



UNIVERSIDADE PAULISTA

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	R.A
Diego Bueno Martins	N0339B3
Gabriella B. Fagnani	T597EJ5
João Vitor Lopes Galvão	N018496
Marco Antonio Nogueira de Lima	G0238B5
Matheus Junichi Takano	G472152
Victor Henrique Barbosa de Andrade	G45EFC3

SÃO JOSÉ DOS CAMPOS – SP

ABSTRACT

NOVEMBRO/2023

Nome	RA
Diego Bueno Martins	N0339B3
Gabriella B. Fagnani	T597EJ5
João Vitor Lopes Galvão	N018496
Marco Antonio Nogueira de Lima	G0238B5
Matheus Junichi Takano	G472152
Victor Henrique Barbosa de Andrade	G45EFC3

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

RESUMO

Este resumo descreve um projeto integrado multidisciplinar que teve como objetivo desenvolver um software de folha de pagamento para uma empresa. A metodologia adotada envolveu pesquisas em diversas fontes, como websites, livros, artigos e conteúdo acadêmico. Durante o projeto, foram realizadas buscas e pesquisas relevantes relacionadas ao tema, destacando a importância da interdisciplinaridade na abordagem do problema. Os resultados indicaram a necessidade de um documento bem redigido com diagramas e requisitos, a criação de um protótipo visual, a análise de diferentes linguagens e a importância de uma base de dados estruturada. Com base nessas descobertas, foram identificadas melhores práticas e escolhas para o desenvolvimento de um software de qualidade, fornecendo um documento útil para orientar decisões e futuros desenvolvimentos. A aplicação prática dos conhecimentos adquiridos levou a uma compreensão mais profunda do tema e à identificação de soluções inovadoras. Em resumo, este projeto multidisciplinar contribuiu para consolidar conhecimentos teóricos, desenvolver habilidades práticas e compreender de forma aprofundada o tema abordado. Os resultados obtidos oferecem insights significativos para a análise e desenvolvimento de software, com a expectativa de que esses aprendizados possam ser aplicados em contextos futuros para impulsionar o progresso e a melhoria contínua das organizações.

Palavras-chave: projeto integrado multidisciplinar, software de folha de pagamento, metodologia, pesquisas, interdisciplinaridade, documento bem redigido, requisitos, protótipo visual, linguagens, base de dados estruturada, melhores práticas, desenvolvimento de software, conhecimentos teóricos, habilidades práticas, soluções inovadoras, análise, progresso, melhoria contínua.

ABSTRACT

This abstract describes a multidisciplinary integrated project that aimed to develop payroll software for a company. The adopted methodology involved research in several sources, such as websites, books, articles and academic content. During the project, relevant searches and research related to the topic were carried out, highlighting the importance of interdisciplinarity in approaching the problem. The results indicated the need for a well-written document with diagrams and requirements, the creation of a visual prototype, the analysis of different languages and the importance of a structured database. Based on these findings, best practices and choices for developing quality software were identified, providing a useful document to guide decisions and future developments. The practical application of the knowledge acquired led to a deeper understanding of the subject and the identification of innovative solutions. In summary, this multidisciplinary project contributed to consolidate theoretical knowledge, develop practical skills and understand in depth the topic addressed. The results obtained offer significant insights for software analysis and development, with the expectation that these learnings can be applied in future contexts to drive the progress and continuous improvement of organizations.

Keywords: multidisciplinary integrated project, payroll software, methodology, research, interdisciplinarity, well-written document, requirements, visual prototype, languages, structured database, best practices, software development, theoretical knowledge, practical skills, innovative solutions, analysis, progress, continuous improvement.

RESUMO

1. INTRODUÇÃO
2. PROGRAMAÇÃO ORIENTADA A OBJETOS II
3. DESENVOLVIMENTO DE SOFTWARE PARA INTERNET
4. TÓPICOS ESPECIAIS DE PROGRAMAÇÃO ORIENTADA A OBJETOS
5. PROJETO DE SISTEMAS ORIENTADO A OBJETOS
6. GERENCIAMENTO DE PROJETO DE SOFTWARE
7. EMPREENDEDORISMO
8. GESTÃO DA QUALIDADE
9. DESENVOLVIMENTO DO PROJETO
10. CONCLUSÃO
11. REFERÊNCIAS

Pág.
5

1. INTRODUÇÃO

1. INTRODUÇÃO

O desenvolvimento de softwares para gerenciamento de folhas de pagamentos é de grande apoio para a gestão e organização dos valores de uma empresa, faz com que as áreas de Recursos Humanos (RH) e Departamento de Pessoal (DP) tenham uma eficiência maior em seus serviços e os organizam da melhor forma que lhes convém mas com uma maior facilidade. O programa que faz a gerência da folha de pagamento, inclusão de pessoas, registro de cargos dentre outras funcionalidades é de suma importância e somente o pessoal autorizado tem acesso a este controle, que é definido o uso para a versão de desktop do software, já para os outros funcionários é disponibilizado uma versão tanto em web quanto para mobile para que seja possível verificar informações pessoais próprias como por exemplo, o holerite.

1.1 - Objetivo Geral

Fazer a implantação do software desenvolvido para que empresa solicitante tenha melhor gerência dos dados, informações e que possa fazer a folha de pagamento dos funcionários de maneira automática baseada nas informações inseridas, gerenciada pelo setor de Recursos Humanos (RH), que contará com a LGPD (Lei Geral de Proteção de dados) e para os usuários, criptografia em suas senhas.

Objetivos Específicos

- Fazer com que o sistema calcule a folha de pagamento;
- Aplicar na prática as informações obtidas através dos casos de uso e diagramas UML;
- Entregar o sistema nas três plataformas citadas: Desktop, Web e Mobile utilizando linguagens de programação orientadas a objetos;
- Aplicar criptografia como método de segurança no login dos usuários;
- Desenvolver utilizando as linguagens de programação: C#, .NET e Dart;

DISCIPLINAS CONTEMPLADAS

- **Base:** Programação Orientada a Objetos II, Desenvolvimento de Software para Internet, Tópicos Especiais de Programação Orientada a Objetos, Projeto de Sistemas Orientado a Objetos, Gerenciamento de Projetos de Software.
- **Complementar:** Empreendedorismo, Gestão da Qualidade.

Neste projeto devem ser entregues:

Nesta etapa do projeto deve ser entregue a implementação do sistema já especificado **(o sistema é implementado do PIM especificado no semestre anterior)**.












O sistema deve automatizar a folha de pagamentos, e deve possuir total controle das **principais funcionalidades (não precisa ser um sistema que atenda a todas as características de um sistema de mercado completo)**. Devem ser apresentados relatórios parciais, diários e consolidados mensalmente, gráficos, alertas, etc.

O sistema deve ser projetado para ser desenvolvido em no máximo **quatro meses**.

O trabalho escrito deve ser o mesmo do semestre passado **REFATORADO** com as mudanças que venham a ser necessárias após as correções e eventual reanálise.

1. Os integrantes do grupo de PIM devem entregar todos os itens abaixo com as devidas alterações do semestre anterior para esse. A regra de negócios deve ser desenvolvida conforme estipulado pelos envolvidos no projeto.
2. Criar um cenário **bem detalhado (regras do negócio, glossário do sistema, pesquisa de programas no mercado e livros sobre o tema sistema para automatização total da folha de pagamento de uma empresa, análise do mercado)**. Definir e justificar o ciclo de vida de desenvolvimento de software.



3. Descrever os requisitos dos usuários, requisitos funcionais, não-funcionais, e do sistema (**usabilidade, desempenho, capacidade etc.**); 
4. Elaborar o **manual de uso do sistema** para aprovação pelos gestores do Sistema para automatização total da folha de pagamento de uma empresa (empresa que contratou o grupo de PIM);  
5. Elaborar todos os principais modelos de **casos de uso** para cada cenário e criar o arquivo descritivo de cada caso de uso. Cada caso de uso deve ter uma descrição sucinta do seu comportamento, dos fluxos principais, alternativos e de exceção de pré e pós-condições. Identificar relacionamentos de *include*, *extend* e generalização; 
6. Elaborar os **diagramas de classes** de análise (Boundary, Control, Entity). Demonstrar o comportamento dos casos de uso através do **diagrama de sequência** 
7. . **Diagrama de implantação** (definir quantos servidores, banco de dados, estrutura utilizada para o sistema e **como instalar o sistema**); 
8. Anexar o Diagrama ER do banco de dados (nova do hit)  e o dicionário de dados revisar antigo. 
9. Anexar o **script de criação do banco** (usar uma ferramenta) e **scripts de dados iniciais de testes (roteiros de teste)** e homologação do sistema; 
10. Anexar **planilhas de testes** para homologação do sistema, identificando como produzir as **evidências do teste** e as **queries no banco**, para certificar que os dados estão corretos (um mínimo de testes que assegurem a funcionalidade básica do sistema); 
- 11.
12. Elaborar o manual de uso do sistema para treinamento; 

13. Anexar relatórios de gestão para análise de evolução dos negócios, análise de mercado, desempenho dos funcionários (RH) etc. ✗

REQUISITOS DO PROJETO

O PIM deverá ser normalizado de acordo com o guia de normalização de trabalhos acadêmicos (disponível no site da UNIP em: <http://www.unip.br/servicos/biblioteca/guia.aspx>).

O PIM deverá ser “postado” no sistema de trabalhos acadêmicos da UNIP dentro do prazo a ser divulgado em momento oportuno. Se isso não for feito, o grupo será **reprovado**.

Deverá ser entregue ao coordenador do curso, ou à banca avaliadora, a versão em papel e em CD ou pendrive ou qualquer outra mídia de comum acordo com a coordenação e/ou a banca avaliadora, com o projeto documentado, apresentação em powerpoint ou equivalente, arquivos com os diagramas e **códigos fonte completamente comentados** (e anexados no final do trabalho impresso), quando for o caso. **Deve haver um manual de instalação do sistema e a uma planilha de testes para homologação pelos professores avaliadores.**

INSTRUÇÕES PARA O DESENVOLVIMENTO

1. O Projeto PIM deverá ser realizado em grupo, de no máximo 6 alunos.
2. Os grupos deverão comparecer nos dias definidos para os encontros com os professores orientadores do projeto, para que eles avaliem o andamento dele.
3. Os professores orientadores do PIM devem escrever, periodicamente, um breve relato de cada projeto, e da situação de cada componente do grupo, e enviar para o Coordenador do curso. As Fichas de Controle deverão ficar em uma pasta em poder do professor coordenador do curso. No final do semestre as fichas deverão ser arquivadas no prontuário dos alunos.
4. Cada grupo deverá fazer uma apresentação oral do projeto com slides no PowerPoint ou equivalente, conforme programação das bancas avaliadoras, estabelecida pela Coordenação do curso.

AVALIAÇÃO

A nota final do PIM é definida exclusivamente pela banca avaliadora e pelo Coordenador do curso, podendo ser diferente para cada aluno do grupo, conforme a participação no projeto ao longo do semestre e, mais especificamente, na apresentação.

Trabalhos que não atendam aos requisitos fundamentais de linguagem e de tecnologias (orientação a objetos e responsividade) poderão ser reprovados.

Repete-se aqui para ênfase os requisitos essenciais:

- a) **Usará o paradigma da orientação a objetos para o sistema, com interface gráfica para desktop.**
- b) **A aplicação Web poderá ser desenvolvida com o uso da tecnologia ASP.Net com a linguagem C#.**

- c) **A aplicação Mobile será desenvolvida com foco em Android (90%) dos clientes. Porém, poderão ser usadas outras tecnologias tais como .Net Maui, flutter, python (Kivy, KivyMD, ...) etc.**
- d) **O banco de dados utilizado deverá ser o MS SQL Server, hospedado em um servidor Windows Server.**

Obviamente, junto destas tecnologias e linguagens, há itens adicionais, como html, javascript, jquery, bootstrap, sqlite para o mobile etc.

