrônicas para os pagamentos.

41. Autenticação e autorização: O sistema deve garantir o login seguro dos usuários e conceder autorizações adequadas com base em suas funções.

42. Controle de benefícios variáveis: O software deve permitir a administração de benefícios variáveis, como bônus e comissões.

43. Cálculo de rescisões: O sistema deve calcular corretamente os pagamentos de rescisões contratuais.

44. Suporte a múltiplas empresas: O software deve suportar o processamento da folha de pagamento para várias empresas, se necessário.

45. Gestão de adiantamentos: O sistema deve permitir a solicitação e o controle

de adiantamentos salariais pelos funcionários.

46. Controle de saldo de férias: O software deve acompanhar e calcular o saldo de férias dos funcionários.

47. Monitoramento de prazos: O sistema deve fornecer insumos e alertas para garantir o cumprimento dos prazos relacionados à folha de pagamento.

48. Backup e arquivamento de documentos: O sistema deve permitir o armazenamento seguro e organizado de documentos relacionados à folha de pagamento, como contratos e declarações.

49. Controle de pagamento de terceiros: O software deve permitir o controle e o pagamento de fornecedores ou retornos de serviço vinculados à folha de pagamento.

50. Atualização regulatória: O sistema deve ser atualizado regularmente para se adequar às mudanças nas leis e regulamentações trabalhistas e fiscais.

### 9.2.3 Casos de uso

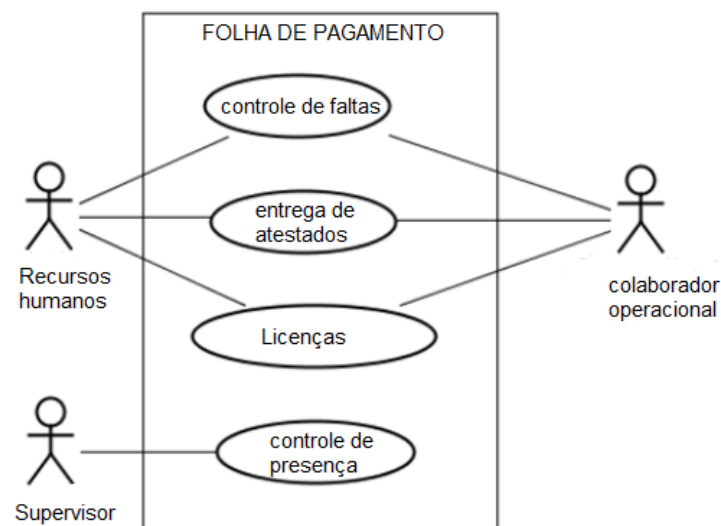


Figura 7: caso de uso 1

Fonte: os autores

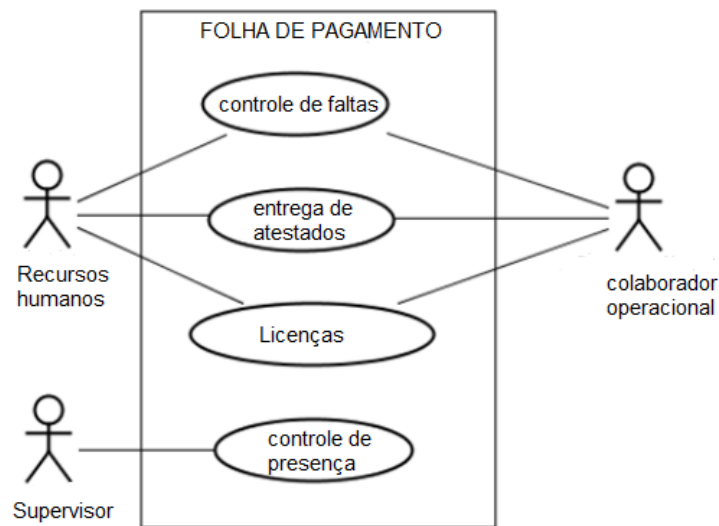


Figura 8: caso de uso 2

Fonte: os autores

### 9.3 ESCOLHA DA EQUIPE

Seguindo o cronograma, foi feita uma escolha da equipe de acordo com as soft skills e hard skills de cada colaborador com o intuito de ao montar a equipe, cada colaborador desenvolva com suas habilidades em cada etapa.

