



ICET - INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

PROJETO INTEGRADO MULTIDISCIPLINAR
PIM IV

Desenvolvimento de um sistema para automatização total da folha de pagamento de uma empresa.

Nome

ALLAN PEREIRA LOPES
GIOVANNA CHIKUINI PRIMON
JOÃO PEDRO GONÇALVES MANARA
MATHEUS FELIX BARBOSA

R.A

N84333-0
N9716G-2
N037FF6
G50511-5

SÃO JOSÉ DOS CAMPOS – SP

NOVEMBRO/2023

NOME	RA
ALLAN PEREIRA LOPES	N84333-0
GIOVANNA CHIKUINI PRIMON	N9716G-2
JOÃO PEDRO GONÇALVES MANARA	N037FF-6
MATHEUS FELIX BARBOSA	G50511-5

Desenvolvimento 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

NOVEMBRO/2023

RESUMO

O projeto abordado teve como objetivo analisar e desenvolver um sistema de folha de pagamento eficiente e preciso para uma determinada organização, visando eliminar a complexidade e o tempo gasto no cálculo de salários, deduções de impostos, benefícios e outras variáveis. O estudo se baseou em uma revisão resumida dos conceitos e práticas relacionados à administração de folha de pagamento, bem como nas melhores práticas de gestão de recursos humanos. Também foram realizados estudos de caso, pesquisas e levantamentos de dados para identificar as necessidades específicas da organização em questão. O trabalho visou fornecer uma compreensão aprofundada dos desafios e benefícios da implementação de um sistema de folha de pagamento adequado e apresentou uma solução viável e eficaz, considerando as características e demandas do departamento estudado, visando otimizar recursos e garantir o cumprimento com as legislações trabalhistas e tributárias vigentes. O esperado foi que o desenvolvimento desse sistema pudesse contribuir para o aprimoramento dos processos de gestão de recursos humanos, oferecendo uma proposta que atendesse às necessidades específicas da organização e promovesse a eficiência operacional e a satisfação dos funcionários. O resultado deste trabalho incluiu a apresentação de um sistema de folha de pagamento implementado e testado, capaz de auxiliar a organização no processamento correto e eficiente dos pagamentos aos funcionários, simplificando tarefas administrativas e garantindo o cumprimento das normas legais cumpridas.

Palavras-Chave: Processamento de pagamento, Automatização, Folha de pagamento

SUMÁRIO

	Pág.
1. INTRODUÇÃO	5
2. PROGRAMAÇÃO ORIENTADA A OBJETOS II	7
3. DESENVOLVIMENTO DE SOFTWARE PARA INTERNET	8
4. TÓPICOS ESPECIAIS DE PROGRAMAÇÃO ORIENTADA A OBJETOS	9
5. PROJETO DE SISTEMAS ORIENTADO A OBJETOS	10
6. GERENCIAMENTO DE PROJETO DE SOFTWARE	12
7. EMPREENDEDORISMO	14
8. GESTÃO DA QUALIDADE	15
9. DESENVOLVIMENTO DO PROJETO	16
10. CONCLUSÃO	30
11. GLOSSÁRIO	31
12. REFERÊNCIAS	33
ANEXO A - SCRIPTS DE CRIAÇÃO DE TABELAS (1)	35
ANEXO B - SCRIPTS DE CRIAÇÃO DE TABELAS (2)	36
ANEXO C - SCRIPTS DE CRIAÇÃO DE TABELAS (3)	37
ANEXO D - SCRIPTS DE INSERÇÃO DE DADOS	38

1. INTRODUÇÃO

O presente trabalho tem como objetivo abordar o desenvolvimento de um sistema de folha de pagamento e sua importância estratégica no contexto organizacional. A administração eficiente da folha de pagamento é uma das principais responsabilidades do setor de recursos humanos de uma empresa, e um sistema organizado pode desempenhar um papel crucial nesse processo.

No atual cenário empresarial, em que as organizações buscam maximizar sua eficiência, reduzir custos e garantir o cumprimento das obrigações legais, a implementação de um sistema de folha de pagamento confiável e preciso torna-se essencial. Um sistema adequado facilita o cálculo dos rendimentos, benefícios, impostos e outras deduções, além de fornecer relatórios gerenciais importantes para auxiliar nas tomadas de decisão.

Além disso, a gestão estratégica de recursos humanos está diretamente relacionada ao sistema de folha de pagamento. A remuneração adequada dos colaboradores é um fator determinante para a atração, motivação e retenção de talentos. Um sistema eficiente e transparente contribui para a equidade salarial e a satisfação dos funcionários.

Neste trabalho, serão explorados os principais aspectos relacionados ao desenvolvimento de um sistema de folha de pagamento, desde a análise de requisitos até a implementação e manutenção, além disso serão abordadas as melhores práticas, tecnologias e ferramentas disponíveis para garantir a eficiência e eficácia desse sistema.

Por fim, espera-se que este estudo contribua para o entendimento da importância estratégica do sistema de folha de pagamento, fornecendo diretrizes e insights relevantes para a implementação bem-sucedida desse sistema em organizações de diferentes portes e setores.

OBJETIVO GERAL

Desenvolver um sistema de automatização de folha de pagamento que integre todos os processos do setor de Recursos Humanos e Departamento Pessoal, com foco na conformidade com a Lei Geral de Proteção de Dados (LGPD), aprimorando a eficiência e a precisão do cálculo da folha de pagamento, bem como a geração digital de documentos relacionados, com o objetivo de automatizar completamente a produção da folha de pagamento.

Objetivos Específicos

- Realizar uma revisão detalhada do trabalho de levantamento e análise de requisitos, identificando possíveis melhorias e ajustes necessários para o desenvolvimento do sistema;
- Desenvolver um sistema de software que seja capaz de capturar, armazenar e processar as informações relacionadas à folha de pagamento, levando em consideração as particularidades e regulamentações da LGPD para proteger os dados pessoais;
- Desenvolver uma interface de usuário intuitiva e amigável para que os funcionários do RH e do Departamento Pessoal possam interagir facilmente com o sistema, realizar verificações e validações necessárias e extrair relatórios de dados relevantes;
- Implementar medidas de segurança robustas para garantir a proteção dos dados pessoais de acordo com a LGPD, incluindo a criptografia, o controle de acesso e a anonimização, quando necessário;
- Fornecer treinamento adequado para os funcionários que utilizarão o sistema, assegurando que estejam aptos a operá-lo eficazmente e em conformidade com a LGPD.

2. PROGRAMAÇÃO ORIENTADA A OBJETOS II

A proposta da orientação a objetos é representar o mais fielmente possível as situações do mundo real nos sistemas computacionais. Nós entendemos o mundo como um todo composto por vários objetos que interagem uns com os outros. Da mesma maneira, a orientação a objetos consiste em considerar os sistemas computacionais não como uma coleção estruturada de processos, mas sim como uma coleção de objetos que interagem entre si. (Apex, 2017)

A orientação a objetos é um paradigma de programação que se baseia na modelagem de sistemas como coleções de objetos interconectados, onde os objetos representam entidades do mundo real e possuem propriedades e comportamentos associados. A aplicação deste paradigma em sistemas de folha de pagamento envolve a utilização de uma série de métodos e tecnologias para criar um software eficiente e robusto. (Aguiar, 2015)

Alguns métodos e tecnologias comuns utilizados nesse contexto envolvem a modelagem de objetos, utilizando diagramas de classe do UML (Unified Modeling Language) para representar as classes, seus atributos e métodos. Além disso existe também a herança, encapsulamento, polimorfismo e abstração, elementos que proporcionam a simplificação de projetos e a manutenção dos sistemas. (FARINELLI, 2007)

Linguagens orientadas a objetos, como o Java, são frequentemente usadas para implementar sistemas, devido à sua capacidade de representar objetos e aplicar os princípios da orientação a objetos, sem contar que existem frameworks orientados a objetos que podem acelerar o desenvolvimento, por exemplo o Spring, fornecendo componentes reutilizáveis e soluções para problemas comuns. (Apex, 2017)

A prática de testes unitários é essencial no desenvolvimento orientado a objetos. O uso de estruturas de testes é extremamente importante para verificar a correção e a integridade do código, Java é frequentemente associado a estruturas de teste como JUnit, amplamente utilizadas para testar códigos desenvolvidos nesta linguagem de programação. (JUNGTHON & GOULART, 2009)

Em resumo, a orientação a objetos oferece um conjunto sólido de princípios e técnicas para o desenvolvimento de sistemas de folha de pagamento eficientes e robustos. A aplicação desses métodos e tecnologias, juntamente com boas práticas de engenharia de software, pode resultar em sistemas confiáveis que atendam às demandas do sistema desejado. (BARANAUSKAS, 1993)

3. DESENVOLVIMENTO DE SOFTWARE PARA INTERNET

O desenvolvimento de software para a Internet envolve a utilização de diversos métodos e tecnologias para criar aplicações web eficientes e seguras. No caso de um sistema de automatização total da folha de pagamento de uma empresa, abordado neste projeto, é fundamental adotar uma abordagem robusta e escalável. (Ferreira, 2015)

A escolha da linguagem de programação é um passo crucial e algumas linguagens populares para o desenvolvimento web incluem JavaScript (para o lado do cliente), Python, Ruby, Java, C#, PHP e Node.js (para o lado do servidor). A seleção depende dos requisitos do projeto e das habilidades da equipe. (BARANAUSKAS, 1993)

Utilizar um framework acelera o desenvolvimento e padroniza as práticas de codificação, alguns frameworks populares incluem Django e Flask (Python), Ruby on Rails (Ruby), Express.js (Node.js), Laravel (PHP), e Spring (Java). A escolha do banco de dados também é essencial, os bancos de dados relacionais (como MySQL, PostgreSQL, SQL Server) e não relacionais (como MongoDB, Cassandra) são comuns. O modelo de dados e as consultas devem ser projetados com cuidado. (ELMASRI & NAVATHE, 2005)

A arquitetura do sistema deve ser escalável e flexível. A arquitetura de três camadas é uma abordagem comum, com a adição de cache e servidores de aplicativos conforme necessário. Outro ponto é o design responsivo, garantindo que o sistema seja acessível em dispositivos móveis e desktop é fundamental, frameworks de CSS, como Bootstrap, são úteis para criar interfaces de usuário responsivas. (Silva, 2023)

No desenvolvimento de software para internet é ideal projetar APIs bem definidas para permitir integrações com outras aplicações e serviços, além de implementar práticas de segurança, incluindo autenticação, autorização, proteção contra-ataques de injeção (como SQL Injection), criptografia e auditoria de registros. (Medeiros, 2016)