ATIVIDADE DE EXTENSÃO UNIVERSITÁRIA

Extensão universitária são todas as atividades promovidas por instituições de ensino superior destinadas à interação entre ela e a comunidade na qual está inserida, constituindo uma ponte permanente entre a universidade e a sociedade. A extensão universitária pode ser uma atividade prática, acadêmica, técnica ou cultural. Vão desde grupos de estudo a congressos e simpósios, onde são discutidos temas relacionados a área de atuação do estudante e permite que eles tenham contato aprofundado com o dia a dia da profissão. Representam uma opção de aprendizado para que o aluno acompanhe as tendências do mercado de trabalho, aprenda conceitos relacionados às áreas escolhidas e desenvolva habilidades específicas em sua área (ou áreas) de atuação.

Atividade Proposta de Extensão Universitária.

Como atividade de extensão universitária propõe-se que o grupo de PIM realize uma pesquisa sobre os tipos de entidades sem fins lucrativos, elaborando posteriormente uma lista com algumas organizações próximas a sua instituição de ensino. Após a pesquisa o grupo deverá tentar realizar uma visita técnica a pelo menos uma entidade sem fins lucrativos, para elaborar um relatório descrevendo a visita e como a TI pode auxiliar a organização visitada a melhorar seus serviços, finalizando com as conclusões finais do grupo.

Observações gerais

Quem define a regra de negócio é o grupo do PIM, não são os professores orientadores. A regra de negócio será definida em função das pesquisas e interesses do grupo do PIM. Atentem-se que deve haver uma parte visual em desktop, outra em web e outra em mobile, que deverá ser definida pela equipe do PIM considerando que terão cerca de 4 meses para desenvolver os programas. Logo, deve ser previsto algo que seja factível.

No projeto deverá ficar clara a contribuição de cada disciplina, o que será evidenciado pelos artefatos entregues. Não se deve no trabalho explicar a matéria, por exemplo: Um diagrama de classes é etc. Presume-se que o aluno, assim como o avaliador, saiba o que é um diagrama de classes.

No projeto deve ficar claro como o sistema desenvolvido funcionará, o que deve estar contido logo no início, quando se descreve em várias páginas como o negócio funciona.

PLÁGIO

Um trabalho é considerado plágio quando contém trechos copiados de outros trabalhos sem citação da fonte. No Brasil, plágio é considerado crime, pois é uma violação do direito autoral.

Esse tema é de grande preocupação das instituições de ensino, pois, além de colocar a reputação dos autores em risco, pode também colocar a reputação da instituição em uma situação desconfortável.

Em trabalhos acadêmicos, é necessário sempre citar a fonte no corpo do texto, logo em seguida à apresentação da ideia. E no final do trabalho, no espaço destinado às referências, é preciso identificar as obras utilizadas seguindo as normas da ABNT.

Os professores usam um software interno da instituição que compara o trabalho apresentado pelo grupo do PIM com TODOS os trabalhos já postados em semestres anteriores por outros alunos. Caso o percentual de similaridade do trabalho esteja em nível elevado, **isso ocasionará a reprovação do grupo.**

Tipos de plágio

Integral: o plágio integral ocorre quando a obra é copiada na sua totalidade e a fonte não é apresentada.

Parcial: o plágio parcial consiste na utilização de trechos de diversas obras para a criação de novo trabalho.

Conceitual: o plágio conceitual acontece quando uma ideia é reescrita com outras palavras, sem apresentação da autoria original. (MENEZES, Pedro. O que é plágio? Disponível em: <https://www.significados.com.br/plagio/>. Acesso em: 28 jul. 2022).

Não se deve também incorrer na prática de má conduta acadêmica do autoplágio, que consiste na apresentação total ou parcial de textos já publicados pelo mesmo autor, sem as devidas referências aos trabalhos anteriores, ou ainda, a publicação do próprio PIM em sites sem credibilidade acadêmica.

2. PROGRAMAÇÃO ORIENTADA A OBJETOS II

Na programação Orientada a Objetos (POO) possuímos alguns métodos mas tudo se baseia em classes e objetos “Uma classe é um gabarito para a definição de objetos. Através da definição de uma classe, descreve-se que propriedades — ou atributos — o objeto terá.”(RICARTE, 2001, p.3) “Um objeto é um elemento que representa, no domínio da solução, alguma entidade (abstrata ou concreta) do domínio de interesse do problema sob análise. Objetos similares são agrupados em classes.”(RICARTE, 2001, p.5). A partir das classes podemos começar a aplicar seus métodos.

Aplicou-se no desenvolvimento do projeto os métodos definidos como pilares da orientação a objetos (OO), polimorfismo, encapsulamento e herança.

“O Polimorfismo é a capacidade de uma variável se referir em tempo de execução a objetos de diversas classes. Podemos defini-lo também como sendo o nome dado à capacidade que objetos diferentes têm de responder a uma mesma mensagem. Uma mesma mensagem pode apresentar formas de execução diferentes, próprias de cada objeto.”(FARINELLI, 2007, p.23)

“Encapsulamento é o princípio de projeto pelo qual cada componente de um programa deve agregar toda a informação relevante para sua manipulação como uma unidade (uma cápsula). Aliado ao conceito de ocultamento de informação, é um poderoso mecanismo da programação orientada a objetos. Com este mecanismo podemos ocultar detalhes de uma estrutura complexa, que poderiam interferir durante o processo de análise.”(FARINELLI, 2007, p.17)

“Herança Provavelmente herança é o recurso que torna o conceito de classe mais poderoso. Em C++, o termo herança se aplica apenas às classes. Variáveis não podem herdar de outras variáveis e funções não podem herdar de outras funções.(BORGES. p.49)”

3. DESENVOLVIMENTO DE SOFTWARE PARA INTERNET

Apresentar os métodos e tecnologias adotadas para o desenvolvimento de software para a Internet, considerando a composição de um desenvolvimento de sistemas do porte proposto no projeto, **com base em referências bibliográficas devidamente citadas e relacionadas.**

Descrever o processo de utilização desses métodos e tecnologias no suporte às atividades de desenvolvimento de software para a Internet, que trate das questões relacionadas às demandas do presente sistema.

Aplicar as técnicas de desenvolvimento de software para a Internet, para a composição do sistema proposto pelo grupo.

ASP, HTML, CSS, BOOTSTRAP, JQUERY, JAVASCRIPT, Falar sobre teste de usabilidade

Nos capítulos não devem ser inseridas informações referentes ao desenvolvimento do projeto, como por exemplo, códigos em ASP. Devem ser inseridos apenas os itens acima relacionados. Todo o desenvolvimento do projeto, relativo a cada disciplina, como por exemplo a codificação em ASP, deve ser inserido no Capítulo 7 (Desenvolvimento do projeto).

4. TÓPICOS ESPECIAIS DE PROGRAMAÇÃO ORIENTADA A OBJETOS

Apresentar os métodos e tecnologias adotadas para a adoção de tópicos especiais de programação orientada a objetos, considerando a composição de um desenvolvimento de sistemas do porte proposto no projeto, **com base em referências bibliográficas devidamente citadas e relacionadas.**

Descrever o processo de utilização desses métodos e tecnologias no suporte às atividades de utilização de tópicos especiais de programação orientada a objetos, que trate das questões relacionadas às demandas do presente sistema.

Aplicar tópicos especiais de programação orientada a objetos, para a composição do sistema proposto pelo grupo.

C#, POO

Nos capítulos não devem ser inseridas informações referentes ao desenvolvimento do projeto, como por exemplo, códigos em Python. Devem ser inseridos apenas os itens acima relacionados. Todo o desenvolvimento do projeto, relativo a cada disciplina, como por exemplo a codificação em Python, deve ser inserido no Capítulo 7 (Desenvolvimento do projeto).

5. PROJETO DE SISTEMAS ORIENTADO A OBJETOS

Para projetar e implementar um sistema de administração para um pequeno empreendimento, é fundamental seguir práticas de desenvolvimento orientado a objetos.

“Uma entidade complexa e organizada que capta, armazena, processa, fornece, usa e distribui a informação” Robredo (2003, p. 110)

Desenvolver um sistema de administração eficiente e bem estruturado para um pequeno empreendimento requer a aplicação cuidadosa de elementos essenciais e recursos de projeto. Vamos discutir uma estratégia para a utilização desses recursos, justificando suas utilidades, características técnicas, vantagens de uso e perspectivas de evolução tecnológica e de aplicação para a melhoria contínua do sistema proposto.

Levantamento de Requisitos

- Utilidade: O levantamento detalhado de requisitos é fundamental para compreender as necessidades do empreendimento e definir as funcionalidades essenciais para a administração do sistema.
- Características Técnicas: Entrevistas, reuniões e análise de documentos são técnicas comumente utilizadas para coletar e documentar os requisitos.
- Vantagens de Uso: Garante que o sistema atenda às expectativas dos usuários e ofereça funcionalidades relevantes para a administração eficaz do empreendimento.
- Perspectivas de Evolução: A evolução tecnológica pode envolver a aplicação de técnicas de análise de big data para insights de negócios mais precisos e preditivos.

Modelagem de Domínio

- Utilidade: Permite uma visão inicial e estruturada do domínio do problema, identificando as entidades e seus relacionamentos.
- Características Técnicas: Diagramas de classe UML representam de forma gráfica as entidades, atributos e relacionamentos.
- Vantagens de Uso: Facilita a compreensão do domínio, fornecendo uma base sólida para a construção do sistema.
- Perspectivas de Evolução: Integração de técnicas de inteligência artificial para análise e predição de padrões de negócios.

Padrões de Projeto

- Utilidade: Promovem a eficiência, reusabilidade e manutenção do código, seguindo boas práticas de desenvolvimento.
- *Características Técnicas*: Padrões como Singleton, Factory e Observer são amplamente utilizados para resolver problemas recorrentes de design.
- Vantagens de Uso: Facilita o desenvolvimento, aprimora a legibilidade do código e promove a manutenção eficiente do sistema.
- Perspectivas de Evolução: Incorporação de padrões emergentes para otimizar a performance e escalabilidade do sistema.

Diagramas de Caso de Uso

- Utilidade: Identifica os atores e suas interações com o sistema, proporcionando uma visão clara das funcionalidades que o sistema deve oferecer.
- Características Técnicas: Diagramas de caso de uso UML ilustram os casos de uso e as relações com os atores.
- Vantagens de Uso: Facilita a comunicação entre stakeholders e ajuda a definir os requisitos funcionais do sistema.
- Perspectivas de Evolução: Integração de ferramentas de modelagem interativas para colaboração mais eficaz com os interessados.

Diagramas de Sequência:

- Utilidade: Permitem visualizar a interação entre os atores e o sistema, auxiliando na compreensão do fluxo de operações.
- Características Técnicas: Diagramas de sequência UML mostram as trocas de mensagens entre objetos.
- Vantagens de Uso: Facilita a compreensão do comportamento dinâmico do sistema e ajuda na identificação de possíveis gargalos.
- Perspectivas de Evolução: Integração com ferramentas de análise de desempenho para otimização de operações críticas.

Persistência de Dados

- Utilidade: Armazena e gerencia dados essenciais para o sistema, como informações de funcionários, produtos, vendas e finanças.
- Características Técnicas: Bancos de dados relacionais ou NoSQL podem ser escolhidos com base nas necessidades do sistema.
- Vantagens de Uso: Garante integridade e acesso eficiente aos dados, essenciais para a operação do sistema.

- Perspectivas de Evolução: Integração de tecnologias de blockchain para garantir a segurança e a imutabilidade dos dados.

Arquitetura em Camadas

- Utilidade: Promove a separação de responsabilidades, facilitando a manutenção e a evolução do sistema.
- Características Técnicas: A arquitetura MVC é um exemplo comum de arquitetura em camadas utilizada em sistemas orientados a objetos.
- Vantagens de Uso: Melhora a organização do código, permitindo escalabilidade e reutilização de componentes.
- Perspectivas de Evolução: Adoção de arquiteturas mais avançadas, como microservices, para maior flexibilidade e escalabilidade.

Frameworks e Bibliotecas

- Utilidade: Aceleram o desenvolvimento, fornecendo funcionalidades e componentes prontos para uso.
- Características Técnicas: Exemplos incluem Spring (Java), Django (Python) e Laravel (PHP), dependendo da escolha tecnológica.
- Vantagens de Uso: Economiza tempo de desenvolvimento, promove a padronização e oferece soluções testadas.
- Perspectivas de Evolução: Incorporação de frameworks especializados em IA para otimização de funcionalidades inteligentes.