#### 9.3.1 Treinamento da equipe

Após a escolha da equipe, foi realizada uma reunião para apresentar o projeto aos colaboradores que irão atuar nesse projeto. Com isso, foram designados treinamentos de capacitação e aprimoramento para que a equipe esteja devidamente capacitada ao desenvolver o projeto.

### 9.4 ÍNICIO DO DESENVOLVIMENTO DO PROJETO

Houve uma reunião com a equipe para iniciar o projeto para verificar quais sistemas irão ser utilizados para o desenvolvimento do sistema.

### 9.4.1 Escolha da linguagem

Para a programação, foi escolhida a linguagem em C# para desenvolver os sistemas em *web* e *desktop*, pois o C# é uma linguagem orientada a objetos, o que significa que suporta os conceitos de encapsulamento, herança e polimorfismo. Isso permite uma estruturação clara e modular do código, facilitando a reutilização e a manutenção. E de acordo com os requisitos, esse sistema pode sofrer alterações constantes de acordo com as leis trabalhistas e alteração de dados.



Figura 9: programação orientada a objeto

Fonte: <https://www.ateomomento.com.br/orientacao-a-objetos-e-semantica/>

### 9.4.2 Escolha do sistema para programar a interface do usuário:

O sistema escolhido para o desenvolvimento da Interface é o *Figma*. Uma das principais vantagens do *Figma* é sua capacidade de permitir a colaboração em tempo real. Várias pessoas podem trabalhar simultaneamente no mesmo projeto, visualizando as atualizações em tempo real. Isso facilita o trabalho em equipe, a revisão e o feedback entre os membros do time.

#### 9.4.2.1 Passo a passo das telas

Em nossa tela inicial de software nota-se heurísticas seguidas como o reconhecimento, na ordem básica de primeiramente ao tentar realizar um *login* ao sistema e caso ocorra o esquecimento da própria senha temos o guiamento de ajuda, onde se encontra “Esqueceu sua senha?”.



Figura 10: login

Fonte: os autores.

Após o Login, temos o menu principal onde já aparece a barra de Menu que ficará fixa como atalho para todo o sistema, seguindo uma consistência de localização e assimilação para o usuário. Avisos e mensagens também podem ser colocados da maneira que for necessário.

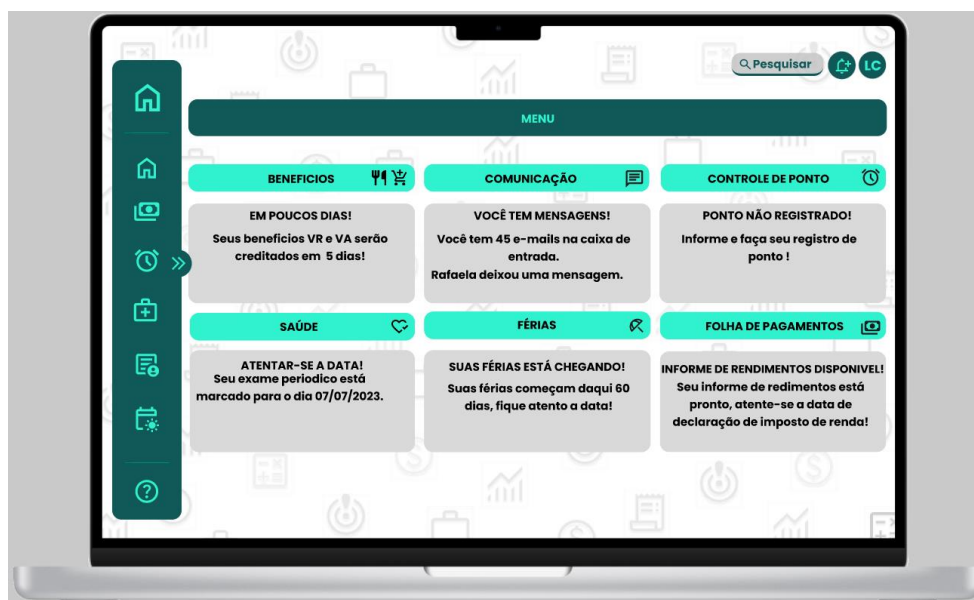


Figura 11: interface do usuário

Fonte: os autores.