Adotar práticas de DevOps e integração contínua (CI) para automatizar a compilação, testes e implantação do software, são excelentes práticas para o desenvolvimento, sem contar a utilização de sistemas e controle de versão, como Git, para rastrear e gerenciar as alterações no código-fonte. (Gandara, 2012)

O desenvolvimento de sistemas de médio porte na web é uma tarefa complexa que requer planejamento, gerenciamento de projeto eficaz e a escolha criteriosa das tecnologias e métodos adequados. Além disso, a colaboração e a comunicação eficaz entre os membros da equipe de desenvolvimento são essenciais para o sucesso do projeto. (Souza & Silva, 2021)

4. TÓPICOS ESPECIAIS DE PROGRAMAÇÃO ORIENTADA A OBJETOS

Tópicos especiais de programação orientada a objetos (POO) são conceitos avançados e técnicas que vão além dos princípios básicos da POO. Eles são frequentemente aplicados em situações complexas de desenvolvimento de software para melhorar a estrutura, manutenção e eficiência dos sistemas. (JUNGTHON & GOULART, 2009)

Esses tópicos especiais podem ser aplicados com a linguagem de programação Python, a qual vem crescendo em várias áreas da computação, como inteligência artificial, banco de dados, biotecnologia, animação 3D, aplicativos móveis (celulares), jogos e mesmo como plataforma web. Além de ser utilizado como linguagem principal no desenvolvimento de sistemas, o Python também é muito utilizado como linguagem script em vários softwares, permitindo automatizar tarefas. (Bento & Moreno)

Alguns dos tópicos envolvem padrões de projetos, herança de classes e polimorfismo, no qual Python permite que diferentes tipos de objetos sejam tratados de maneira uniforme, tornando o código mais flexível. Já em relação a interfaces e abstração, Python não possui um conceito de interface como em algumas outras linguagens, mas você pode criar interfaces usando classes abstratas e especificar contratos comuns usando métodos abstratos. Isso ajuda a garantir que as classes implementem comportamentos específicos. (Lutz, 2013)

Python suporta a composição de objetos, permitindo que sejam criados objetos complexos a partir de objetos mais simples. Isso é útil para construir estruturas de dados complexas em sistemas, como composição de objetos para representar funcionários, salários, impostos etc. Sem contar que é uma linguagem adequada para práticas de refatoração, que envolvem a reestruturação do código para melhorar a sua qualidade e manutenção e existem ferramentas e bibliotecas disponíveis para auxiliar na refatoração de código Python. (SAADE, 2010)

Python é uma linguagem flexível e poderosa que permite a aplicação de diversos tópicos especiais de programação orientada a objetos em sistemas de qualquer porte. A combinação de tais técnicas com a simplicidade e produtividade do Python pode resultar em sistemas eficientes e escaláveis que atendam às demandas necessárias. (Lutz, 2013)

5. PROJETOS DE SISTEMAS ORIENTADOS A OBJETOS

Ao projetar e implementar um sistema de administração para um pequeno empreendimento, especialmente um sistema de automatização total da folha de pagamento, é possível se beneficiar de diversos recursos e elementos essenciais orientados a objetos. (Apex, 2017)

Os principais dos recursos são as Classes e Objetos, são os fundamentos da programação orientada a objetos, permitindo a modelagem de entidades do sistema. As classes definem atributos e métodos, enquanto objetos são instâncias dessas classes, ambos facilitam a representação de entidades do mundo real, como funcionários, pagamentos e impostos e a evolução pode envolver a adição de novos atributos e métodos conforme as necessidades do sistema crescem. (FARINELLI, 2007)

O encapsulamento é outro elemento essencial, ele protege o estado interno dos objetos, ocultando detalhes de implementação e melhora a manutenibilidade do código, permitindo alterações internas sem afetar o uso externo. Os atributos são definidos como privados e o acesso a eles é feito por métodos públicos (getters e setters), à medida que o sistema cresce, novos atributos podem ser encapsulados para manter a integridade dos dados. (JUNGTHON & GOULART, 2009)

A herança permite criar hierarquias de classes, compartilhando características comuns entre elas. Nela uma classe filha (subclasse) herda atributos e métodos da classe pai (superclasse), reduzindo a duplicação de código, além de promover a reutilização e permitir modelar relacionamentos hierárquicos, como diferentes tipos de funcionários. O polimorfismo permite que objetos de classes diferentes respondam ao mesmo método de maneira única através da implementação de métodos com a mesma assinatura, e vale ressaltar que facilita a criação de código genérico e flexível, simplificando a lógica de processamento. (Aguiar, 2015)

Outro ponto importante é a interface, a qual define contratos que as classes devem cumprir, permitindo a criação de código mais flexível, melhorando a criação de componentes reutilizáveis, tornando o sistema mais modular. Neste caso as classes podem implementar múltiplas interfaces, obrigando-as a fornecer implementações para os métodos definidos na interface e conforme novas funcionalidades são adicionadas, novas interfaces podem ser introduzidas. (BARANAUSKAS, 1993)

A persistência de dados é necessária para armazenar informações importantes, como registros de pagamento e históricos, pode ser implementada com bancos de dados relacionais

ou não relacionais, porém no caso deste projeto foi implementada com banco de dados relacional MS SQL Server. Garante a integridade e a disponibilidade dos dados, e futuramente pode ser necessário escalonar a infraestrutura de armazenamento de dados. (Melo, 2016)

Vale ressaltar que os frameworks e bibliotecas orientados a objetos podem acelerar o desenvolvimento e fornecer soluções predefinidas para problemas comuns, variam dependendo da linguagem e do contexto. Para este caso foram utilizados os frameworks ASP.NET, Ionic e Angular, os quais economizam tempo de desenvolvimento e garantem boas práticas, além de gerar o aproveitamento de novos recursos e melhorias futuramente. (Souza & Silva, 2021)

Por fim, uma estratégia extremamente importante é a documentação clara, essencial para entender o funcionamento do sistema e facilitar a colaboração, onde foi realizada a inclusão de comentários no código, documentação de API e manuais de usuário. Ela melhora a comunicação entre os membros da equipe e reduz o tempo de treinamento, mas deve ser atualizada à medida que o sistema é aprimorado e novos recursos são adicionados. (Fernandes, Joia, & Andrade, 2012)

6. GERENCIAMENTO DE PROJETOS DE SOFTWARE

O Project Management Institute (PMI) especificou um conjunto de procedimentos que padronizam a gerência de projetos, envolvendo a aplicação das melhores práticas e diretrizes, com a intenção de satisfazer os clientes e outras pessoas envolvidas e/ou afetadas pelo projeto em questão. Abaixo será apresentado um resumo das principais etapas e processos com base no padrão PMI do “PMBOK Guide” (P. Management Institute, 2017):

1. Iniciação do projeto: nesta fase o Termo de Abertura do Projeto é desenvolvido, documento que autoriza formalmente o projeto, definindo seu objetivo, escopo, propósito e as partes interessadas.

2. Planejamento do projeto: etapa em que são definidos e refinados os objetivos, além do planejamento da ação necessária para alcançar a meta e o escopo para os quais o projeto foi realizado. Ela é dividida em quatro planos, especificados abaixo:

- i. Plano de gerenciamento do projeto: abrange como o projeto vai ser executado, monitorado e controlado, incluindo informações sobre o escopo, cronograma, custo, qualidade, comunicação, riscos e partes interessadas;
- ii. Escopo do projeto: são especificados os detalhes do que exatamente será entregue pelo projeto, envolvendo a documentação dos requisitos;
- iii. Cronograma do projeto: definição das atividades que serão desenvolvidas no projeto, suas dependências e sequências, resultando em um cronograma detalhado e completo;
- iv. Custos do projeto: cálculo de todos os custos envolvidos na realização do projeto, incluindo mão de obra, materiais e outros recursos.

3. Execução do projeto: fase em que são integradas as pessoas e outros recursos para que possa ser executado o plano de gerenciamento do projeto, desenvolvido na etapa anterior. São coordenadas as atividades, garantindo que os recursos sejam alocados adequadamente e que as tarefas sejam executadas conforme planejado.

4. Monitoramento e controle do trabalho do projeto: etapa que é medido e monitorado regularmente o progresso para identificar desvios e variações em relação ao plano de gerenciamento do projeto, de forma que possam ser tomadas ações corretivas para atender aos objetivos do projeto. Além disso também é realizado o controle integrado de mudanças, avaliando e aprovando.

5. Encerramento do projeto: finalização de todas as atividades do projeto, garantindo a entrega dos produtos ou serviços, documentando lições aprendidas e liberando recursos, e analisando o sucesso do projeto em relação aos critérios de aceitação e objetivos iniciais.

7. EMPREENDEDORISMO

O empreendedorismo na área de Tecnologia da Informação (TI) é um propulsor que tem desempenhado um papel fundamental na moldagem do cenário tecnológico atual. O seu impacto tem sido notável nos últimos anos, à medida que pessoas visionárias identificam oportunidades e conduzem a criação de soluções inovadoras baseadas em tecnologias de ponta. (Dornelas, 2008)

A tecnologia é uma área dinâmica e em constante evolução, tornando a inovação uma necessidade imperativa para os empreendedores, os quais se destacam ao aplicar criativamente as mais recentes tecnologias, como inteligência artificial, computação em nuvem, segurança cibernética e Internet das Coisas, para resolver problemas e atender às demandas do mercado. Além disso, os empreendedores de TI enfrentam desafios únicos relacionados à rápida obsolescência de tecnologias e ao surgimento de novas tendências. A capacidade de adaptação e aprendizado constante é um pré-requisito para o sucesso, já que os empreendedores devem estar dispostos a ajustar suas estratégias e modelos de negócios conforme o ambiente tecnológico evolui. (Gomes & Filho)

Além dos aspectos comerciais, o empreendedorismo na área de TI também se preocupa cada vez mais com questões éticas e sociais, como privacidade de dados, equidade tecnológica e sustentabilidade ambiental. Esses profissionais estão cada vez mais conscientes da responsabilidade que têm na criação de tecnologias que não apenas proporcionem benefícios econômicos, mas também considerem o impacto social e ambiental. (Freitas, 2012)

Em resumo, o empreendedorismo na área de TI é uma força transformadora que impulsiona a inovação tecnológica, promove o crescimento econômico e desempenha um papel significativo na configuração do mundo digital. Os empreendedores de sucesso nessa área são aqueles que combinam visão, criatividade, adaptabilidade e responsabilidade social para criar um impacto duradouro e significativo na sociedade. (Gomes & Filho)