Testes Unitários e de Integração

- Utilidade: Garante a robustez e a confiabilidade das funcionalidades do sistema por meio de testes automatizados.
- Características Técnicas: Utilização de ferramentas de teste e práticas como TDD (Test-Driven Development).
- Vantagens de Uso: Identifica e corrige problemas mais cedo no ciclo de desenvolvimento, reduzindo erros e melhorando a qualidade do software.
- Perspectivas de Evolução: Integração de ferramentas de teste mais avançadas para análise de cobertura e detecção de falhas mais eficazes.

Documentação Adequada

- Utilidade: Fornece informações claras e abrangentes sobre o sistema para facilitar a compreensão e a manutenção futura.
- Características Técnicas: Manuais de uso, descrição da arquitetura, diagramas UML e outros artefatos relevantes.

- Vantagens de Uso: Facilita a integração de novos membros da equipe, auxilia na solução de problemas e promove a transparência.
- Perspectivas de Evolução: Utilização de ferramentas de documentação dinâmica para manter a documentação sempre atualizada.

Revisão e Melhoria Contínua

- Utilidade: Garante que o sistema esteja sempre alinhado com as expectativas dos usuários e as necessidades do empreendimento.
- Características Técnicas: Revisões de código, análises de desempenho e coleta de feedback dos usuários são práticas comuns.
- Vantagens de Uso: Promove a evolução constante do sistema, aprimorando sua eficiência e eficácia ao longo do tempo.
- Perspectivas de Evolução: Incorporação de processos de DevOps para integração contínua e entrega contínua, facilitando atualizações e melhorias frequentes.

Estratégias de Utilização dos Recursos

Classes e Objetos:

- Utilidade: Modelagem das entidades do sistema, representando características (atributos) e comportamentos (métodos).
- Características Técnicas: Encapsulamento, herança e polimorfismo são facilitados com uma correta modelagem de classes.
- Vantagens de Uso:
 - Reutilização de Código: Classes bem definidas podem ser reutilizadas em diferentes partes do sistema.
 - Abstração: Permitem representar de forma abstrata os conceitos do domínio do problema.
- Perspectivas de Evolução:
 - Integração com Inteligência Artificial: Incorporação de capacidades de aprendizado de máquina nas classes para tomada de decisões mais sofisticadas.

Herança:

- Utilidade: Permite a criação de classes mais especializadas a partir de classes genéricas.
- Características Técnicas: Estabelece uma relação "é um" entre classes, possibilitando a reutilização de métodos e atributos.
- Vantagens de Uso:
 - Generalização e Especialização: Facilita a criação de hierarquias de classes, proporcionando uma organização eficiente do código.
- Perspectivas de Evolução:
 - Melhoria na Resolução de Conflitos de Múltipla Herança: Evolução para resolver os desafios associados à múltipla herança.

Polimorfismo:

- Utilidade: Permite que diferentes classes implementem um mesmo método de maneira específica.
- Características Técnicas: Sobrecarga de métodos e substituição de métodos (sobrescrita).
- Vantagens de Uso:
 - Flexibilidade: Facilita a extensibilidade e adaptabilidade do sistema às mudanças.
- Perspectivas de Evolução:
 - Melhoria na Performance: Otimização de mecanismos de polimorfismo para reduzir o overhead de execução.

Interfaces:

- Utilidade: Define contratos que as classes devem seguir, promovendo o cumprimento de padrões.
- Características Técnicas: Contém apenas assinaturas de métodos.
- Vantagens de Uso:
 - Padronização: Facilita a integração de diferentes partes do sistema.
- Perspectivas de Evolução:

- Múltipla Herança de Interface: Permitir a implementação de múltiplas interfaces por uma mesma classe.

A aplicação estratégica desses recursos em um sistema de administração de um pequeno empreendimento resulta em um código mais organizado, reutilizável e flexível. A utilização adequada de classes e objetos permite uma modelagem eficiente do domínio do problema. A herança proporciona uma estrutura hierárquica bem organizada, facilitando a manutenção e a evolução do sistema. O polimorfismo melhora a flexibilidade do código e facilita a implementação de funcionalidades específicas para diferentes contextos. Por fim, as interfaces ajudam a estabelecer contratos, promovendo a padronização e a integração.

Ao aplicar essas estratégias de forma consciente e alinhada ao contexto do sistema proposto, é possível garantir uma arquitetura robusta e de alta qualidade, atendendo às demandas de administração do empreendimento de maneira eficaz e escalável.

Apresentar recursos de projeto de sistemas orientado a objetos que possam ser utilizados como elementos essenciais ao projeto e implementação para um sistema de administração de um pequeno empreendimento, com base em pesquisa bibliográfica, devidamente referenciada.

Utilizando os resultados da pesquisa realizada, propor estratégias para a utilização dos recursos, justificando a utilidade de cada um, suas características técnicas, vantagens de uso e perspectivas de evolução tecnológica e de aplicação para a melhoria da qualidade do sistema proposto.

Descrever a metodologia de projeto utilizada para o desenvolvimento e controle do projeto.

Nos capítulos não devem ser inseridas informações referentes ao desenvolvimento do projeto, como por exemplo, artefatos UML ou a aplicação do Scrum. Devem ser inseridos apenas os itens acima relacionados. Todo o desenvolvimento do projeto, relativo a cada disciplina, como por exemplo artefatos UML ou a aplicação do Scrum, deve ser inserido no Capítulo 7 (Desenvolvimento do projeto).

6. GERENCIAMENTO DE PROJETO DE SOFTWARE

O Project Management Institute (PMI) é uma organização sem fins lucrativos que oferece o Project Management Professional (PMP), uma das certificações mais reconhecidas para gerentes de projeto em todo o mundo. A metodologia PMI é baseada no Guia PMBOK® (Project Management Body of Knowledge), que oferece um conjunto de boas práticas e diretrizes para o gerenciamento de projetos, incluindo projetos de software.

“Gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de cumprir os seus requisitos.”Project Management Institute (2017, p. 46)

A metodologia PMI divide o ciclo de vida do projeto em cinco grupos de processos interconectados;

Iniciação

Definição clara dos objetivos e propósitos do projeto.

Identificação das partes interessadas e avaliação de sua influência no projeto.

Planejamento

Desenvolvimento de um plano de projeto detalhado, incluindo escopo, cronograma, orçamento, comunicações, riscos, recursos, etc.

Criação de uma estratégia de gerenciamento de mudanças.

Execução

Implementação do plano de projeto conforme definido no processo de planejamento.

Coordenar pessoas e recursos para atingir os objetivos do projeto.

Monitoramento e Controle

Acompanhamento e medição do desempenho do projeto para garantir que os objetivos sejam alcançados.

Identificação e correção de desvios em relação ao plano.

Encerramento

Formalização da aceitação do projeto pelo cliente.

Avaliação de lições aprendidas e documentação para futuros projetos.

Utilização da Metodologia PMI no Suporte às Atividades de Gerenciamento de Projeto de Software

No contexto de um projeto de software, a metodologia PMI aplicada-se da seguinte forma;

Iniciação

Identificar claramente os objetivos do software a ser desenvolvido, suas funcionalidades e restrições.

Envolvimento dos stakeholders para entender suas necessidades e expectativas em relação ao sistema.

Planejamento

Elaborar um plano de projeto detalhado, considerando o escopo, recursos necessários, prazos e orçamento.

Estabelecer uma estratégia de gerenciamento de riscos específica para projetos de software.

Execução

Desenvolver o software conforme definido nas fases de planejamento, seguindo as práticas e padrões de qualidade definidos.

Realizar testes para garantir a funcionalidade e a qualidade do software.

Monitoramento e Controle

Acompanhar o progresso do desenvolvimento, comparando com o planejado.

Realizar revisões e auditorias para garantir a conformidade com os requisitos e padrões estabelecidos.

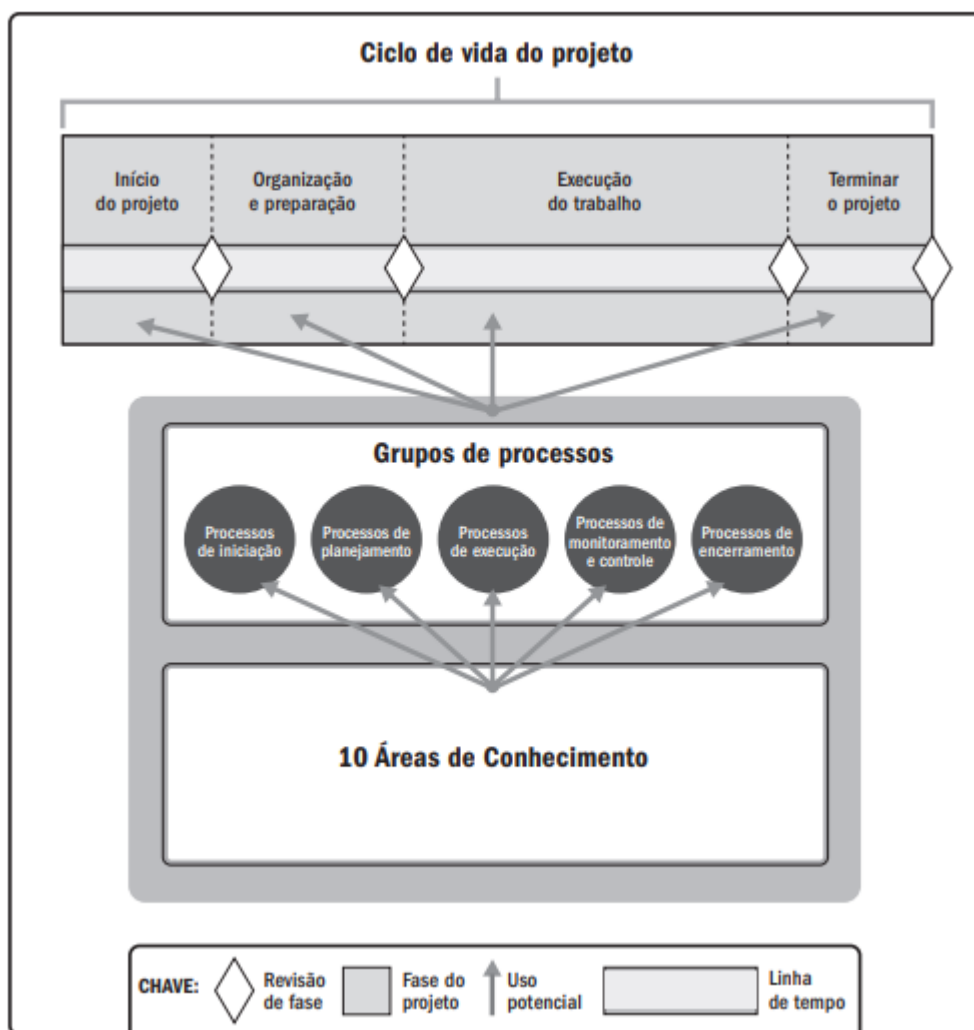
Encerramento

Formalizar a aceitação do software pelo cliente e garantir que todos os requisitos foram atendidos.

Documentar as lições aprendidas para futuros projetos de software.

Com esses procedimentos se tem uma visão uniforme do ciclo do projeto, Conforme a figura elucidada.

Figura - Ciclo de vida do projeto PMBOK.



Fonte: Project Management Institute Autor Project Management Institute, 2017

Ganhos com a Utilização da Metodologia PMI no Gerenciamento de Projetos de Software

A adoção da metodologia PMI no gerenciamento de projetos de software proporciona diversos benefícios, incluindo

Maior Eficiência

O planejamento detalhado ajuda a otimizar o uso de recursos e a evitar atrasos.

Controle Eficaz

Processos de monitoramento e controle permitem identificar desvios rapidamente e tomar ações corretivas.

Qualidade Aprimorada

A aplicação de práticas de gerenciamento de qualidade resulta em software mais confiável e alinhado com as expectativas dos stakeholders.

Satisfação do Cliente

A gestão de expectativas e o envolvimento constante do cliente contribuem para a entrega de um produto que atenda às suas necessidades.

Melhoria Contínua

As lições aprendidas são documentadas e incorporadas em futuros projetos, promovendo a evolução contínua dos processos e práticas.