No atalho em cima da “Home” representado pela “Casinha” encontramos a página de Perfil do usuário, onde se encontra informações mais completas sobre ele mesmo com uma estética minimalista e precisa.

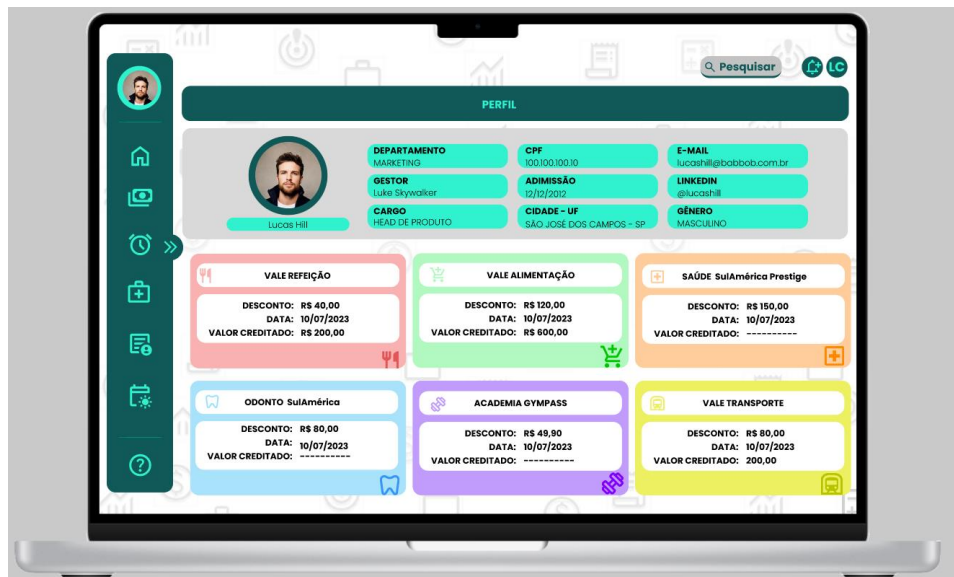


Figura 12: perfil do usuário

Fonte: os autores.

Uma das funcionalidades mais importantes é a Folha de Pagamento, nela o usuário pode gerar seu holerite do mês desejado e seus informes de rendimentos de cada ano trabalhado na empresa.