Um Plano de Negócios é uma ferramenta fundamental para desenvolver e validar a viabilidade de um projeto de negócio, como o sistema de automatização da folha de pagamento abordado neste trabalho. Os principais elementos desse plano relacionados ao presente projeto são: missão; objetivos de curto, médio e longo prazo; público-alvo; principais concorrentes e estratégias para vencê-los; e estratégia de comercialização. (Salim, Hochman, Ramal, & Ramal, 2005)

8. GESTÃO DA QUALIDADE

A gestão da qualidade no desenvolvimento de sistemas informatizados é fundamental para garantir que os produtos de software atendam às necessidades dos clientes, sejam confiáveis e livres de defeitos. (Cesar, 2012)

A garantia da qualidade estabelece padrões de processos, desenvolvendo e os implementando, usando normas e metodologias que orientem o desenvolvimento do software de acordo com as melhores práticas conhecidas. Enquanto isso as auditorias de qualidade realizam auditorias regulares para verificar se os processos estão sendo seguidos de maneira correta, e se os padrões de qualidades estão sendo aplicados. (Bartié, 2002)

Os testes de software fazem parte do controle de qualidade, no qual são feitos testes funcionais, desempenho e de segurança, para identificar e corrigir erros no software, juntamente com as revisões de código que identificam os padrões estabelecidos. Já as melhorias contínuas são a análise de dados e métricas, coletando e analisando dados sobre o desempenho do software, utilizando métricas de qualidade e ações corretivas e preventivas, que implementam medidas para corrigir problemas e identificar futuros problemas. (Vasconcelos, Rouiller, Machado, & Medeiros, 2006)

Outro ponto importante é a gestão de riscos, a qual identifica e avalia os riscos que possam afetar a qualidade e funcionamento do software, implementando estratégias para reduzi-los. O envolvimento dos stakeholders também é de extrema importância, visto que mantêm uma comunicação aberta e transparente com os clientes e com as partes interessadas para garantir que as necessidades sejam atendidas. (Gusmão & Moura, 2004)

No desenvolvimento de sistemas também vale incluir o treinamento e capacitação, usados para desenvolver as habilidades da equipe e garantir que os membros tenham a capacitação adequada para desenvolver suas funções com eficiência. Além da padronização, adere a padrões e boas práticas, como seguir normas e padrões postos pela indústria, exemplo, ISO 9001, para ter uma melhor qualidade. (Rabelo, Filho, & Oliveira, 1995)

O ciclo de vida de desenvolvimento envolve escolher adotar um modelo apropriado, que atenda às necessidades do projeto e promova a sua qualidade. Esses aspectos em geral são amplamente reconhecidos na literatura acadêmica e indústria de desenvolvimento de software, sendo essenciais para o sucesso na entrega do sistema de alta qualidade, e a sua aplicação adequada pode ajudar a evitar problemas de qualidades e reduzir os custos do sistema, além de aumentar a satisfação do cliente. (Cesar, 2012)

9. DESENVOLVIMENTO DO PROJETO

Para iniciar o desenvolvimento do presente projeto foi definido o ciclo de vida do software para automação total da folha de pagamento, iniciando pela fase de análise e planejamento, onde a equipe de desenvolvimento, em conjunto com as partes interessadas, realizou uma análise detalhada dos requisitos dos usuários e das necessidades de automação da folha de pagamento. Isso incluiu a compreensão das regulamentações fiscais e trabalhistas locais e nacionais, bem como as especificidades da organização.

Durante a fase de projeto, a equipe definiu a arquitetura do software, as funcionalidades, a interface do usuário, o modelo de dados e todos os componentes necessários para automatizar a folha de pagamento, além de estabelecer os planos de teste e de implantação. Já na fase de desenvolvimento, a equipe de desenvolvimento passou a realizar a criação do software com base nas especificações definidas na fase de projeto. Isso envolve a codificação, testes unitários e a integração de componentes, e o desenvolvimento eficiente e livre de erros é essencial para garantir a qualidade e a confiabilidade do software, reduzindo a necessidade de correções pós-implantação.

Foi definido que o software será implantado em ambiente de produção, e os usuários finais começarão a utilizá-lo para a gestão da folha de pagamento. Uma implantação bem-sucedida requer planejamento cuidadoso, treinamento dos usuários e garantia de que o sistema esteja configurado corretamente para atender às necessidades da organização.

Por fim, na fase de operação e manutenção o software entra em operação e é continuamente mantido e atualizado para atender às mudanças nas regulamentações fiscais, requisitos de negócios e correções de bugs. A operação contínua e a manutenção são essenciais para garantir que o sistema permaneça eficiente, seguro e cumpra os regulamentos em constante evolução.

Com base em uma abordagem de Engenharia de Software, o levantamento e análise de requisitos para a automatização total da folha de pagamento de uma empresa envolve a identificação das necessidades dos usuários, a modelagem do sistema, a implementação de uma interface amigável, o desenvolvimento de um banco de dados eficiente e a análise econômica do projeto.

Antes de iniciar o desenvolvimento dos sistemas abordados neste projeto foi necessário definir algumas regras de negócio, detalhadas abaixo:

- O sistema deve conter os dados do empregador (CNPJ, razão social, endereço, inscrição municipal entre outros);
- O sistema deve conter os dados do empregado (nome, CPF, endereço, matrícula entre outros);
- O sistema deve apresentar o salário base, salário bruto e líquido do funcionário;
- O sistema deve apresentar descontos e impostos, como: INSS, IRRF, e ou qualquer outro que venha a ser efetuado;
- O sistema deve apresentar a data a qual pertence o demonstrativo;
- O sistema deve calcular automaticamente as deduções de impostos federais, estaduais e municipais com base nas leis tributárias vigentes;
- O sistema deve calcular e registrar os pagamentos de férias com base no tempo de serviço do funcionário;
- O sistema deve calcular e registrar os pagamentos do décimo terceiro salário de acordo com a legislação;
- O sistema deve calcular e registrar os benefícios, como plano de saúde, vale-refeição e vale-transporte, de acordo com as políticas da empresa;
- O sistema deve garantir que apenas pessoas autorizadas tenham acesso aos dados confidenciais da folha de pagamento;
- O sistema deve estar em conformidade com todas as leis trabalhistas e regulamentações fiscais aplicáveis;
- O sistema deve manter um histórico completo de pagamento e informações pessoais dos funcionários;
- O sistema deve gerar relatórios detalhados para a administração e gestão completa.

O processo de levantamento de requisitos foi feito através de “feedbacks” dos usuários, com o objetivo de entender suas expectativas e necessidades em relação ao sistema de folha de pagamento. Foram identificadas as funcionalidades essenciais, como o cálculo de salários, descontos, benefícios e impostos, além de outras características desejadas, como a geração de relatórios.

Neste projeto foi aplicada a metodologia PMI como uma forma de dar suporte às atividades de gerenciamento de projeto de software relacionadas à automação total da folha de pagamento de uma empresa. Inicialmente foram identificadas as necessidades e demandas para a automação da folha de pagamento, o que envolveu entrevistar as partes interessadas, principalmente os departamentos de recursos humanos e financeiros, que repassaram todo o conhecimento necessário para que seja possível entender os requisitos do sistema.

Em seguida foi definido o escopo do projeto, primeiro documentando detalhadamente os requisitos do sistema para essa automatização da folha de pagamento, incluindo as funcionalidades, integrações, relatórios e regulamentos fiscais a serem considerados.

Requisitos funcionais:

- R1: O sistema precisa realizar o cadastro do funcionário;
- R2: O sistema precisa permitir inclusão de novas informações do funcionário;
- R3: O sistema precisa permitir alterações cadastrais do funcionário;
- R4: O sistema precisa realizar a leituras das informações inseridas no cadastro de cada funcionário.
- R5: O sistema deve ser capaz de calcular os salários com base em informações como horas trabalhadas, salários base, benefícios, descontos, adicionais etc.
- R6: O sistema deve registrar todos os pagamentos efetuados aos funcionários, incluindo detalhes como data, tipo de pagamento, valor etc.
- R7: O sistema deve realizar a inclusão e gerenciamento de benefícios, como vales-alimentação, e descontos, como empréstimos.
- R8: O sistema precisa gerar a folha de pagamento com as informações armazenadas.

Requisitos Não-funcionais:

- RNF1: O sistema deve garantir a segurança dos dados sensíveis dos funcionários, cumprindo regulamentos de privacidade de dados.
- RNF2: O sistema deve ser capaz de lidar com grandes volumes de dados e garantir um desempenho rápido e responsivo, especialmente durante os períodos de pagamento.
- RNF3: O sistema deve ser intuitivo e de fácil utilização para os usuários, com uma interface amigável.

- RNF4: O sistema deve estar disponível de forma confiável durante os períodos de processamento de folha de pagamento, evitando tempo de inatividade.
- RNF5: O sistema deve cumprir todas as regulamentações fiscais e trabalhistas locais, garantindo a precisão dos cálculos e relatórios.
- RNF6: O sistema deve possuir um sistema robusto de backup e recuperação de dados para proteger contra perda de informações críticas.
- RNF7: Deve haver suporte técnico disponível para resolver problemas e fornecer assistência aos usuários.
- RNF8: Deve existir uma documentação clara, incluindo manuais de usuário e documentação técnica para administradores do sistema.

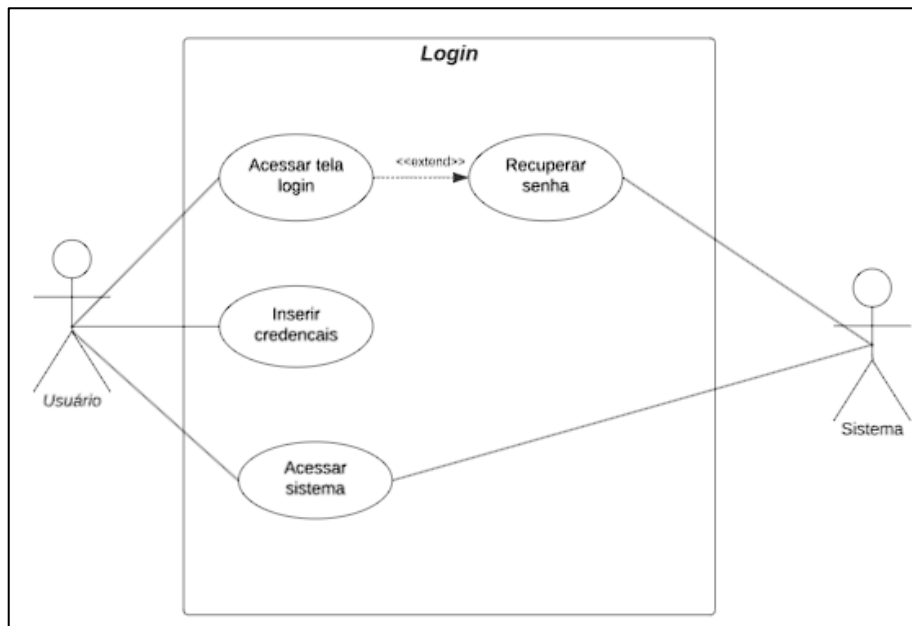
Requisitos dos Usuários:

- RU1: Os usuários devem ser capazes de cadastrar novos funcionários no sistema, fornecendo informações detalhadas, como nome, dados de contato, número de CPF, informações bancárias, informações de adicional/desconto etc.
- RU2: Os usuários devem poder registrar e gerenciar informações sobre pagamentos, incluindo horas trabalhadas, cálculo de salários, informações de impostos e deduções.
- RU4: Os usuários devem ter a capacidade de gerar relatórios personalizados, incluindo resumos de pagamentos e relatórios de gestão.
- RU5: O sistema deve permitir a definição de diferentes níveis de acesso e permissões para os usuários, garantindo que apenas pessoal autorizado possa acessar informações confidenciais.