Apresentar métodos de gerenciamento de um projeto de software, em especial adotando a Metodologia PMI (Project Management Institute®), com base em referências bibliográficas devidamente citadas e relacionadas.

Descrever o processo de utilização dessa metodologia no suporte às atividades de gerenciamento de projeto de software, que trate das questões relacionadas às demandas do presente sistema.

Demonstrar os ganhos que o sistema e as pessoas terão ao fazerem uso dessa metodologia, de tal maneira a melhorar o gerenciamento do projeto de sistema proposto.

Nos capítulos não devem ser inseridas informações referentes ao desenvolvimento do projeto, como por exemplo, aplicação de princípios do PMI. Devem ser inseridos apenas os itens acima relacionados. Todo o desenvolvimento do projeto, relativo a cada disciplina, como por exemplo a aplicação de princípios do PMI, deve ser inserido no Capítulo 7 (Desenvolvimento do projeto).

7. EMPREENDEDORISMO

Com base em pesquisa bibliográfica devidamente referenciada, caracterizar o empreendedorismo e a visão empreendedora como essencial em todas as atividades para as quais sejam necessárias ações para a garantia de resultados em um empreendimento.

Relacionar os principais elementos que compõem um Plano de Negócios, tendo como referência de negócio o projeto de sistema proposto.

Apresentar os elementos que podem contribuir com a viabilidade do sistema proposto no mercado, constituindo essa apresentação com, no mínimo, os seguintes elementos:

- Objetivo principal (missão).
- Objetivos de curto, médio e de longo prazo (visão – versões do sistema).
- Público-alvo (localização, caracterização – recursos financeiros, demanda etc).
- Principais concorrentes e as estratégias principais para vencê-los.
- Estratégia de comercialização (licenças, venda, suporte, pós-venda etc).

Nos capítulos não devem ser inseridas informações referentes ao desenvolvimento do projeto, como por exemplo, o Plano de Negócios. Devem ser inseridos apenas os itens acima relacionados. Todo o desenvolvimento do projeto, relativo a cada disciplina, como por exemplo o Plano de Negócios, deve ser inserido no Capítulo 7 (Desenvolvimento do projeto).

8. GESTÃO DA QUALIDADE

Caracterizar, com base em pesquisa bibliográfica devidamente referenciada, os aspectos principais da gestão da qualidade aplicados ao desenvolvimento de sistemas informatizados.

Realizar levantamento dos impactos tecnológicos e operacionais para um sistema informatizado, decorrentes da utilização eficiente de uma ou mais ferramentas da qualidade, para a composição de indicadores de mensuração e de métricas para a avaliação da qualidade do software desenvolvido.

Aplicar pelo menos uma ferramenta de gestão da qualidade no sistema desenvolvido, apresentando suas características principais e eventuais impactos na qualidade do sistema.

Nos capítulos não devem ser inseridas informações referentes ao desenvolvimento do projeto, como por exemplo, requisitos de qualidade. Devem ser inseridos apenas os itens acima relacionados. Todo o desenvolvimento do projeto, relativo a cada disciplina, como por exemplo os requisitos de qualidade, deve ser inserido no Capítulo 7 (Desenvolvimento do projeto).

9. DESENVOLVIMENTO DO PROJETO

O Projeto Integrado Multidisciplinar (PIM) visou automatizar a folha de pagamento, integrando conhecimentos de diversas disciplinas. Foram utilizadas técnicas de Engenharia de Software, Análise e Programação Orientada a Objetos, Projeto de Interface com o Usuário, Banco de Dados, Economia e Mercado, e Gestão Estratégica de Recursos Humanos.

Na análise de Sistemas Orientada a Objetos, a linguagem UML foi empregada para a análise e desenvolvimento do sistema. A escolha se baseou na capacidade da UML de auxiliar no levantamento de requisitos e modelagem. As vantagens incluem ampla aceitação de mercado, integração com Orientação a Objetos e disponibilidade de ferramentas gratuitas.

Para a Programação Orientada a Objetos, estrategicamente usaram-se C#, ASP.net com C# e Flutter, cada uma escolhida pelas suas vantagens específicas. O Microsoft SQL Server foi adotado para o banco de dados, fundamentado na amplitude de operações que possibilita e sua ampla aceitação no mercado.

Na disciplina de Projeto de Interface com o Usuário, utilizou-se a ferramenta Figma para desenvolver uma interface de alta qualidade, visando aprimorar a usabilidade e eficiência do trabalho.

Na disciplina de Banco de Dados, escolheu-se o Microsoft SQL Server devido às operações abrangentes que oferece e à estrutura tabular dos bancos de dados relacionais.

Na disciplina de Economia e Mercado, a automação da folha de pagamento foi proposta para obter benefícios econômicos e de mercado, incluindo redução de custos e aumento da eficiência.

Na disciplina de Gestão Estratégica de Recursos Humanos, a automação da folha de pagamento e a interoperabilidade entre Recursos Humanos e Departamento Pessoal foram enfatizadas para trazer eficiência, inovação e economia de custos.

O projeto abrangeu diversas disciplinas, utilizou modelos e padrões estabelecidos, priorizou a viabilidade econômica e fez uso de tecnologias disponíveis. A solução pode ser aprimorada para incluir operações adicionais, integração com sistemas de gestão de RH e customizações para diferentes regras de negócio.

2.1 - Manual do Sistema Desktop

1. Como cadastrar um funcionário ?
 - a. Acesse o item do Menu Funcionários
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, o novo funcionário aparecerá na tabela de funcionários

2. Como cadastrar os estabelecimentos existentes ?
 - a. Acesse o item do Menu Tabelas -> Submenu Estabelecimento
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, o novo estabelecimento aparecerá na tabela de estabelecimentos

3. Como cadastrar os cargos existentes na Empresa ?
 - a. Acesse o item do Menu Tabelas -> Submenu Cargo
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, o novo cargo aparecerá na tabela de cargos

4. Como cadastrar um Perfil de acesso ?
 - a. Acesse o item do Menu Tabelas -> Submenu Perfil
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, o novo perfil aparecerá na tabela de perfis

5. Como cadastrar as operações ?
 - a. Acesse o item do Menu Tabelas -> Submenu operações
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, a nova operação aparecerá na tabela de operações
 - e. Caso a operação que esteja precisando não esteja no seletor, realize o cadastro dessa operação no item do Menu Tabelas -> Submenu operações disponíveis.

- f. Clique em novo
 - g. Preencha todos os campos obrigatórios
 - h. Clique em salvar
6. Como vincular as operações a um funcionário ?
- a. Acesse o item do Menu Operações do funcionário
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, a operação vinculada a um funcionário aparecerá na tabela.
7. Como vincular os cargos aos funcionários ?
- a. Acesse o item do Menu Cargos do funcionário
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, o cargo vinculado a um funcionário aparecerá na tabela.
8. Como vincular os estabelecimentos aos funcionários ?
- a. Acesse o item do Menu Estabelecimentos do funcionário
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, o estabelecimento vinculado a um funcionário aparecerá na tabela.
9. Como cadastrar um Holerite ?
- a. Acesse o item do Menu Holerite
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, o Holerite cadastrado aparecerá na tabela.
10. Como cadastrar a Folha de Pagamento ?
- a. Acesse o item do Menu Folha de Pagamento
 - b. Clique em novo
 - c. Preencha todos os campos obrigatórios
 - d. Clique em salvar, a Folha de Pagamento cadastrada aparecerá na tabela.

11. Como visualizar informações antes cadastradas

- a. Nas tabelas de cada tabela existe uma coluna de ações
- b. Clique no botão com ícone de olho para visualizar todas as informações daquela linha da tabela

12. Como visualizar o Perfil

- a. Acesse o menu Perfil
- b. Clique em Visualizar

2.2 - Manual do Sistema Web e Sistema Mobile

1. Como visualizar os seus holerites ?

- a. Acesse o menu holerites
- b. Visualize ou busque o item na tabela
- c. Clique no botão com ícone de olho

2. Como gerar um pdf do holerite ?

- a. Ao visualizar um item
- b. Aperte o botão gerar pdf

3. Como visualizar as suas operações ?

- a. Acesse o menu operações
- b. Visualize ou busque o item na tabela

4. Como visualizar o seu perfil e alterar sua senha ?

- a. Acesse o menu perfil
- b. Caso queira alterar a sua senha, preencha os campos e clique em salvar

2.3 - Glossário do Sistema

- **CBO:** Classificação Brasileira de Ocupações. É um sistema que classifica e identifica as ocupações profissionais no Brasil, utilizado para diversos fins, como estatísticas de emprego e contribuição previdenciária.

- **Periculosidade:** É um adicional pago ao trabalhador que exerce atividades consideradas perigosas, com riscos à integridade física e à saúde. É regulamentado por lei e varia de acordo com a atividade exercida.
- **Salário Contribuição INSS:** É o valor sobre o qual é calculada a contribuição previdenciária do trabalhador ao INSS (Instituto Nacional do Seguro Social), com base na sua remuneração mensal.
- **Base de Cálculo FGTS:** É o valor sobre o qual é calculado o Fundo de Garantia do Tempo de Serviço (FGTS), um benefício trabalhista no Brasil. A base de cálculo geralmente corresponde ao salário bruto do funcionário.
- **Base de Cálculo IRF:** É o valor sobre o qual é calculado o Imposto de Renda na Fonte (IRF), que é retido diretamente do salário do trabalhador. A base de cálculo geralmente corresponde ao salário bruto do funcionário, com deduções legais.
- **Faixa IRF:** Refere-se às diferentes faixas de alíquota do Imposto de Renda na Fonte. O IRF é calculado de acordo com a renda do trabalhador e as faixas estabelecidas pela legislação tributária.
- **Login:** É um processo de autenticação em um sistema ou plataforma online, onde o usuário fornece suas credenciais (como nome de usuário e senha) para acessar sua conta pessoal.
- **Perfil:** Refere-se às informações e configurações associadas a uma conta de usuário em um sistema ou plataforma. O perfil pode incluir dados pessoais, preferências, permissões de acesso e outras informações relevantes.
- **Folha de Pagamento:** É o documento ou registro contendo informações sobre os salários, benefícios, descontos e demais pagamentos relacionados aos funcionários de uma empresa em um determinado período. É utilizado para calcular os valores a serem pagos aos funcionários e cumprir as obrigações legais.
- **Holerite:** É um documento ou comprovante fornecido ao funcionário pela empresa, contendo informações detalhadas sobre o seu salário, incluindo os rendimentos, descontos, benefícios e demais valores relacionados.
- **Operações do Funcionário:** Refere-se às atividades e transações relacionadas a um funcionário específico, como registro de ponto, solicitação de férias, atualização de dados pessoais, entre outras ações que envolvem a gestão e administração do colaborador na empresa.

- **Cargos do Funcionário:** São as posições ou funções ocupadas por um funcionário dentro de uma organização. Os cargos podem variar em níveis de responsabilidade, habilidades necessárias e remuneração.
- **Estabelecimento do Funcionário:** Refere-se ao local físico onde o funcionário está alocado ou lotado, seja uma filial, escritório, fábrica, loja, entre outros tipos de estabelecimentos da empresa.
- **Funcionários:** São as pessoas que trabalham para uma organização em regime de emprego, realizando atividades e funções definidas pelo empregador.
- **Tabelas:** São conjuntos de dados organizados em formato de tabela, contendo informações específicas relacionadas a um determinado contexto. No contexto mencionado, pode-se referir a tabelas utilizadas para cálculos e registros relacionados à folha de pagamento, impostos, contribuições e demais informações pertinentes à área de recursos humanos e administração de pessoal.