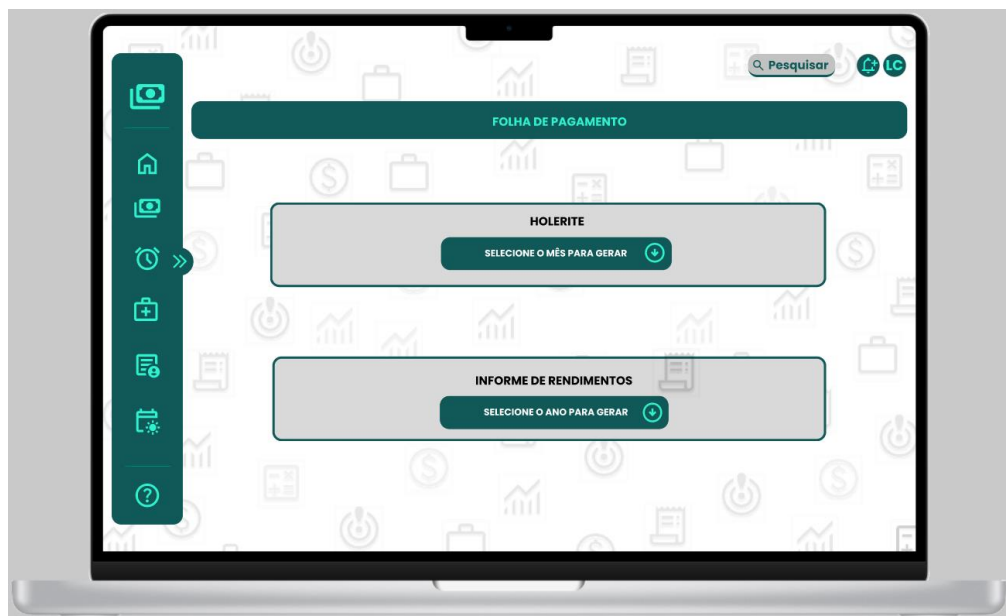


Figura 13: folha de pagamento

Fonte: os autores.

### 9.4.2.2 Teste de usabilidade do sistema

Durante o andamento do processo, foram realizados controles de andamento do projeto para identificar possíveis erros no desenvolvimento do projeto. Após esse controle, foram realizados testes com as equipes internas para verificar a usabilidade do sistema antes de chegar ao usuário final.

### 9.4.3 Desenvolvimento do banco de dados do sistema

Durante o desenvolvimento do software, uma das principais partes foi a modelagem do banco de dados, que foi feita da seguinte forma.

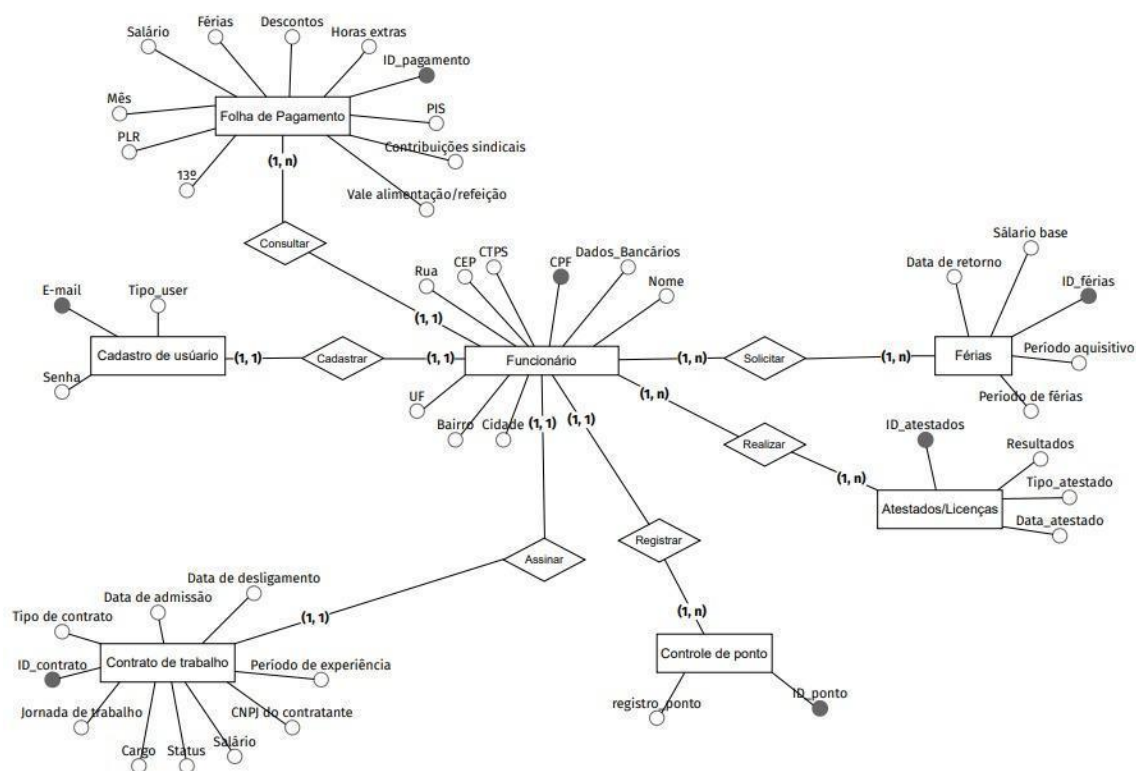


Figura 14: banco de dados para gerar folha de pagamento.