Com base na análise dos requisitos, foi proposta uma solução que contemplasse os aspectos levantados pelos usuários. Nela serão utilizadas técnicas de modelagem, como diagramas de caso de uso e diagramas de classes, para representar as funcionalidades e as entidades envolvidas no sistema de folha de pagamento e neste projeto serão representados alguns dos diagramas gerados.

Na Figura 1 abaixo, o caso de uso exhibe o login que o ator usuário deve realizar para acessar o sistema. O usuário insere suas credenciais no sistema, onde são verificados pelo mesmo, caso seja autorizado o sistema permitirá que o usuário acesse o sistema.

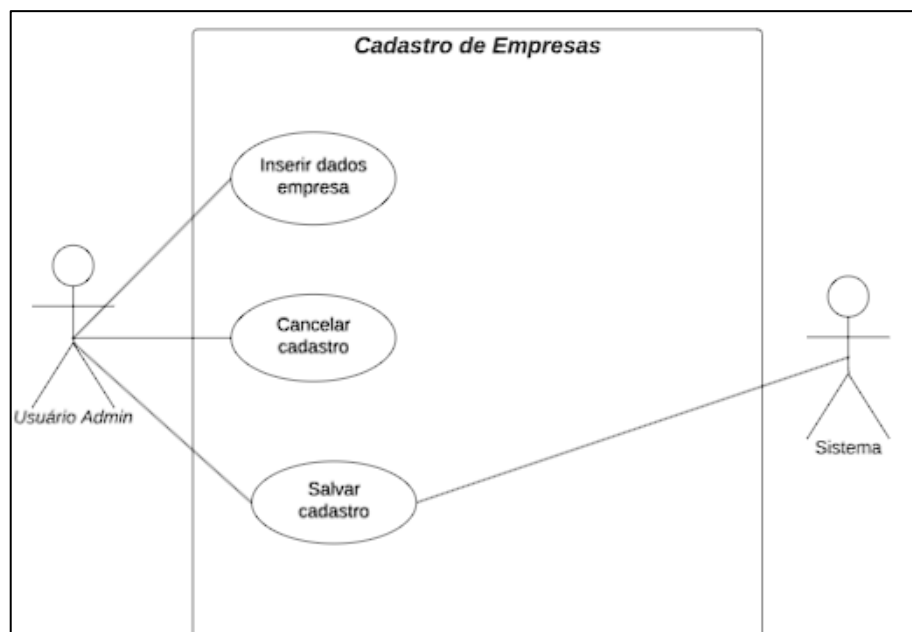
Figura 1 - Login de Usuário



Fonte: Sistema DDI

Na Figura 2, é representado a função de cadastro de empresas, onde apenas o ator designado como Admin (usuário administrador) possui as permissões necessárias para realizar o cadastro. Após efetuar o login, o usuário acessa o menu de cadastro de empresa, insere os dados empresariais e termina a sessão de cadastro, onde o sistema salva todos os dados.

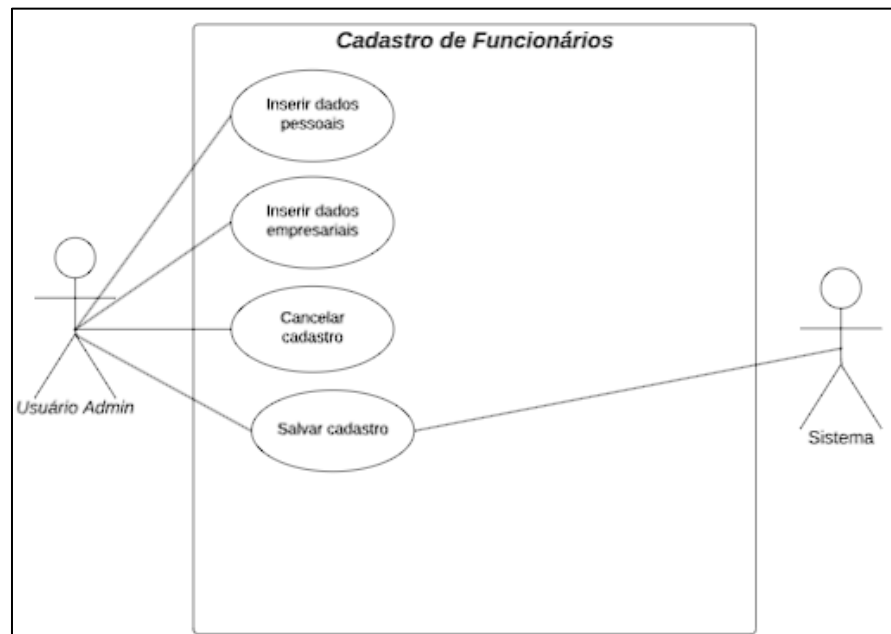
Figura 2 - Cadastro de Empresas



Fonte: Sistema DDI

Na Figura 3, o caso de uso abaixo representa a função de cadastro de funcionário, onde como o exemplo anterior somente os usuários Admin são permitidos realizar essa função. O admin após acessar o menu de cadastro de funcionário, deve inserir os dados pessoais e depois os dados empresariais.

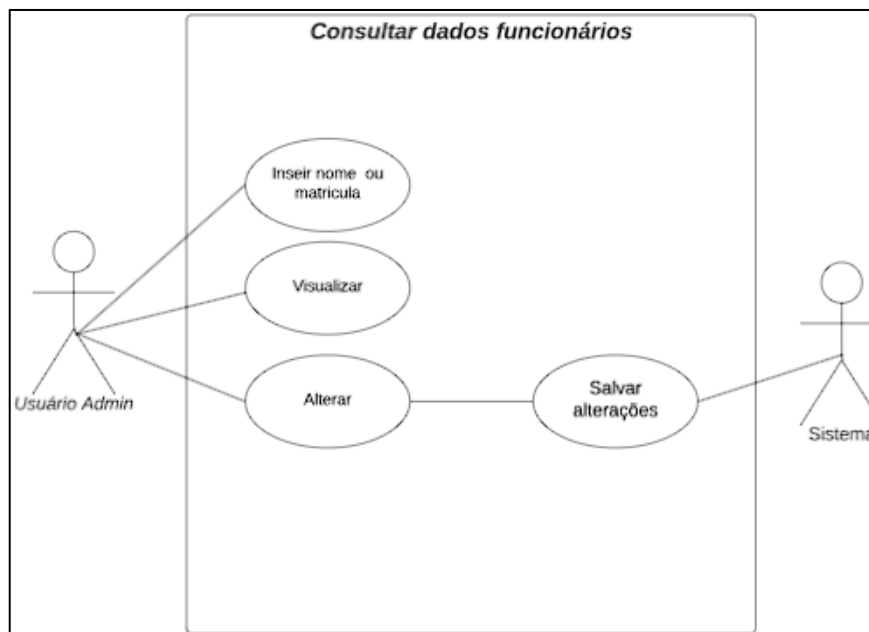
Figura 3 - Cadastro de Funcionários



Fonte: Sistema DDI

Na Figura 4, é representada a função que possibilita a consulta dos dados dos colaboradores de uma empresa, pode ser consultado tanto por usuários RH quanto por usuários administradores. Ao acessar o sistema, o usuário deve ir ao menu de consultas e pesquisar o usuário desejado.

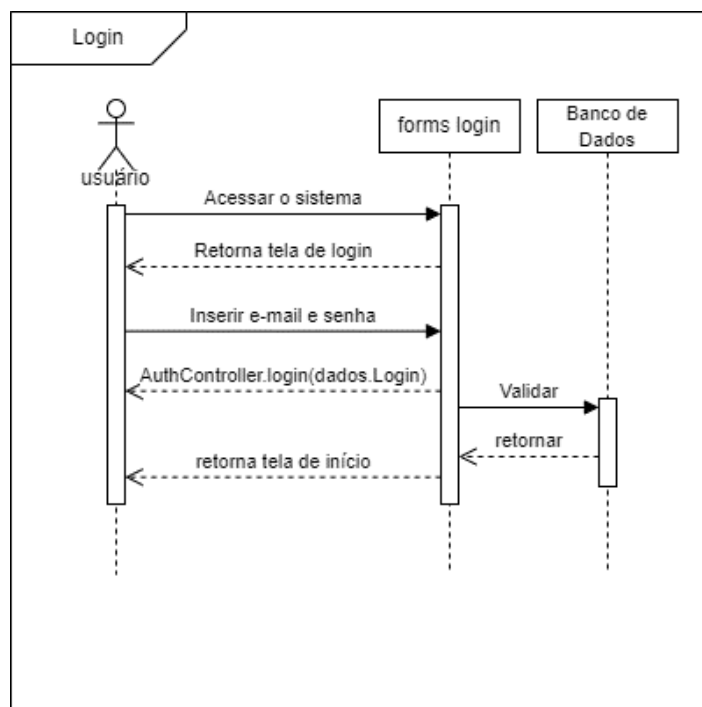
Figura 2 Consultar dados de funcionários



Fonte: Sistema DDI

Após o desenvolvimento dos casos de uso, representando as funcionalidades que o sistema oferece ao usuário, foram desenvolvidos os diagramas de sequência. Como um exemplo foi inserida a Figura 5, onde é representado o login do sistema que pode ser realizado por um usuário de qualquer nível.