2.4 - Regras de Negócios

1. **Registro de ponto:** é obrigatório o registro de ponto de todos os colaboradores que trabalham mais de 8 horas por dia ou 44 horas por semana.
2. **Horas extras:** as horas extras devem ser pagas sempre no mês seguinte ao trabalhado. Não pode exceder até 2 horas de trabalho por dia ou 10 horas semanais extras.
3. **Salário mínimo:** o salário mínimo é definido pelo governo federal brasileiro (2023: R\$ 1.302,00). a empresa deve pagar pelo menos o salário mínimo aos seus colaboradores.
4. **Férias:** os colaboradores têm direito a férias remuneradas após 12 meses de trabalho. As férias devem durar pelo menos 30 dias corridos e devem ser pagas com um acréscimo de 1/3 do salário.
5. **Décimo terceiro salário:** os colaboradores têm direito a receber um salário extra no final do ano, conhecido como décimo terceiro salário. O valor é calculado com base no salário mensal e no período trabalhado durante o ano.
6. **Vale-transporte:** as empresas são obrigadas a fornecer vale-transporte para seus empregados, que devem ser suficientes para cobrir o custo do deslocamento de casa para o trabalho e vice-versa.
7. **Contribuições previdenciárias:** as empresas são obrigadas a descontar a contribuição previdenciária do salário de seus empregados e a fazer uma contribuição para a previdência social.

8. **FGTS:** as empresas devem depositar uma parcela do salário de seus empregados em uma conta do Fundo de Garantia do Tempo de Serviço (FGTS), que pode ser sacado pelo empregado em situações específicas, como demissão sem justa causa ou aposentadoria.

9. **Descontos em folha:** as empresas podem descontar do salário dos empregados valores referentes a planos de saúde, alimentação e outros benefícios que o empregado tenha aderido.

10. **Adicional noturno:** os empregados que trabalham durante a noite têm direito a receber um adicional noturno, que é calculado com base no valor da hora normal de trabalho.

11. **Licença-maternidade e licença-paternidade:** as colaboradoras gestantes têm direito a licença-maternidade remunerada de 120 dias, enquanto os empregados pais têm direito a licença-paternidade remunerada de 5 dias úteis.

12. **Vale-alimentação:** as empresas podem fornecer vale-alimentação para seus colaboradores, que é um benefício não obrigatório e pode ser descontado do salário do colaborador.

13. **Plano de saúde:** as empresas podem oferecer plano de saúde para seus colaboradores, que pode ser pago integralmente pela empresa ou compartilhado entre a empresa e o colaborador.

Adicional de periculosidade: os colaboradores que trabalham em atividades que envolvem riscos à sua saúde ou integridade física têm direito a um adicional de periculosidade, que é um valor adicional pago sobre o salário.

Telas e Funções Gerais:

1. Visualização das Informações em Tabelas:

- Os usuários devem ser capazes de visualizar informações em tabelas de forma organizada e legível.
- As tabelas devem apresentar dados relevantes, como funcionários, cargos, operações, etc.

2. Campo de Busca:

- Cada tela de visualização deve conter um campo de busca para permitir aos usuários encontrar informações específicas nas tabelas.
- O campo de busca deve realizar buscas em tempo real enquanto o usuário digita.

3. Tela de Login:

- Uma tela de login segura deve ser fornecida para autenticação dos usuários.
 - Deve haver campos para inserir nome de usuário (ou email) e senha.
4. Menu de Acesso:
- Deve existir um menu de acesso que varie de acordo com o perfil do usuário.
 - O menu deve exibir opções de acordo com as permissões atribuídas ao perfil.
5. CRUD para Perfil:
- Os administradores devem poder criar, ler, atualizar e excluir perfis de usuário.
6. CRUD para Estabelecimento:
- Os administradores devem poder criar, ler, atualizar e excluir informações sobre estabelecimentos.
7. CRUD para Cargo:
- Os administradores devem poder criar, ler, atualizar e excluir informações sobre cargos.
8. CRUD para Operações Disponíveis:
- Os administradores devem poder gerenciar as operações disponíveis, que podem ser atribuídas aos funcionários.
9. CRUD para Operações:
- Os administradores devem poder criar, ler, atualizar e excluir informações sobre operações específicas.
10. CRUD para Funcionários:
- Os administradores devem poder criar, ler, atualizar e excluir informações sobre funcionários.
11. CRUD para Vincular Cargos a Funcionários:
- Os administradores devem poder associar cargos a funcionários específicos.
12. CRUD para Vincular Operações a Funcionários:
- Os administradores devem poder atribuir operações específicas a funcionários.
13. CRUD para Geração de Holerites:
- Os administradores devem poder criar, ler, atualizar e excluir holerites para os funcionários.
14. CRUD para Geração da Folha de Pagamento:
- Os administradores devem poder gerar a folha de pagamento com base nos dados dos funcionários e seus holerites.
15. Dispositivos Web e Mobile para Colaboradores:

- Os colaboradores devem ser capazes de acessar seus holerites por meio de dispositivos web e mobile.
- Eles devem poder gerar os holerites em formato PDF.
- Os colaboradores devem ser capazes de visualizar as operações vinculadas a eles e informações sobre seus perfis.

Requisitos Não Funcionais:

16. Segurança:

- O sistema deve implementar autenticação segura para proteger os dados sensíveis.
- Deve haver níveis de acesso diferentes, garantindo que os usuários vejam apenas as informações relevantes para suas funções.

17. Desempenho:

- O sistema deve ser responsivo, garantindo tempos de carregamento rápidos para as diferentes telas.
- Consultas às informações devem ser eficientes para evitar atrasos.

18. Usabilidade:

- A interface do usuário deve ser intuitiva e fácil de usar, mesmo para usuários não técnicos.
- As operações devem ser autoexplicativas e minimizar a possibilidade de erros.

19. Escalabilidade:

- O sistema deve ser projetado para lidar com um aumento no número de funcionários, cargos, estabelecimentos, etc.

20. Compatibilidade:

- A interface web deve ser compatível com uma variedade de navegadores modernos.
- A aplicação móvel deve ser desenvolvida para plataformas populares (iOS e Android).

21. Documentação:

- Deve haver documentação detalhada que explique como usar cada funcionalidade do sistema.
- Um manual passo a passo deve ser fornecido para auxiliar os usuários em suas tarefas.

22. Confiabilidade:

- O sistema deve ser confiável e minimizar a ocorrência de erros, especialmente ao gerar folhas de pagamento e holerites.

23. Privacidade:

- Os dados dos funcionários e suas informações pessoais devem ser tratados com confidencialidade e em conformidade com regulamentações de privacidade.

2.8 - Requisitos dos usuários

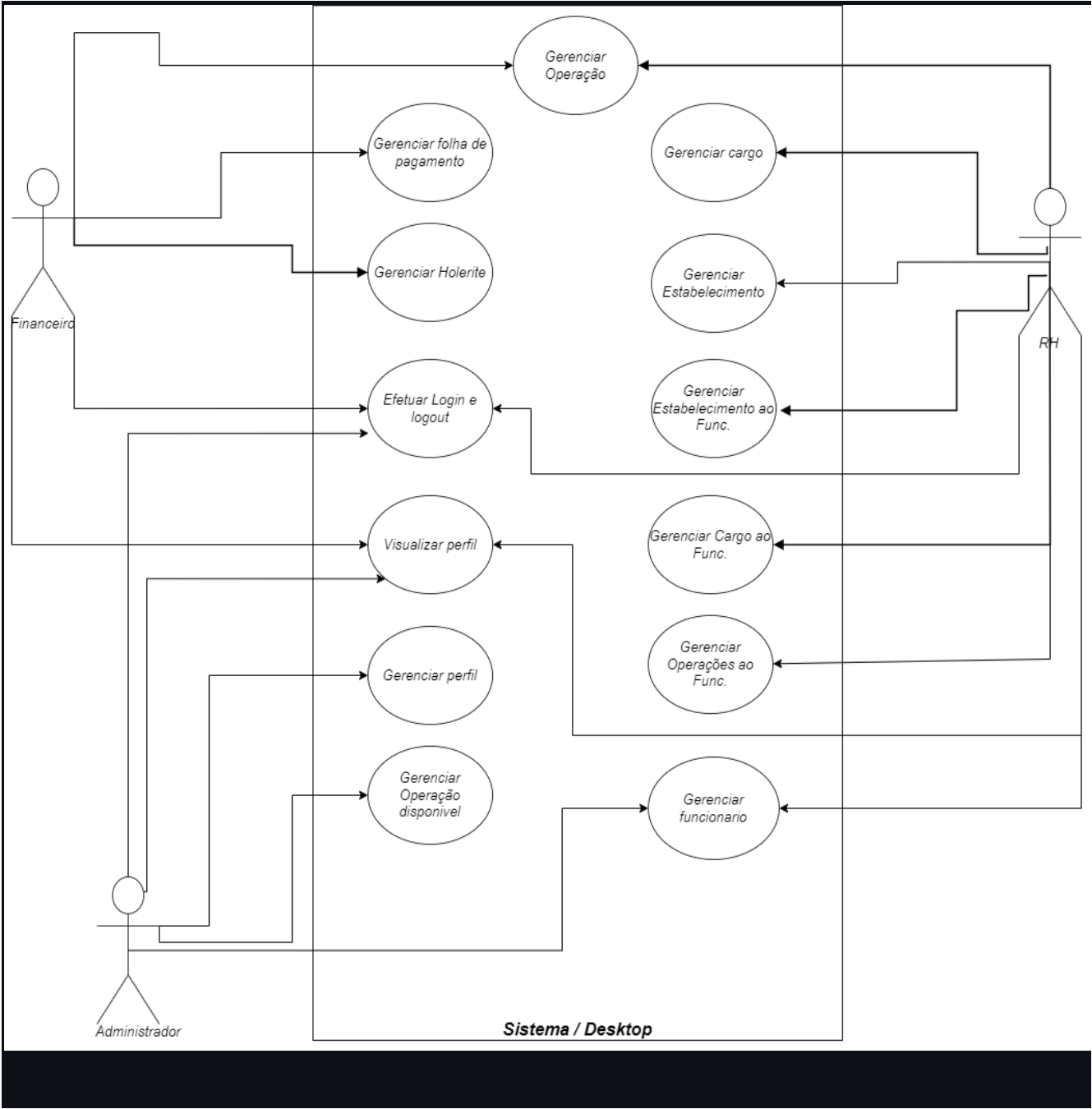
- O software realiza cálculos precisos dos salários com base nas informações fornecidas, incluindo horas trabalhadas, benefícios, deduções, impostos e quaisquer outras variáveis relevantes.
- O software permite a criação de perfis individuais para cada funcionário, onde informações pessoais, dados de contato, benefícios, deduções e outras informações relacionadas possam ser armazenadas e gerenciadas de forma segura.
- O software tem a capacidade de no futuro gerar relatórios precisos e personalizáveis, como holerites, demonstrativos de pagamento, resumos de impostos retidos, guias de recolhimento, entre outros. Além disso, deverá permitir a geração de declarações obrigatórias, como a Declaração do Imposto de Renda Retido na Fonte (DIRF) e a Relação Anual de Informações Sociais (RAIS).
- O software de folha de pagamento se integrará a outros sistemas internos da organização, como sistemas de gestão de recursos humanos (RH), sistemas de ponto eletrônico ou sistemas de contabilidade, a fim de automatizar a transferência de dados e melhorar a eficiência do processo.
- o software cumpra as regulamentações de segurança de dados e privacidade, garantindo a proteção das informações confidenciais dos funcionários. Além disso, é importante que o software esteja em conformidade com as leis e regulamentações trabalhistas e fiscais aplicáveis.