Fonte: os autores.



#### **9.4.3.1 Atributos da Folha de pagamento**

- Descontos: que são tudo o que for ter de desconto do salário do funcionário.  
EX: INSS, IRRF, Convênio médico, etc;
- ID\_pagamento: Código identificador do pagamento;
- PIS: Atributo referente ao programa de integração social.
- Contribuições sindicais: valor opcional que o funcionário poderá enviar como contribuição ao sindicato;
- Vale alimentação/refeição: benefício para o colaborador;
- Décimo terceiro: benefício garantido por lei referente ao um salário extra pago no final do ano;
- Horas extras: caso o funcionário trabalhe além de sua jornada de trabalho, será acrescido um valor referente a esse tempo a mais;
- Salário: será o valor acordado, que será pago pela jornada de horas trabalhadas;
- Mês: nesse caso, será um campo para que o usuário saiba o mês de referência da folha que ele está visualizando;
- PLR: nesse atributo vai ser usado para representar a Participação nos Lucros e Resultados, recebida pelo colaborador na sua folha de pagamento;
- Férias: atributo referente ao pagamento das férias do funcionário.

#### **9.4.3.2 Atributos da tabela férias**

- ID\_férias: Código identificador das férias;
- Período férias: é o tempo em que o funcionário ficou de férias;
- Período aquisitivo: é referente ao período em que o funcionário trabalhou na empresa e que ele vai usar para tirar suas férias;
- Salário base: representa o salário que vai ser utilizado para o cálculo das férias.
- Data retorno: representa a data em que o funcionário vai retornar para a empresa.

#### **9.4.3.3 Atributos na tabela de controle de ponto**

- ID\_Ponto: Código identificador do ponto.
- Registros ponto: horário que o funcionário a marcação do ponto.

#### **9.4.3.4 Atributos na tabela de cadastro de usuário**

- Tipo\_user: limita o acesso de certas funções do sistemas dependendo do funcionário.
- Senha: chave de segurança utilizada para que o usuário possa fazer login no sistema.
- E-mail: endereço de e-mail do usuário cadastrado pelo RH que será utilizado para que o usuário possa efetuar o login.

#### 9.4.3.5 Atributos de tabela de contrato

- Id\_contrato: código identificador do contrato.
- Período de experiência: referente aos 90 dias de experiência do funcionário.
- Data de desligamento: data em que o funcionário deixou a empresa.
- Jornada de trabalho: representará a carga horária que o funcionário deverá cumprir;
- Data de admissão: esse atributo será a data que o usuário foi contratado;
- Tipo de contrato: tem o intuito de especificar o tipo de contrato do colaborador, pode ser temporário, efetivo etc.
  - Cargo: nesse atributo, iremos representar a função estabelecida pelo colaborador;
  - Salário: será registrado qual a remuneração que o usuário terá, valor bruto;
  - Status: esse atributo é bem importante, pois nele temos o registro de que funcionário está ativo (atuando), ou se ele foi desligado;
  - CNPJ do contratante: corresponde ao CNPJ da empresa que contratará os serviços do colaborador.

#### 9.4.3.6 Atributos na tabela funcionário

- Dados bancários: conta bancária em que será realizado o pagamento do funcionário;
- Nome: nesse atributo será armazenado o nome completo do funcionário;
- CPF: documento de identificação do funcionário;
- CEP: código postal do funcionário;
- CTPS: carteira de trabalho e previdência social;
- Rua: rua em que o funcionário reside;
- UF: estado em que o funcionário reside;
- Bairro: bairro em que o funcionário reside;
- Cidade: cidade em que o funcionário reside.