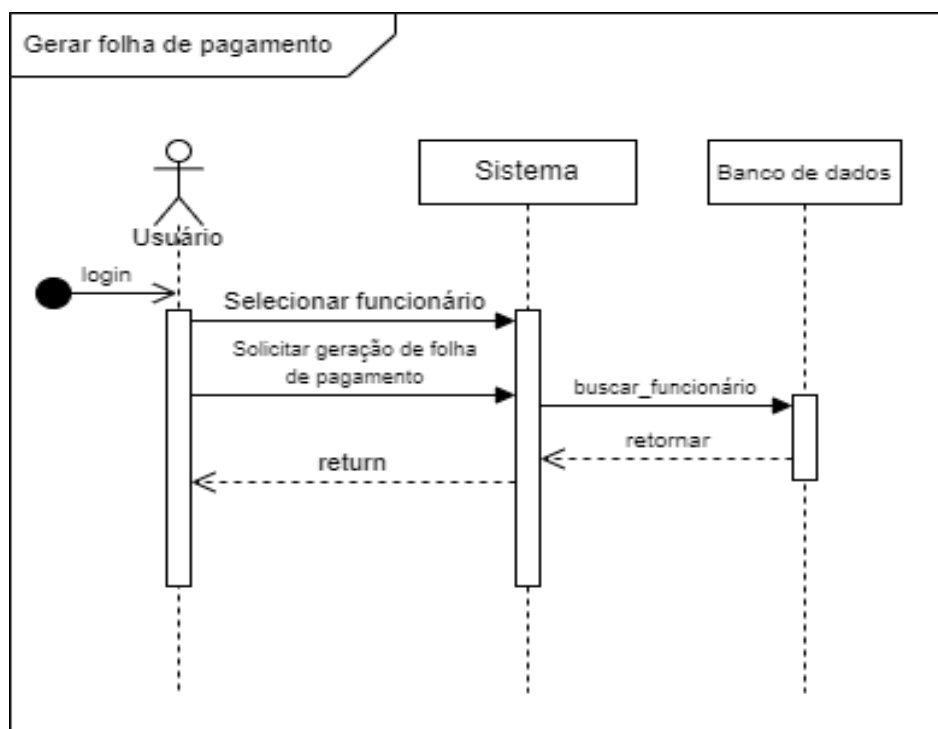
Figura 5 – Diagrama de Sequência



Fonte: Sistema DDI

A Figura 6, representa a geração da folha de pagamento, ou também chamado holerite. O Usuário, após realizar o login no sistema deve acessar o menu pagamentos e selecionar a opção gerar folha de pagamento. Logo após o sistema realizará a consulta no banco de dados, e fará todos os cálculos e operações referentes as predefinições do funcionário em questão, devolvendo a folha de pagamento.

Figura 6 – Diagrama de Sequência (Gerar folha de pagamento)

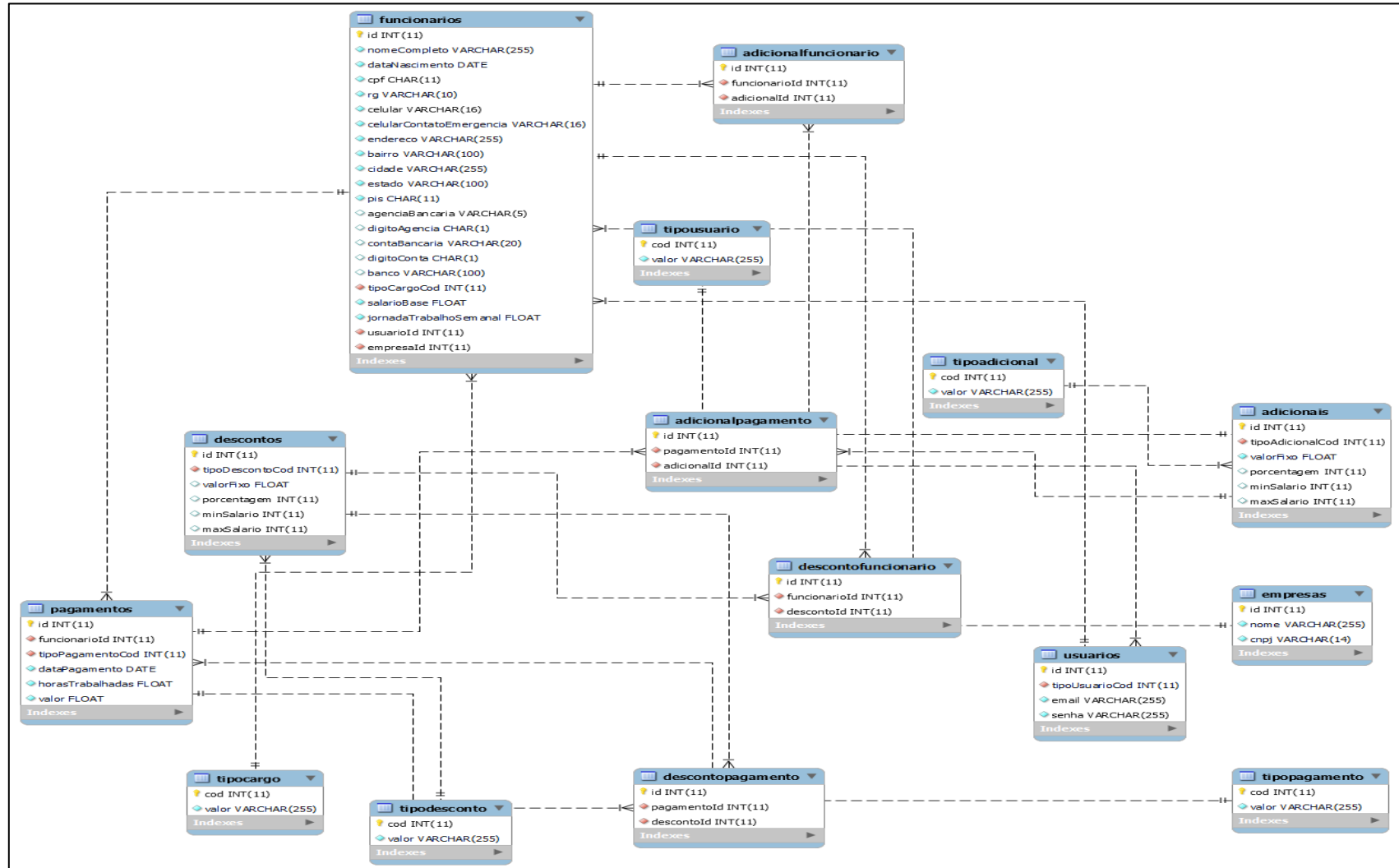


Fonte: Sistema DDI

O banco de dados servirá para armazenar as informações relacionadas à folha de pagamento de forma segura e eficiente, aplicando conceitos de modelagem de dados e integridade referencial para garantir a consistência e a integridade dos dados armazenados. Foi utilizado o Microsoft SQL Server como sistema de gerenciamento de banco de dados (SGBD) para a folha de pagamento, esse SGBD relacional possui uma interface intuitiva para manipulação, criação, alteração e visualização dos dados armazenados.

Em seguida será representado o diagrama ER do banco de dados, na Figura 7. Além disso, também é possível encontrar os scripts de criação do banco, juntamente com os scripts de inserção de dados para homologação do sistema e a planilha de testes anexados ao fim do documento.

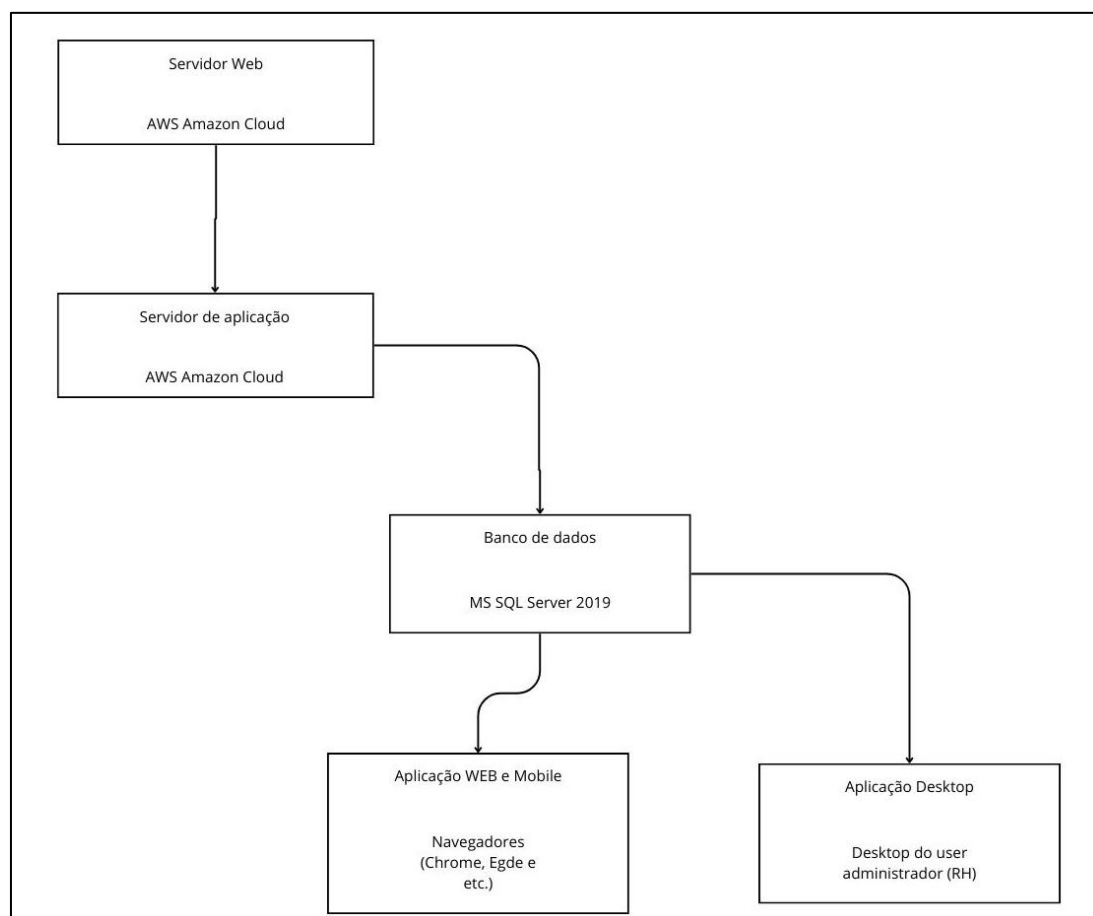
Figura 7 – Diagrama ER Banco de Dados



Fonte: Sistema DDI

O diagrama de implantação representado na Figura 8 abaixo fornece uma visão geral da estrutura e dos passos para instalar um sistema de automação total de folha de pagamento nas plataformas web e desktop. A complexidade real da implantação pode variar dependendo das necessidades específicas da sua organização e da escolha de tecnologias e ferramentas.