2.9 - Requisitos do sistema

- Interface intuitiva e fácil de usar, com fluxos de trabalho claros e navegabilidade simplificada. (Usabilidade)
- Tempo de resposta rápido para evitar atrasos significativos durante o processamento de cálculos e geração de relatórios. Capacidade de processar grandes volumes de dados de forma eficiente, especialmente em organizações com um grande número de funcionários. Otimização de algoritmos e processos para minimizar o consumo de recursos do sistema, como CPU e memória. (Desempenho)
- Capacidade de lidar com a complexidade das leis e regulamentações trabalhistas e fiscais. Suporte a múltiplas empresas ou filiais em uma única instância do software. Escalabilidade para acomodar o crescimento futuro da organização, seja em termos de número de funcionários, volume de dados ou necessidades adicionais de processamento. (Capacidade)
- Manutenção da integridade dos dados, garantindo que as informações armazenadas e processadas sejam precisas e consistentes. Implementação de medidas de segurança robustas para proteger as informações confidenciais dos funcionários, como dados pessoais, registros financeiros e detalhes de pagamento. (Confiabilidade e Segurança)
- Capacidade de se integrar com outros sistemas relevantes, como sistemas de gestão de recursos humanos (RH), sistemas de ponto eletrônico, sistemas contábeis e sistemas de pagamento, para facilitar a troca de dados e evitar redundâncias de entrada de dados. (Integração)

Diagramas de caso de Uso

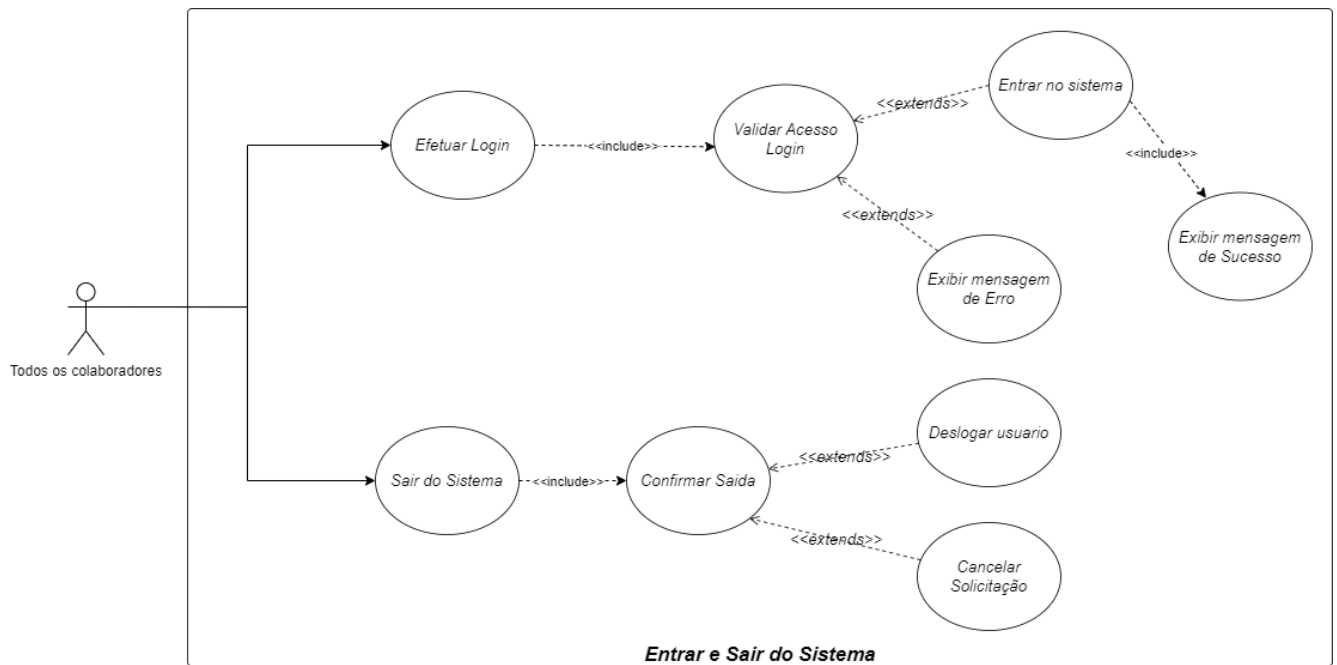
Figura 1 - Diagrama de Caso de Uso do Sistema Desktop



Fonte: Autoral (2023)

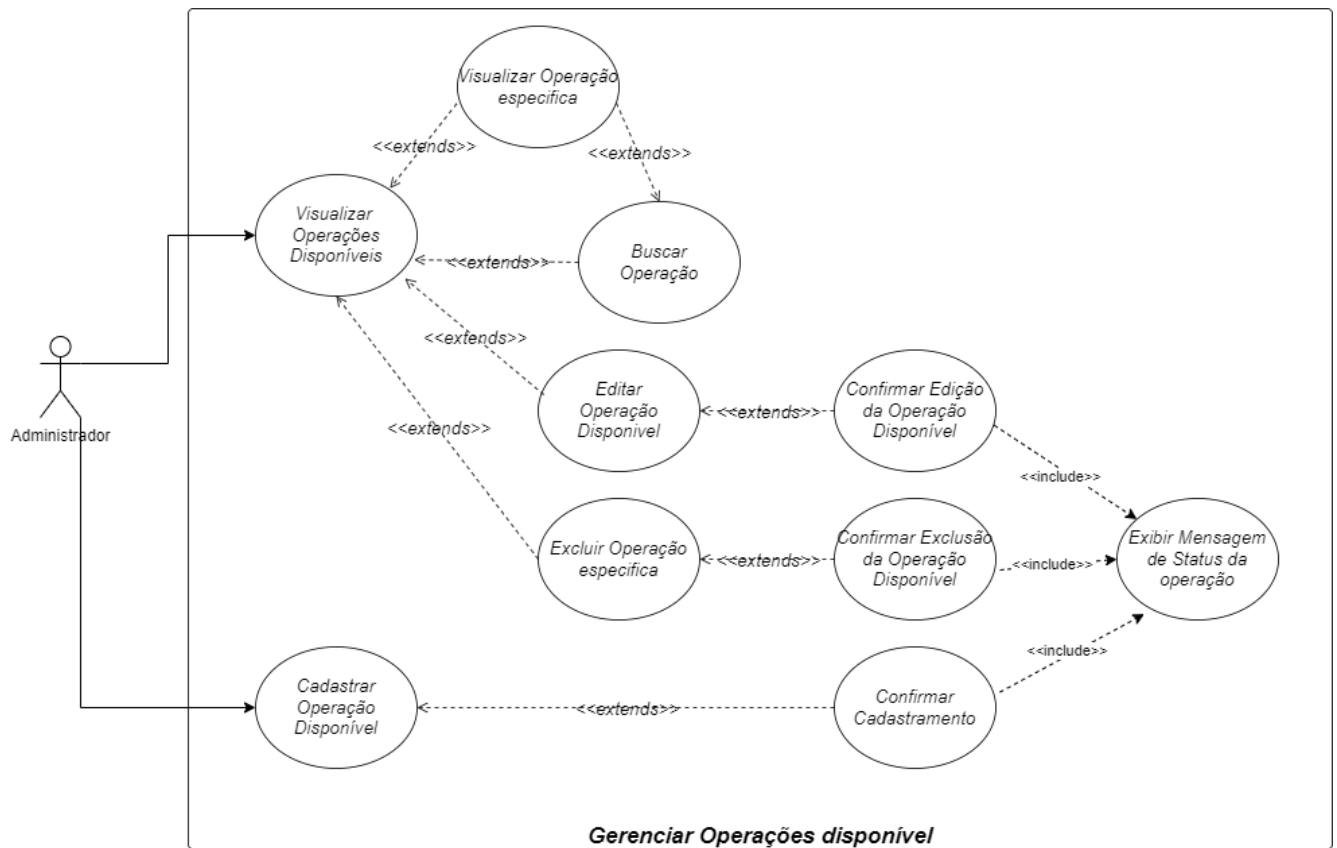
Caso de uso específico do Desktop

Figura 2 - Diagrama de Caso de Uso Desktop específico



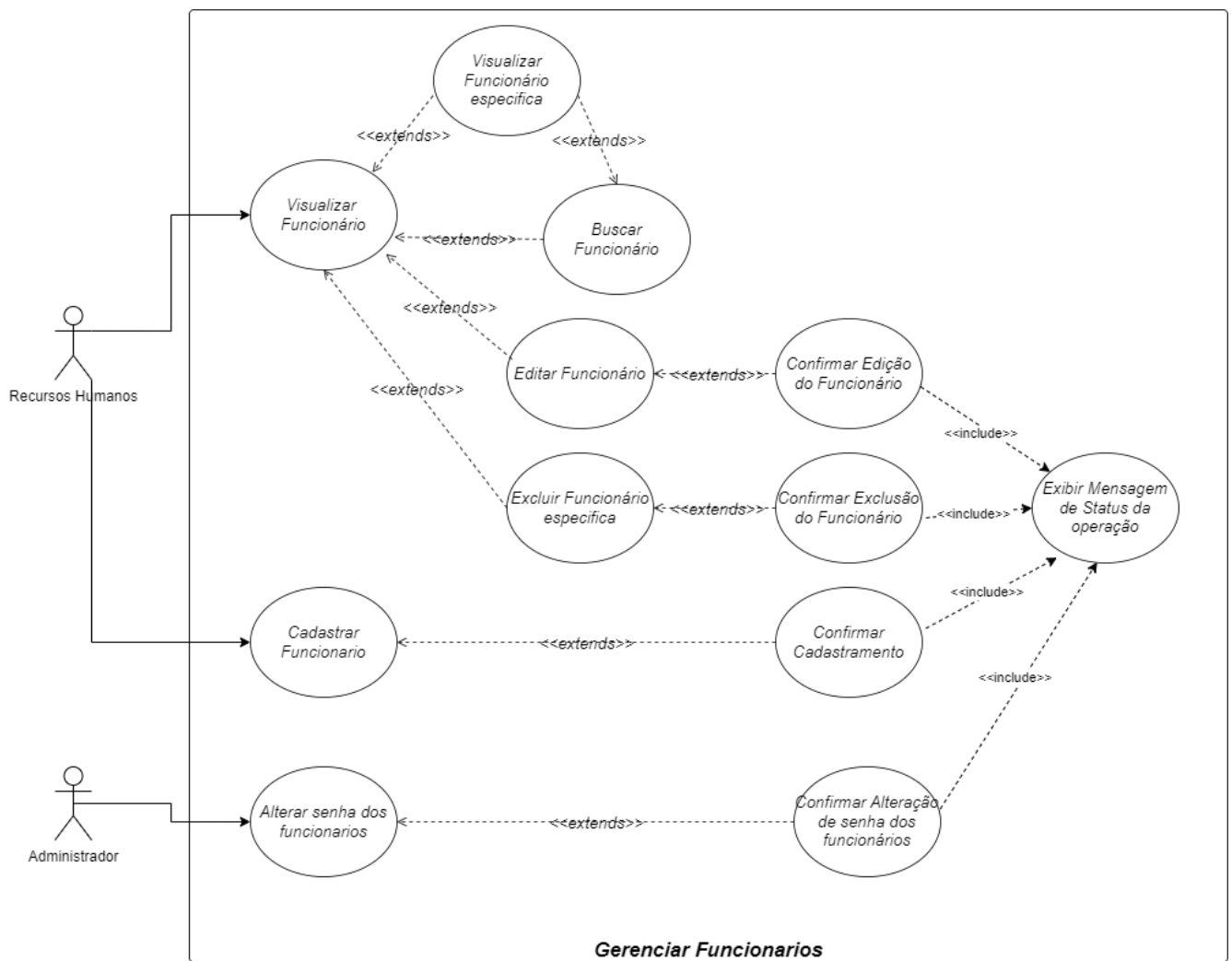
Fonte: Autoral (2023)

Figura 3 - Diagrama de Caso de Uso Desktop específico



Fonte: Autoral (2023)

Figura 4 - Diagrama de Caso de Uso Desktop específico



Fonte: Autoral (2023)