### 9.5 SCRIPT DE BANCO DE DADOS

```
CREATE TABLE Funcionário
(
    nome INT,
    CPF INT PRIMARY KEY,
    idCadastro de usuário INT,
);
```

```
CREATE TABLE Férias
(
    Período aquisitivo INT,
    id_ferias INT PRIMARY KEY,
    Data de retorno INT,
);
```

```
CREATE TABLE Controle de ponto
(
    registro_ponto INT,
    ID_ponto INT PRIMARY KEY,
    idFuncionário INT,
);
```

```
CREATE TABLE Exames médicos
(
  data_exame INT,
  tipo_exame INT,
  resultado_exame INT,
  ID_exame INT PRIMARY KEY,
);
```

```
CREATE TABLE Contrato de trabalho
(
  CNPJ do contratante INT,
  Salário INT,
  Cargo INT,
  Tipo de contrato INT,
  Jornada de trabalho INT,
  Data de admissão INT,
  Status INT,
  ID_contrato INT PRIMARY KEY,
  idFuncionário INT,
);
```

```
CREATE TABLE Folha de Pagamento
(
  Horas extras INT,
  Descontos INT,
  PLR INT,
  Férias INT,
  Mês INT,
  ID_pagamento INT PRIMARY KEY,
  Salário INT,
  idFuncionário INT,
);
```

```
CREATE TABLE Cadastro de usuário
(
  Tipo_user INT,
  Senha INT,
  E-mail INT PRIMARY KEY,
);
```

```
CREATE TABLE Solicitar
(
  CPF INT PRIMARY KEY,
  id_ferías INT PRIMARY KEY,
);
```

```
CREATE TABLE Realizar
(
  CPF INT PRIMARY KEY,
  ID_exame INT PRIMARY KEY,
);
```

```
ALTER TABLE Funcionário ADD FOREIGN KEY(idCadastro de usuário) REFERENCES
Cadastro de usuário (idCadastro de usuário)
```

```
ALTER TABLE Controle de ponto ADD FOREIGN KEY(idFuncionário) REFERENCES
Funcionário (idFuncionário)
```

```
ALTER TABLE Contrato de trabalho ADD FOREIGN KEY(idFuncionário) REFERENCES
Funcionário (idFuncionário)
```

ALTER TABLE Folha de Pagamento ADD FOREIGN KEY(idFuncionário) REFERENCES Funcionário (idFuncionário)  
 ALTER TABLE Solicitar ADD FOREIGN KEY(CPF) REFERENCES Funcionário (CPF)  
 ALTER TABLE Solicitar ADD FOREIGN KEY(id\_ferias) REFERENCES Férias (id\_ferias)  
 ALTER TABLE Realizar ADD FOREIGN KEY(CPF) REFERENCES Funcionário (CPF)  
 ALTER TABLE Realizar ADD FOREIGN KEY(ID\_exame) REFERENCES Exames médicos (ID\_exame)

## 9.6 DICIONÁRIO DE DADOS

Tabela	Folha de pagamento			
Descrição	Armazera os dados financeiros dos funcionário.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio
ID_pagamento	Código para identificar a folha.	Int	1000	PK/NOT NULL
PLR	Representa a participações no lucros e resultados do funcionário.	Decimal	6	NULL
Mês	É o mês de referência da folha visualizada.	Varchar	9	NOT NULL
Salário	Valor pago pela jornada de horas trabalhadas.	Decimal	6	NOT NULL
Férias	Pagamento de férias do funcionário.	Decimal	6	NULL
Descontos	Descontos do salário do funcionário.	Decimal	5	NOT NULL
Contribuições sindicais	Contribuição opcional para o sindicato da região.	Decimal	4	NULL
13º	Pagamento definido por lei, como um salário adicional no final do ano.	Decimal	6	NULL
Vale alimentação/refeição	Benefício oferecido pela empresa.	Decimal	6	NOT NULL
PIS	Atributo referente ao program de integração social.	Decimal	6	NOT NULL
Horas extras	Valor refente ao trabalho além da sua jornada.	Decimal	5	NULL

Tabela	Cadastro de usuário			
Descrição	Armazera os dados de login dos usuários.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio
tipo_user	Representa o cargo do funcionário.	Varchar	20	NOT NULL
E-mail	E-mail corporativo utilizado para acessar o sistema.	Varchar	40	PK/NOT NULL
Senha	Chave de securanca para acesso ao sistema.	Varchar	8	NOT NULL