Figura 8 – Diagrama de Implantação



Fonte: Sistema DDI

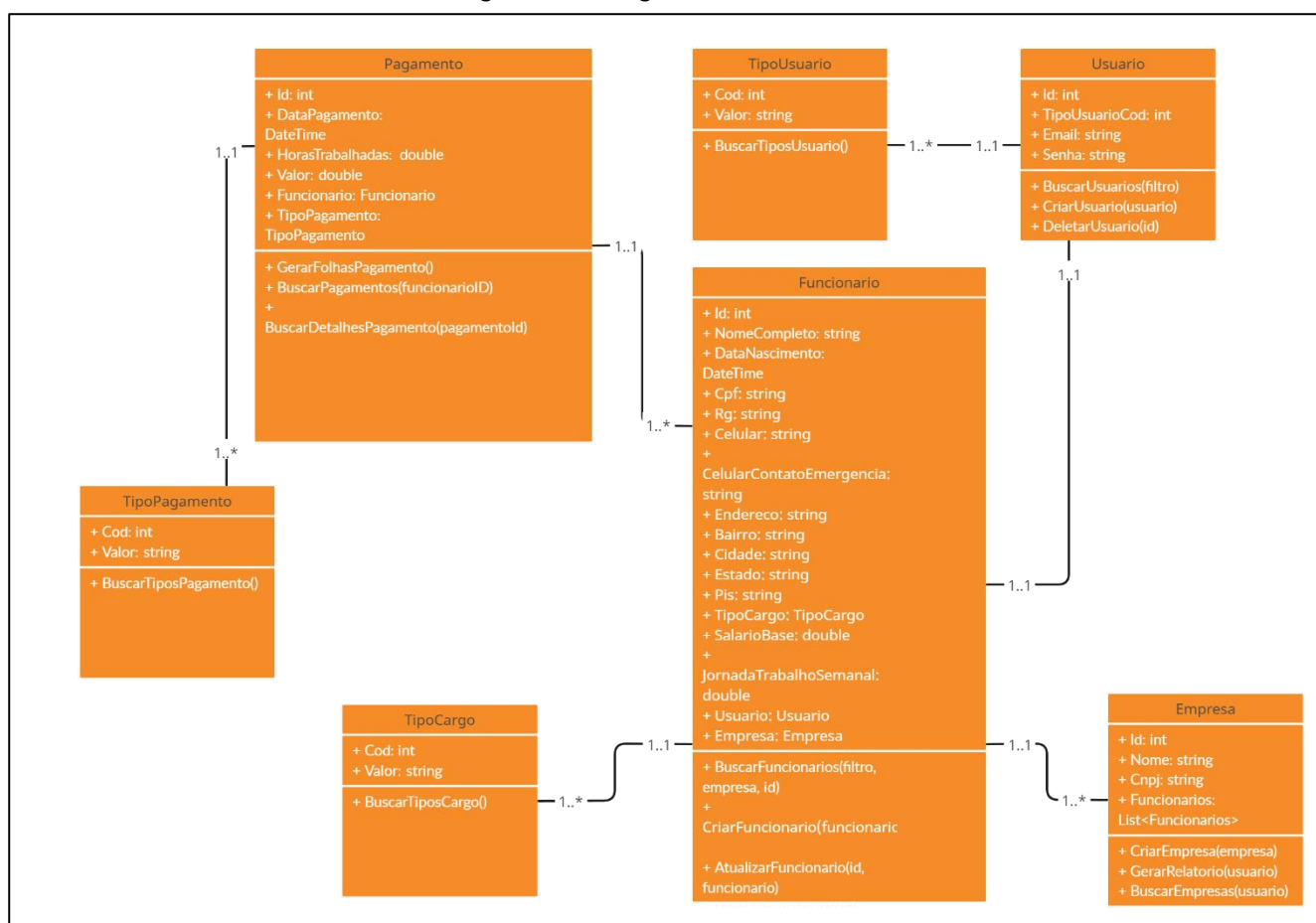
Na etapa de execução do projeto foi montada uma equipe de desenvolvimento de software e liderança do projeto, utilizando a metodologia ágil *SCRUM*, onde definimos o *SCRUM Master*, o *Product Owner* e os integrantes do desenvolvimento, explicando claramente o papel de cada um e suas responsabilidades.

Para o desenvolvimento back-end foi selecionada a linguagem de programação C# com framework ASP.NET e em relação ao desenvolvimento front-end foi utilizada a tecnologia JavaScript para criar a interface do usuário (UI), considerando o uso de Ionic e Angular para garantir eficiência no projeto.

Quanto à segurança, o uso da Lei Geral de Proteção de Dados se tornou um fator obrigatório, visto a necessidade de garantir a confiabilidade no armazenamento e acesso dos dados de todos os funcionários da empresa em questão. Além disso, também foi incluída a hierarquia de usuários (especificados em ADMIN, RH e Funcionário geral) para que apenas pessoas previamente autorizadas possam realizar funções específicas e tenham acesso a dados pessoais.

A Figura 9 abaixo representa o diagrama de classe referente ao sistema, especificando o nome das classes, a lista de atributos e os métodos dela.

Figura 9 – Diagrama de Classes



Fonte: Sistema DDI

Durante o projeto houve o monitoramento e controle do trabalho do projeto, acompanhando o desenvolvimento do sistema em relação ao cronograma e escopo estipulados. Também foi realizado o controle integrado de mudanças, onde precisamos avaliar constantemente e aprovar ou rejeitar as mudanças propostas ou necessárias no escopo, cronograma e custo do projeto conforme foram aparecendo.

Para a liberação do sistema foram realizados testes de usabilidade com cinco participantes representativos do produto, com idade entre 25 e 60 anos, ambos os sexos e atuação voltada para o ramo de administração e recursos humanos. Essa pesquisa qualitativa teve foco em testar pontos que ofereciam riscos ao sistema e que poderiam impactar diretamente no usuário ou no negócio, proporcionando uma avaliação direta de quais aspectos do design eram problemáticos e quais funcionavam bem.

Para a realização dos testes qualitativos foram definidas duas tarefas, sendo uma para o sistema DESKTOP e uma para o WEB, as quais serão descritas abaixo:

- **Tarefa 1:** Cadastro de Funcionário
 - Descrição da tarefa: Imagine que você é um gerente de RH responsável por adicionar um novo funcionário ao sistema. Sua tarefa é realizar o processo completo de cadastro de um novo funcionário, incluindo os dados de informações pessoais e empresariais.
 - Plataforma testada: DESKTOP
- **Tarefa 2:** Exclusão de usuário
 - Descrição da tarefa: Pense que você é um dos funcionários do RH e precisa realizar a exclusão do usuário de um ex-funcionário. Acesse o sistema e exclua o usuário de Gabriel Henrique Pereira.
 - Plataforma testada: WEB

Ambas as tarefas foram executadas de forma virtual, gravando as sessões pela plataforma do Teams, as quais foram autorizadas previamente pelos participantes. Verificando os feedbacks das tarefas realizadas foi possível identificar a necessidade de mudança nas cores dos botões de “Avançar” e “Voltar/Cancelar” visto que houve dificuldade na leitura das palavras. Outro ponto importante foi a inclusão de dados dos últimos funcionários consultados na tela de “Consulta”, identificado como um ponto importante pelos participantes, visto que já tem experiência com esse tipo de sistema.

Foi definido que o sistema será atualizado frequentemente, para proporcionar qualidade constantemente e uma excelente experiência ao usuário, além da frequência em manutenções para corrigir quaisquer bugs e aumentar a segurança. Para facilitar a manutenção e a colaboração será mantida em documentação clara o código fonte e as APIs, explicitamente comentadas para fácil entendimento da equipe.

Por fim, após a conclusão bem-sucedida do desenvolvimento do sistema de folha de pagamento automatizada, a equipe se certificou de que todos os requisitos foram atendidos e

que o sistema estava totalmente pronto para uso. Foi feita uma revisão pós-projeto como forma de identificar lições aprendidas e oportunidades de melhoria, além da avaliação do sucesso do projeto em relação aos critérios de aceitação e objetivos iniciais, que foi compartilhada com as partes interessadas.

Neste projeto também foi levada em consideração a gestão estratégica de recursos humanos, que está diretamente relacionada ao sistema de folha de pagamento de uma organização. Algumas maneiras pelas quais a gestão estratégica de recursos humanos está relacionada ao sistema de folha de pagamento são:

- Precisão e conformidade: um sistema de folha de pagamento eficiente garante que os efeitos e benefícios sejam calculados corretamente e em conformidade com as leis trabalhistas e regulamentações fiscais. Isso evita erros no pagamento e ajuda a evitar problemas legais ou financeiros para a organização.
- Remuneração competitiva: a remuneração é uma das principais ferramentas de gestão de pessoas. Um sistema de folha de pagamento eficaz permite que uma organização estabeleça e administre recompensas e benefícios competitivos, satisfatórios com o mercado e capazes de atrair e reter talentos qualificados.

A média salarial para os grupos profissionais relacionados à equipe técnica do projeto de sistema de folha de pagamento pode variar dependendo de vários fatores, como a localização geográfica, o nível de experiência, a qualificação e a demanda do mercado. Para a formação da equipe técnica do projeto seriam necessários os profissionais de: Analista de Sistemas: R\$ 4.000 - R\$ 8.000 por mês; Programador: R\$ 3.000 - R\$ 6.000 por mês; e Especialista em Folha de Pagamento: R\$ 5.000 - R\$ 10.000 por mês.

Focalizando na questão de empreendedorismo, o pontapé inicial é a definição da missão do sistema, a qual foi realizada em equipe durante reuniões específicas: “Nossa missão é simplificar e otimizar o processo de gerenciamento de folha de pagamento para empresas de todos os tamanhos, oferecendo soluções eficazes e econômicas.”

Em seguida foram determinados outros objetivos, sendo eles o objetivo de curto prazo, ao qual o time se dispõe em lançar a versão beta do sistema para o primeiro cliente (empresa especializada em Recursos Humanos e Departamento Pessoal) em 4 meses, na intenção de atender a geração de folha de pagamento e coletar feedbacks para melhorias. O objetivo de médio prazo está integrado entre 1 e 2 anos, com foco na expansão da base de clientes, além de aprimorar a funcionalidade do sistema e consolidar a presença no mercado. Por fim, o

último objetivo foi o de longo prazo, definido em tornar-se líder de mercado, explorar mercados internacionais e diversificar os produtos oferecidos, em um tempo de mais de 4 anos.