Caso de uso Mobile / Web

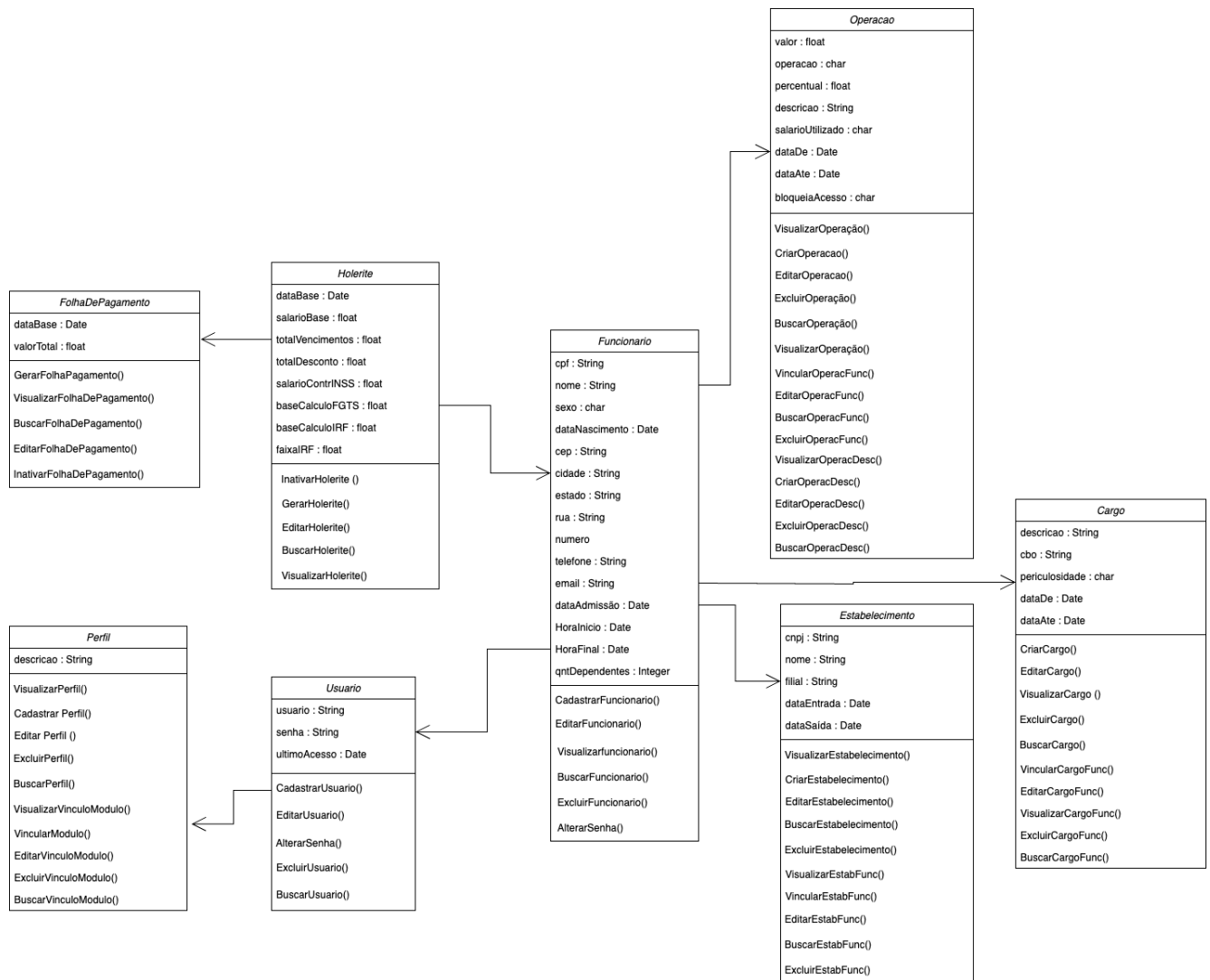
Figura 5 - Diagrama de Caso de Uso Mobile / Web Geral



Fonte: Autoral (2023)

Diagramas de classe

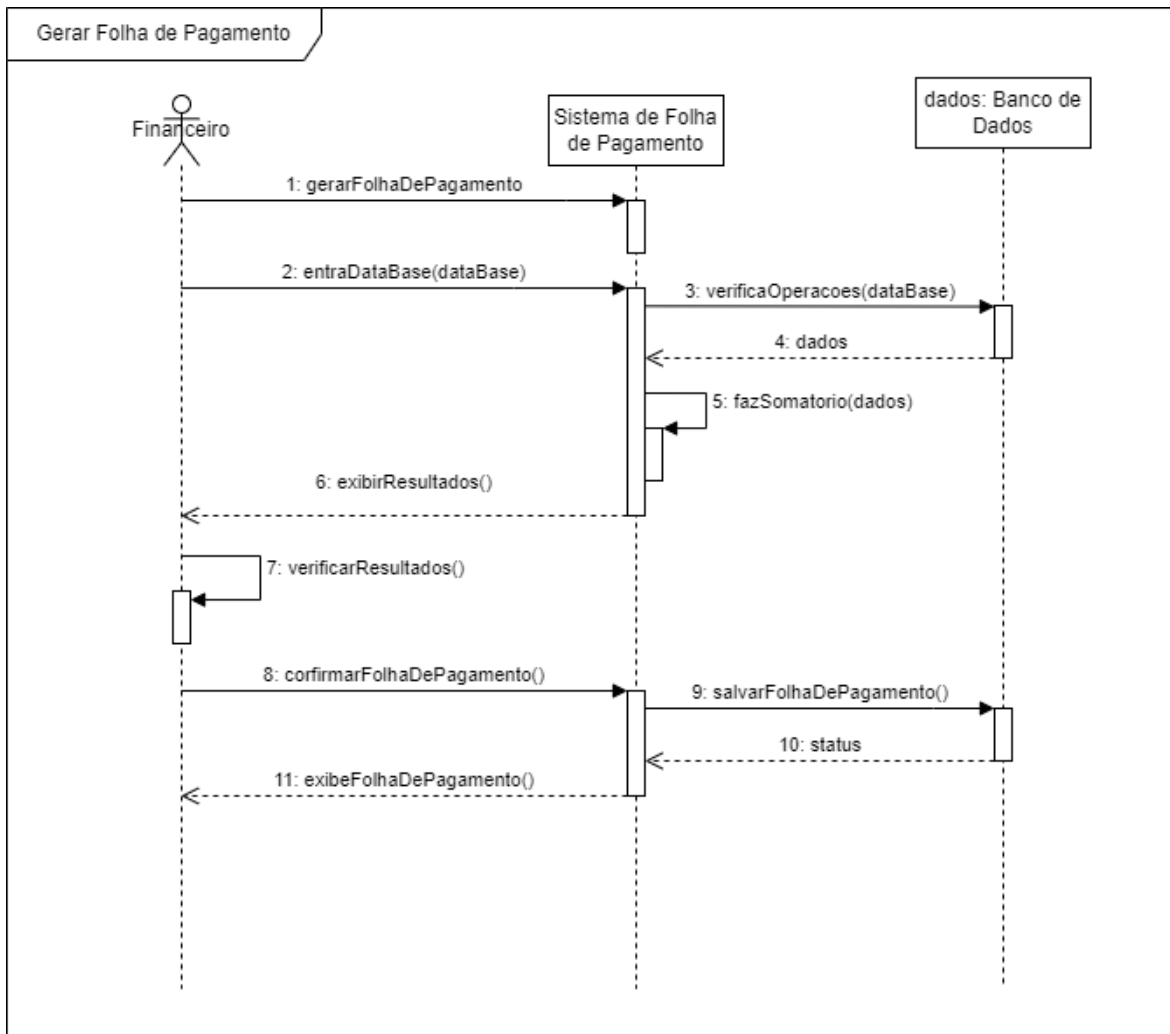
Figura 6 - Diagrama de Classe



Fonte: Autoral (2023)

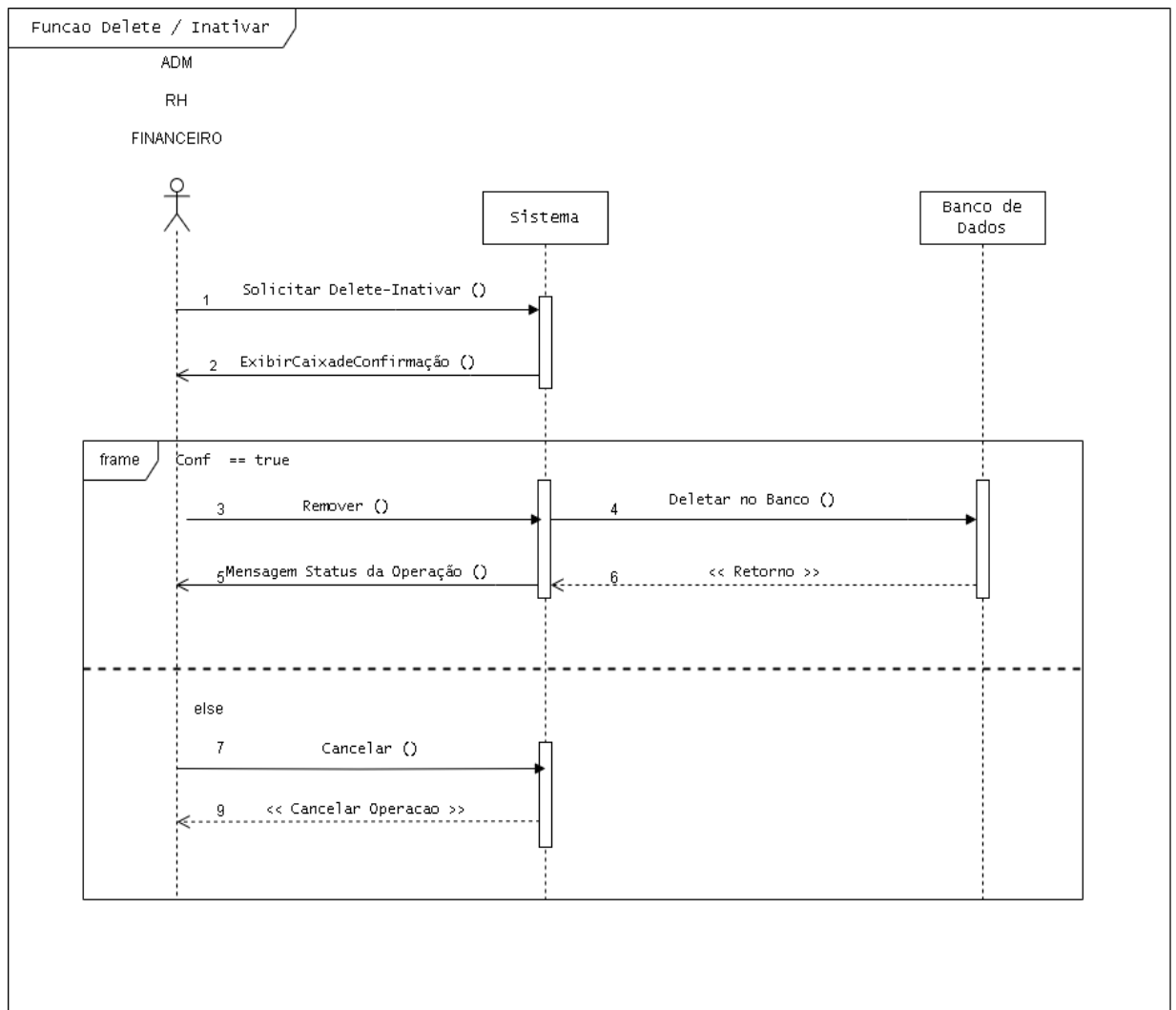
Diagramas de sequência

Figura 7 - Diagrama de sequência Gerar folha de pagamento.



Fonte: Autoral (2023)

Figura - Diagrama de sequência Função Delete / Inativar



Fonte: Autoral (2023)

2.10 - Padrões de Software

Os padrões de software que utilizaremos para o desenvolvimento do sistema de acordo com sua linguagem serão:

1. C# e .NET:

- Padrão de arquitetura MVC (Model-View-Controller): Este é um padrão amplamente utilizado para o desenvolvimento de aplicativos web e desktop com C# e .NET. Ele separa a aplicação em três componentes principais: o Modelo (Model), a Visualização (View) e o Controlador (Controller). O Modelo representa os dados e a lógica de negócios, a Visualização lida com a apresentação dos dados ao usuário e o Controlador gerencia as interações entre o Modelo e a Visualização.
- Padrão de Injeção de Dependência: Esse padrão promove a separação de responsabilidades e a modularidade do código. Ele permite que as dependências de um componente sejam injetadas por meio de interfaces, facilitando a substituição de implementações e a realização de testes unitários.

2. Flutter:

- Padrão de arquitetura BLoC (Business Logic Component): O BLoC é um padrão de gerenciamento de estado utilizado no desenvolvimento de aplicativos Flutter. Ele separa a lógica de negócios da interface do usuário, utilizando streams para controlar o fluxo de dados entre os componentes. O BLoC ajuda a manter a separação de preocupações e a reatividade na atualização da interface do usuário.
- Padrão Provider: O Provider é um padrão de gerenciamento de estado que fornece uma maneira conveniente de compartilhar dados entre os diferentes componentes de um aplicativo Flutter. Ele permite que os widgets se inscrevam para receber atualizações dos dados e atualizem a interface do usuário de forma reativa.