Tabela	Contrato de trabalho			
Descrição	Armazera as informações e documentos do funcionário.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio
ID_contrato	Código para identificar o contrato.	Int	1000	PK/NOT NULL
Período de experiência	Período definido pela empresa, para que a empresa defina a contratação	Varchar	10	NOT NULL
Data de admissão	Data que o colaborador entrou na companhia.	Varchar	10	NOT NULL
Data de desligamento	Data que o colaborador saiu da companhia.	Varchar	10	NULL
Tipo de contrato	Para saber se o colaborador é temporário ou efetivo.	Varchar	10	NOT NULL
Jornada de trabalho	Tempo de trabalho que o colaborador deverá cumprir.	Varchar	5	NOT NULL
Cargo	Representa o cargo do funcionário.	Varchar	20	NOT NULL
Status	Representa a situação atual do colaborador.	Varchar	7	NOT NULL
Salário	Valor bruto que será pago mensalmente.	Varchar	6	NOT NULL
CNPJ do contratante	CNPJ da empresa que está contratando os serviços do colaborador.	Varchar	14	NOT NULL

Tabela	Controle de ponto			
Descrição	Armazera o controle de jornada do funcionário.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio
ID_ponto	Código para identificar o ponto do colaborador	Int	1000	PK/NOT NULL
Registro_ponto	Horário que o funcionário deverá estar atuando na empresa	Date	10	NULL

Tabela	Atestados/Licenças			
Descrição	Armazera as informações de exames médicos do funcionário.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio
ID_atestados	Código para identificar o registro do atestado.	Int	1000	PK/NOT NULL
Tipo_atestado	Qual o procedimento realizado pelo funcionário.	Varchar	50	NOT NULL
Resultados	Como ficou o resultado do procedimento.	Varchar	20	NOT NULL
Data_atestado	Data que o usuário realizou o procedimento.	Varchar	10	NOT NULL

Tabela	Férias			
Descrição	Armazera as informações de férias do colaborador.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio
ID_férias	Código para registrar as férias.	Int	1000	PK/NOT NULL
Período_aquisitivo	Período disponível como saldo de férias.	Varchar	50	NOT NULL
Período_férias	Período que o colaborador adquiriu de férias.	Varchar	20	NOT NULL
Salário_base	Salário bruto do colaborador usado no cálculo das férias.	Varchar	6	NOT NULL
Data_retorno	Data que o colaborador retornou de férias.	Varchar	10	NOT NULL

Tabela	Funcionário			
Descrição	Conterá dados sobre o funcionário.			
Observações				
Campos				
Nome	Descrição	Tipo de dado	Tamanho	Restrições de domínio
CPF	Documento de identificação.	Varchar	11	PK/NOT NULL
Nome	Nome do colaborador.	Varchar	60	NOT NULL
Cidade	Cidade de residência.	Varchar	40	NOT NULL
UF	Estado de residência.	Varchar	2	NOT NULL
Bairro	Bairro de residência.	Varchar	45	NOT NULL
Rua	Rua de residência.	Varchar	50	NOT NULL
CEP	CEP de residência.	Varchar	8	NOT NULL
CTPS	Carteira de trabalho e Previdência Social do funcionário.	Varchar	20	NOT NULL
Dados Bancários	Dados bancários para pagamento do salário.	Varchar	20	NOT NULL

## **10. CONCLUSÃO:**

O projeto consistiu no desenvolvimento de um software para o RH de uma empresa. Inicialmente, definimos os requisitos do projeto, estabelecendo um ponto de partida. Com base nesses requisitos, elaboramos os diagramas de caso de uso e o diagrama de sequência. Com os diagramas prontos, criamos o diagrama de classes. Após essa etapa, iniciamos o trabalho na interface utilizando o Figma, mantendo sempre em mente os requisitos levantados. Na nossa interface, aplicamos os conceitos das Heurísticas de Nielsen com o objetivo de entregar uma interface intuitiva e de fácil compreensão para o usuário, garantindo assim uma experiência consistente. Dessa forma, alcançamos um protótipo de alta fidelidade. Um dos pontos que sempre consideramos foi a LGPD, uma vez que um software de RH lida com dados sensíveis e sigilosos de várias pessoas.