O público-alvo inicial estará localizado geograficamente no estado de São Paulo, porém com intenção de expansão para o Brasil inteiro, visto que é um produto digital e pode ser vendido e acompanhado on-line. O sistema poderá atender empresas de pequeno, médio e grande porte, que desejam economizar tempo e recursos, reduzir erros e garantir a conformidade com regulamentações fiscais, além de gestores financeiros, que têm interesse direto na gestão eficiente dos recursos financeiros da empresa.

No mercado já existem diversas empresas especializadas na automatização da folha de pagamento, sendo algumas delas: DQL Sistemas, Trisoftcon, SuperSoft e JB Software. A estratégia para inovar tecnologicamente é de ter um pré-cadastro de cargos e salários específico para a empresa e geração dos cálculos de impostos, férias, 13º salário, entre outros automaticamente, reduzindo erros de digitação e proporcionando mais rapidez. Além disso, será oferecido um preço mais atraente e flexível, com suporte de alta qualidade e treinamento aos usuários.

Para comercializar o sistema foi definido que serão vendidas licenças de uso com assinatura mensal, trimestral, semestral ou anual, com valor específico de acordo com as funcionalidades liberadas já pré-definidas em contrato, que inclui também o suporte técnico remoto, com atualizações e manutenções. Como forma de alcançar os clientes será utilizado o marketing digital, com website profissional, mídias sociais e geração de conteúdo demonstrando o conhecimento na área e fornecendo valor, além do investimento na publicidade paga (Google Ads) e tráfego pago nas redes sociais, participação em eventos do setor, liberação de períodos de teste para potenciais clientes e parcerias estratégicas com outras organizações que possam promover o sistema.

10. CONCLUSÃO

Com base nas extensas pesquisas realizadas pela nossa equipe, o desenvolvimento deste projeto se destina a atender empresas de diversos tamanhos, abrangendo desde pequenos negócios até grandes corporações. A proposta do projeto é substituir processos repetitivos e a utilização de documentos em papel na criação de fichas de funcionários.

Este sistema oferece às empresas que o adotam a capacidade de promover uma gestão mais ágil e eficiente em suas operações. Além disso, possibilita a redução de custos e minimiza erros frequentemente decorrentes de falhas humanas. A automação da folha de pagamento economiza consideravelmente o tempo dos colaboradores, permitindo que eles se concentrem em tarefas de maior relevância, em vez de perder tempo calculando manualmente salários e impostos, por exemplo.

O sistema também se destaca por suas sólidas medidas de segurança, que visam proteger as informações pessoais dos usuários, reduzindo substancialmente o risco de vazamentos de dados e violações de privacidade. Além disso, a automação da folha de pagamento é notavelmente mais precisa do que os processos manuais, garantindo precisão nos cálculos salariais, impostos e benefícios.

Apesar de um custo inicial significativo associado à implementação do sistema, os benefícios a longo prazo tendem a compensar esses custos iniciais quando comparados aos processos manuais. No geral, essas vantagens contribuem de forma considerável para a eficiência operacional e a saúde financeira de uma organização. Ao mesmo tempo, oferecem maior segurança e reduzem significativamente o risco de violações de dados e invasões.

GLOSSÁRIO

API - conjunto padrão de formatos de dados, funções, chamadas e interrupções de software que pode ser utilizado por uma aplicação para acessar serviços de rede, dispositivos ou sistemas operacionais.

Arquitetura da Informação (AI) – prática que busca desenvolver um método de apresentação e disposição das informações.

Atributo - são os elementos que definem a estrutura de uma classe.

Back-end – é a estrutura que possibilita a operação do sistema.

Ciclos de Desenvolvimento do Software – é um modelo composto por sete fases: Análise de Requisitos, Estudo de Viabilidade, Design, Codificação, Teste, Implementação e Manutenção.

Classe – é uma forma de definir um tipo de dado, sendo formada por dados e comportamentos

Computação em Nuvem - recursos de TI sob demanda por meio da Internet com definição de preço de pagamento conforme o uso.

Dados - são informações brutas. Caracterizadas como coleções de fatos que devem ser processados para serem significativos.

Design de Interação (IxD) - é a forma como um produto interage em conjunto com seus usuários.

Engenharia de Software - engloba um conjunto de métodos, ferramentas e processos, que possibilitam ao administrador controlar o processo de desenvolvimento de software e que fornecem ao desenvolvedor uma base para construir com eficiência um software com alta qualidade.

Estudo de caso – método de pesquisa que consiste na utilização de um ou mais métodos qualitativos, coletados a partir de eventos reais, com objetivo de explicar fenômenos inseridos no seu próprio contexto.

Experiência do usuário (UX) – experiência que o usuário tem com o produto de software.

Feedback - resposta dada a um estímulo como uma maneira de avaliá-lo.

Fluxo do Usuário - técnica que permite mapear todo o fluxo de telas do sistema. Sendo útil para alinhar os caminhos e ações que o usuário que pode realizar.

Framework - é um conjunto de bibliotecas, que abordam funcionalidades, e estruturas, para o desenvolvimento de aplicações.

Front-end - é tudo que envolve a parte visível de um site ou aplicação, com a qual os usuários podem interagir.

Hardware - equipamento físico, englobando dispositivos mecânicos, elétricos ou eletrônicos

Inteligência artificial (IA) - área das ciências da computação interessada em fazer com que os computadores se comportem como seres humanos.

Interface do Usuário (UI) - espaço onde os usuários interagem com o usuário.

Java - linguagem de programação criada pela Sun Microsystems e projetada especificamente para escrever applets, os quais podem ser transferidos com segurança, via Internet, para o computador do usuário.

Linguagem de programação – um vocabulário com conjunto de regras gramaticais com a finalidade de ser executada pelo computador

Objeto - entidade autocontida que inclui dados e procedimentos para manipulá-los.

OTAN – Organização do Tratado do Atlântico Norte.

Product Owner - responsável por garantir que todo o time envolvido nesse projeto esteja fazendo o seu máximo e utilizando os recursos disponíveis da maneira mais eficiente.

Requisitos - requisitos de um projeto de software são as funções, recursos e restrições que precisam ser atendidos pelo produto.

Requisitos funcionais - referem-se às funcionalidades que o sistema deve ser capaz de realizar.

Requisitos não funcionais – refere-se as restrições que um sistema deve operar.

Scrum – metodologia ágil baseada em sprints, garantindo a otimização do tempo para o desenvolvimento de um projeto.

Scrum Master - responsável por gerenciar todo o processo de implementação da metodologia Scrum.

Sistema – Consiste em um conjunto de dispositivos eletrônicos capazes de processar informações de acordo com um programa.

Software - conjunto de instruções que permitem ao usuário executar uma tarefa.

Stakeholders – são todas as pessoas, empresas ou instituições que têm algum tipo de interesse na gestão e nos resultados de um projeto ou organização.

11.REFERÊNCIAS

- Aguiar, J. (2015). Experiência baseada em Gamificação no Ensino sobre Herança em Programação Orientada a Objetos.
- Andrade, S. C., & Tait, T. F. (16 de 04 de 2012). Uma aplicação do guia PMBOK na gestão de projetos de software. pp. 2-11.
- Apex. (2017). O que é programação orientada a objetos?
- BARANAUSKAS, M. C. (1993). Procedimento, função, objeto ou lógica? Linguagens de programação vistas pelos seus paradigmas. Computadores e Conhecimento: Repensando a Educação. Campinas, SP, Gráfica Central da Unicamp.
- Bartié, A. (2002). Garantia da Qualidade de Software. Campus.
- Cavalcante, L. E. (2000). Gestão estratégica de recursos humanos na era da tecnologia da informação e da globalização.
- Cesar, L. R. (2012). Gestão da Qualidade - Conceitos e Técnicas. São Paulo: Atlas S.A.
- Dornelas, J. C. (2008). Empreendedorismo - Transformando ideias em negócios. Rio de Janeiro: Elsevier.
- ELMASRI, R., & NAVATHE, S. B. (2005). Sistemas de Banco de Dados. São Paulo: Pearson.
- FARINELLI, F. (2007). Conceitos básicos de programação orientada a objetos. Instituto Federal Sudeste de Minas Gerais.
- Fernandes, C. C., Joia, L. A., & Andrade, A. (Março de 2012). Resistência à implantação de sistemas de folha de pagamento na administração pública: um estudo multi-caso.
- Ferreira, G. F. (31 de Agosto de 2015). Arquitetura da informação no desenvolvimento de aplicação web.
- Freitas, C. S. (Agosto de 2012). O SOFTWARE PÚBLICO BRASILEIRO: novos modelos de cooperação econômica entre Estado e Sociedade Civil. pp. 99-113.
- Gandara, F. (2012). Qualidade e Teste em Software. Março: Clube de Autores.
- Gomes, J. A., & Filho, J. M. (s.d.). Comportamento empreendedor em empresas desenvolvedoras de software: um estudo exploratório.
- Greatti, L., & Senhorini, V. M. (Outubro de 2000). Empreendedorismo - Uma Visão Comportamentalista. pp. 22-34.
- Gusmão, C. M., & Moura, H. P. (2004). Gerência de Risco em Processos de Qualidade de Software: uma Análise Comparativa.

JUNGTHON, G., & GOULART, C. M. (2009). Paradigmas de Programação. Monografia (Monografia)—Faculdade de Informática de Taquara.

Lutz, M. (2013). Learning Python. O'Reilly Media.

Martens, C. D., Freitas, H. M., & Andres, R. (2 de Agosto de 2011). Desenvolvimento da orientação empreendedora em empresas de software: proposições preliminares.

Martins, J. C. (2007). Técnicas para Gerenciamento de Projetos de Software. Rio de Janeiro: Brasport.

Martins, J. C. (2010). Gerenciando Projetos de Desenvolvimento de Software Com Pmi, Rup e Uml. Brasport.

Medeiros, O. L. (16 de Dezembro de 2016). Segurança em aplicações Web.

Melo, L. S. (2016). Uma abordagem para geração de código de persistência de dados hibernate baseado em MDD e POA.

P. Management Institute. (2017). Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK). Newtown Square: Project Management Institute, Inc.

Rabelo, F. M., Filho, E. B., & Oliveira, C. A. (1995). Treinamento e gestão da qualidade.

SAADE, J. (2010). C# Guia do programador. São Paulo: Novatec.

Salim, C. S., Hochman, N., Ramal, A. C., & Ramal, S. A. (2005). Construindo Planos de Negocios. Rio de Janeiro: Elsevier.