A pesquisa e o estudo em economia e mercado nos ajudaram a compreender o funcionamento e as práticas adotadas pelas empresas em relação à administração de recursos humanos e folha de pagamento.

A Gestão Estratégica de Recursos Humanos possibilitou a compreensão dos processos internos dos departamentos da empresa e das demandas dos funcionários em relação ao sistema de folha de pagamento.

Após todas as pesquisas feitas, requisitos levantados e desenvolvimento conseguimos concluir o projeto conforme o planejamento inicial.

## 11. REFERÊNCIAS BIBLIOGRÁFICAS:

Schulz, Wolfgang. Direitos Humanos e Encriptação. Série de livros da Unesco sobre liberdade da internet . Paris, França. Editora UNESCO. Disponível em: <https://unesdoc.unesco.org/ark:/48223/pf0000380587?posInSet=4&queryId=9a25c6c5-cfb6-4973-85c2-7d12fcc888c1>. Acesso em: 30 mai. 2023.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais. Diário Oficial da União, Brasília, DF, 15 ago. 2018. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/l13709.html](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.html). Acesso em: 31 mai. 2023.

Pfleeger S. L. Software Engineering: theory and practice. Prentice-Hall. 1998.

Pressman R. Engenharia de Software, 6a edição, AMGH Editora. 2010.

Rezende D. A. Engenharia de software e sistemas de informação, 2a edição. Brasport, 2002.

Rocha A. R. C., Maldonado J. C., Weber K. C. Qualidade de Software: Teoria e Prática, Prentice Hall, 2001. Sommerville, I.

Engenharia de Software. 8a Edição. Addison Wesley. 2007.

Pinhata, Gabriela. "10 Heurísticas de Nielsen: melhorando a usabilidade". Disponível em: <https://www.programmers.com.br/blog/10-heuristicas-de-nielsen-melhorando-a-usabilidade/>

Norman, Donald. O Design do Dia a Dia. Rio de Janeiro: Editora Rocco, 2006.

Faria, Flaubi. "10 motivos para você usar um site responsivo e suas principais vantagens". Disponível em: <https://resultadosdigitais.com.br/marketing/site-responsivo/#:~:text=Responsivo%20%C3%A9%20todo%20o%20site,em%20determinado s%20tipos%20de%20dispositivos>

Turban, E., Volonino, L. (2019). Information Technology for Management: Digital Strategies for Insight, Action, and Sustainable Performance. 11th edition. Wiley.

O'Brien, J.A., Marakas, G.M. (2018). Administração de Sistemas de Informação. 18ª edição. McGraw-Hill.

Laudon, K.C., Laudon, J.P. (2016). Sistemas de Informação Gerenciais. 14ª edição. Pearson. (editado)

Osterwalder, Alexander e Pigneur, Yves. Business Model Generation - Inovação em Modelos de Negócios, Rio de Janeiro: Alta Books Editora, 2011.

<https://www.canva.com>

Sobrecarga e sobreposição de métodos em orientação a objetos (devmedia.com.br)

CAMPOS, M, L. Modelização de domínios de conhecimento: uma investigação de princípios fundamentais. Revista ciência da informação versão v.33 n.1. Brasília 2004.

RICARTE, I,L. Programação Orientada a Objetos: Uma Abordagem com Java. Faculdade de engenharia elétrica e de computação UNICAMP. Campinas 2001.

CARDELLI, L.; WEGNER, P. On Understanding Types, Data Abstraction, and Polymorphism. ACM Computing Surveys (CSUR). vol.17, pag. 471 – 524. 1985.

CARVALHO, V; TEXEIRA, G. Programação orientada a objetos: Curso técnico de informática IFES. Colatina 2012.