Silva, G. (2023). O que é usabilidade? Fonte: Coodesh

Silva, V. L. (2012). Utilização de técnicas de gestão de projetos na análise de requisitos de projetos de software.

Souza, M., & Silva, E. A. (7 de Abril de 2021). Estudo Comparativo de Tecnologias de Desenvolvimento.

Vasconcelos, A. M., Rouiller, A. C., Machado, C. A., & Medeiros, T. M. (2006). Introdução à Engenharia de Software e à Qualidade de Software. 81-82.

Xavier, C. M. (2010). Metodologia de Gerenciamento de Projeto. pp. 1-3.

ANEXO A – SCRIPTS DE CRIAÇÃO DE TABELAS (1)

```
CREATE TABLE TipoPagamento (  
  cod INT IDENTITY(1,1) NOT NULL,  
  valor VARCHAR(255),  
  PRIMARY KEY (cod)  
);
```

```
CREATE TABLE TipoDesconto (  
  cod INT IDENTITY(1,1) NOT NULL,  
  valor VARCHAR(255),  
  PRIMARY KEY (cod)  
);
```

```
CREATE TABLE TipoAdicional (  
  cod INT IDENTITY(1,1) NOT NULL,  
  valor VARCHAR(255),  
  PRIMARY KEY (cod)  
);
```

```
CREATE TABLE TipoCargo (  
  cod INT IDENTITY(1,1) NOT NULL,  
  valor VARCHAR(255),  
  PRIMARY KEY (cod)  
);
```

```
CREATE TABLE TipoUsuario (  
  cod INT AUTO_INCREMENT NOT NULL,  
  valor VARCHAR(255),  
  PRIMARY KEY (cod)  
);
```

```
CREATE TABLE Empresa (  
  id INT IDENTITY(1,1) NOT NULL,  
  nome VARCHAR(255) NOT NULL,  
  cnpj VARCHAR(14) NOT NULL UNIQUE,  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE Pagamentos (  
  id INT IDENTITY(1,1) NOT NULL,  
  funcionariold INT NOT NULL,  
  tipoPagamentoCod INT NOT NULL,  
  dataPagamento DATE NOT NULL,  
  horasTrabalhadas FLOAT NOT NULL,  
  valor FLOAT NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (tipoPagamentoCod) REFERENCES TipoPagamento (cod),  
  FOREIGN KEY (funcionariold) REFERENCES Funcionario (id)  
);
```

ANEXO B – SCRIPTS DE CRIAÇÃO DE TABELAS (2)

```
CREATE TABLE Funcionario (  
  id INT IDENTITY(1,1) NOT NULL,  
  nomeCompleto VARCHAR(255) NOT NULL,  
  dataNascimento DATE NOT NULL,  
  cpf CHAR(11) NOT NULL UNIQUE,  
  rg VARCHAR(10) NOT NULL UNIQUE,  
  celular VARCHAR(16) NOT NULL,  
  celularContatoEmergencia VARCHAR(16) NOT NULL,  
  endereco VARCHAR(255) NOT NULL,  
  bairro VARCHAR(100) NOT NULL,  
  cidade VARCHAR(255) NOT NULL,  
  estado VARCHAR(100) NOT NULL,  
  pis CHAR(11) NOT NULL,  
  agenciaBancaria VARCHAR(5) NOT NULL,  
  digitoAgencia CHAR(1),  
  contaBancaria VARCHAR(20) NOT NULL,  
  digitoConta CHAR(1),  
  banco VARCHAR(100) NOT NULL,  
  tipoCargoCod INT NOT NULL,  
  salarioBase FLOAT NOT NULL,  
  jornadaTrabalhoSemanal FLOAT NOT NULL,  
  usuariId INT NOT NULL,  
  empresId INT NOT NULL,  
  FOREIGN KEY (empresId) REFERENCES Empresa (id),  
  FOREIGN KEY (usuariId) REFERENCES Usuarios (id),  
  FOREIGN KEY (tipoCargoCod) REFERENCES TipoCargo (cod),  
  PRIMARY KEY (id)  
);  
  
CREATE TABLE Descontos (  
  Id INT IDENTITY(1,1) NOT NULL,  
  tipoDescontoCod INT NOT NULL,  
  valorFixo FLOAT,  
  porcentagem INT,  
  minSalario INT,  
  maxSalario INT,  
  PRIMARY KEY (id),  
  FOREIGN KEY (tipoDescontoCod) REFERENCES TipoDesconto (cod)  
);  
  
CREATE TABLE Usuarios (  
  id INT IDENTITY(1,1) NOT NULL,  
  tipoUsuarioCod INT NOT NULL,  
  email VARCHAR(255) NOT NULL UNIQUE,  
  senha VARCHAR(255) NOT NULL,  
  FOREIGN KEY (tipoUsuarioCod) REFERENCES TipoUsuario (cod),  
  PRIMARY KEY (id)  
);
```

ANEXO C – SCRIPTS DE CRIAÇÃO DE TABELAS (3)

```
CREATE TABLE Adicionais (  
  id INT IDENTITY(1,1) NOT NULL,  
  tipoAdicionalCod INT NOT NULL,  
  valorFixo FLOAT NOT NULL,  
  porcentagem INT,  
  minSalario INT,  
  maxSalario INT,  
  PRIMARY KEY (id),  
  FOREIGN KEY (tipoAdicionalCod) REFERENCES TipoAdicional (cod)  
);
```

```
CREATE TABLE AdicionalFuncionario (  
  id INT IDENTITY(1,1) NOT NULL,  
  funcionarioId INT NOT NULL,  
  adicionalId INT NOT NULL,  
  FOREIGN KEY (funcionarioId) REFERENCES Funcionario (id),  
  FOREIGN KEY (adicionalId) REFERENCES Adicionais (id),  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE DescontoFuncionario (  
  id INT IDENTITY (1,1) NOT NULL,  
  funcionarioId INT NOT NULL,  
  descontoId INT NOT NULL,  
  FOREIGN KEY (funcionarioId) REFERENCES Funcionario (id),  
  FOREIGN KEY (descontoId) REFERENCES Descontos (id),  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE AdicionalPagamento (  
  id INT IDENTITY (1,1) NOT NULL,  
  pagamentoId INT NOT NULL,  
  adicionalId INT NOT NULL,  
  FOREIGN KEY (pagamentoId) REFERENCES Pagamentos (id),  
  FOREIGN KEY (adicionalId) REFERENCES Adicionais (id),  
  PRIMARY KEY (id)  
);
```

```
CREATE TABLE DescontoPagamento (  
  id INT IDENTITY(1,1) NOT NULL,  
  pagamentoId INT NOT NULL,  
  descontoId INT NOT NULL,  
  FOREIGN KEY (pagamentoId) REFERENCES Pagamentos (id),  
  FOREIGN KEY (descontoId) REFERENCES Descontos (id),  
  PRIMARY KEY (id)  
)
```

ANEXO D – SCRIPTS DE INSERÇÃO DE DADOS

INSERT INTO TipoPagamento (valor) VALUES ('Mensal');

INSERT INTO TipoDesconto (valor) VALUES ('INSS');

INSERT INTO TipoAdicional (valor) VALUES ('Hora extra');

INSERT INTO TipoCargo (valor) VALUES ('Analista de sistemas');

INSERT INTO TipoUsuario (valor) VALUES ('Administrador');

INSERT INTO Empresas (cnpj) values ('49942559000120');

INSERT INTO Usuarios (tipoUsuarioCod, email, senha) values (1, 'pedro@ddiapp.com', '2gd%2dD*s');

INSERT INTO Funcionarios (id, nomeCompleto, dataNascimento, cpf, rg, celular, celularContatoEmergencia, endereço, bairro, cidade, estado, pis, agenciaBancaria, digitoAgencia, contaBancaria, digitoConta, banco, tipoCargoCod, salarioBase, jornadaTrabalhoSemanal, usuarioid, empresaid) VALUES (1, 'Pedro Ribeiro', '2001-05-16', '67505212826', '302087795', '12996438369', '12997255585', 'Rua dos Pardais', 'Vila Tatetuba', 'Sao Jose dos Campos', 'SP', '34941678150', '00000', '1', '29170808', '8', 'Nu Pagamentos SA', 1, 10000, 40, 1, 1);

INSERT INTO Pagamentos (funcionarioid, tipoPagamentoCod, dataPagamento, horasTrabalhadas, valor) values (1,1,'2023-11-05', 40, 10000);

INSERT INTO Descontos (tipoDescontoCod, valorFixo, porcentagem, minSalario, maxSalario) values (1,1500, 10, 1000, 2000);

INSERT INTO Adicionais (tipoAdicionalCod, valorFixo, porcentagem, minSalario, maxSalario) values (1,1500, 10, 1000, 2000);

INSERT INTO AdicionalFuncionario (funcionarioid, adicionalId) values (1,1);

INSERT INTO DescontoFuncionario (funcionarioid, descontoid) values (1,1);

INSERT INTO AdicionalPagamento (pagamentoid, adicionalId) values (3,1);

INSERT INTO DescontoPagamento (pagamentoid, descontoid) values (3,1);

FICHA DE CONTROLE DO PIM

Grupo Nº Ano: 2023 Período: 3ºDS3Q48/4ºDS4R48

Coordenador: Prof Roberto Cordeiro Waltz

Tema (Área do projeto): Desenvolvimento de um sistema para automatização total da folha de pagamento de uma empresa

Alunos

RA	Nome	E-mail	Curso	Visto do aluno
N84333-0	Allan Pereira Lopes	allan.lope5@aluno.unip.br	CST em ADS	
N9716G-2	Giovanna Chiquini Primon	giovanna.primon@aluno.unip.br	CST em ADS	
N037FF-6	João Pedro Gonçalves Manara	joao.manara@aluno.unip.br	CST em ADS	
G50511-5	Matheus Felix Barbosa	matheus.barbosa55@aluno.unip.br	CST em ADS	

Registros

Data do encontro	Observações
28/08/2023	Reunião para criação de nome do sistema, regras de negócio e requisitos
04/09/2023	Reunião para divisão das sprints e priorização do backlog
11/09/2023	Reunião para verificação dos diagramas e criação dos projetos
18/09/2023	Reunião para atualização do trabalho já realizado
25/09/2023	Reunião para revisão do escopo do projeto
02/10/2023	Reunião para visualização dos testes de usabilidade

09/10/2023	Reunião para ajustes finais da documentação
16/10/2023	Reunião para finalização dos manuais do sistema
23/10/2023	Reunião para verificação da formatação e avaliação pela equipe