3. MS SQL:

- Padrão de acesso a dados: Ao trabalhar com o MS SQL, é comum adotar padrões de acesso a dados, como o Object-Relational Mapping (ORM) ou o

ADO.NET. O ORM, por exemplo, permite que os dados sejam mapeados de forma transparente entre as tabelas do banco de dados e objetos na linguagem de programação, simplificando as operações de leitura e gravação de dados. O ADO.NET, por outro lado, oferece um conjunto de classes e interfaces para interagir diretamente com o banco de dados, permitindo o controle granular das operações de banco de dados.

Diagrama de implantação (definir quantos servidores, banco de dados, estrutura utilizada para o sistema e ***como instalar o sistema***);

14. Anexar o Diagrama ER do banco de dados e o dicionário de dados.

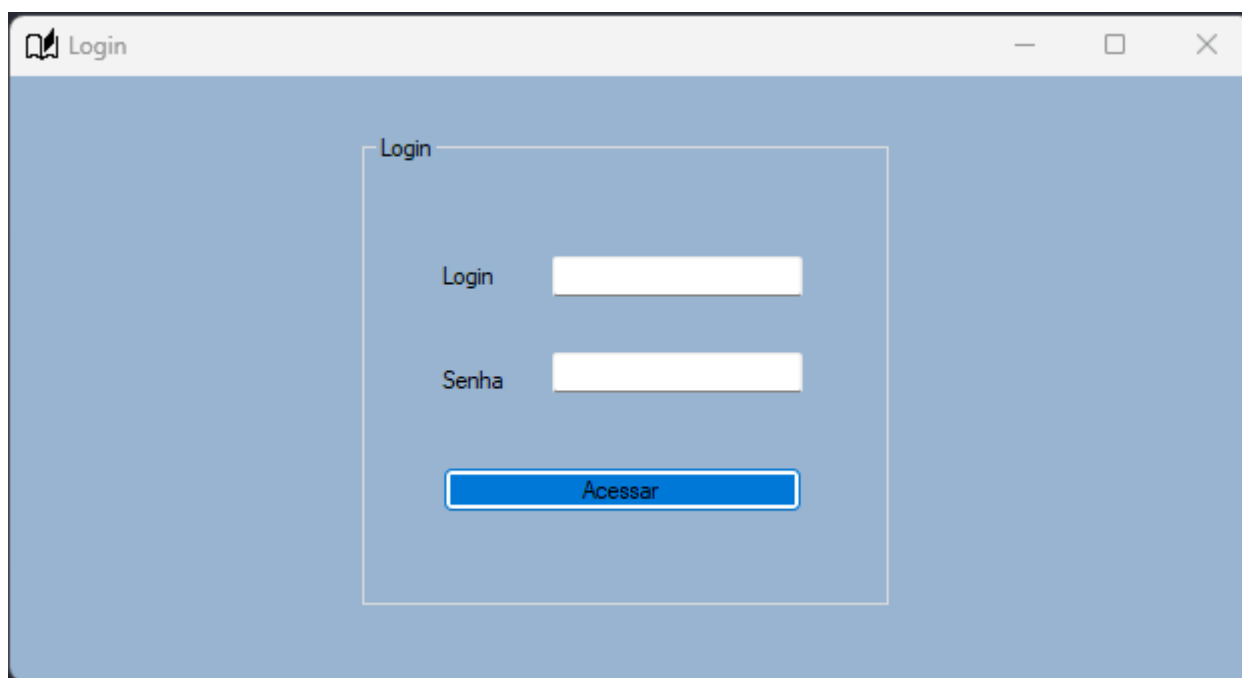
15. Anexar o **script de criação do banco** (usar uma ferramenta) e **scripts de dados iniciais de testes (roteiros de teste)** e homologação do sistema;

16. Anexar **planilhas de testes** para homologação do sistema, identificando como produzir as **evidências do teste** e as **queries no banco**, para certificar que os dados estão corretos (um mínimo de testes que assegurem a funcionalidade básica do sistema);

Sistema Desktop

O sistema Desktop contemplará modelos conforme abaixo

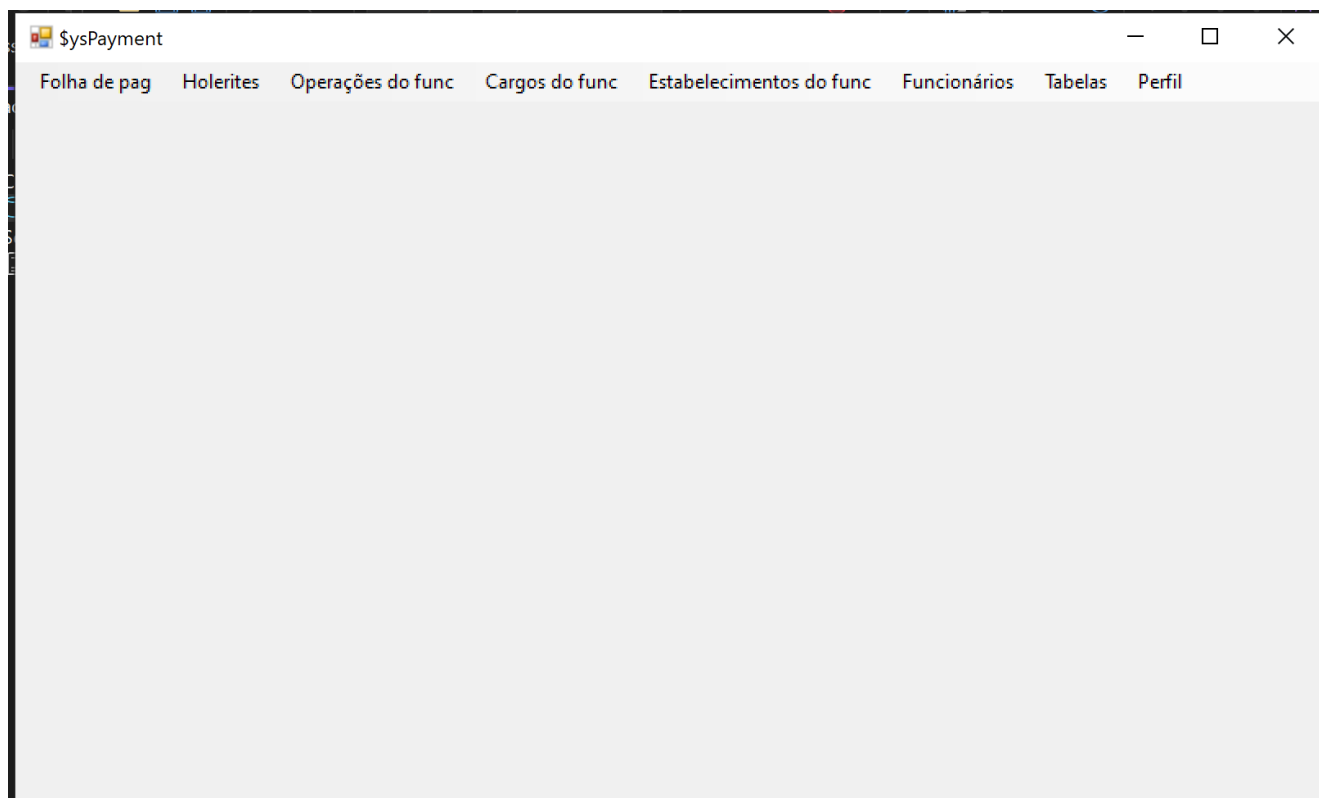
Figura - Tela de Login Desktop



The image shows a desktop application window titled "Login". The window has a light blue background. In the center, there is a white rectangular box containing the login form. The form includes two input fields: one labeled "Login" and another labeled "Senha" (Password). Below these fields is a blue button with the text "Acessar" (Access). The window's title bar is light gray and contains standard minimize, maximize, and close buttons.

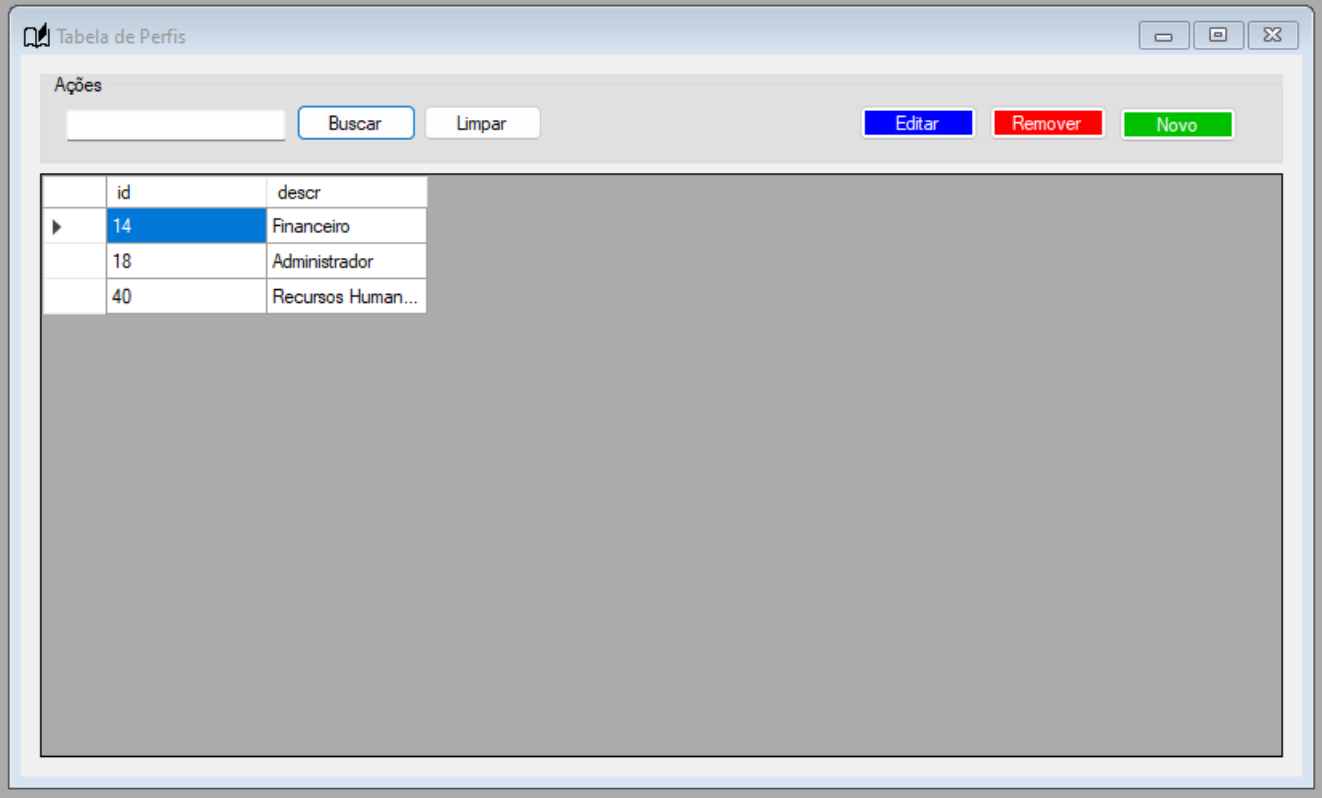
Fonte: Autoral (2023)

Figura - Todos os Menus Desktop



Fonte: Autoral (2023)

Figura - Tela de Perfil



Fonte: Autoral (2023)

Figura - Tela de Tabela - Exemplo da Tela de Funcionários

	id	sexo	nome	cpf	tel	email	cep	cidade
▶	10	Masculino	Testando	000.000.000-00	(00) 00000-00	admin@unip.br	12233-002	São Jo
	20	Masculino	João	452.552.668-84	(12) 98301-0463		12233-002	São Jo
	24	Masculino	Test	564.498.498-48	(12) 15641-6548		12233-002	São Jo

Fonte: Autoral (2023)

Figura - Tela de Cadastro - Exemplo da Tela de Funcionários

Dados Pessoais

Nome

Data Nascimento

Sexo

CPF

Endereço

CEP

Cidade

Estado

País

Rua

Bairro

Numero

Complemento

Profissional

Data Admissão

Horario Inicio

Horario Final

Quantidade de Dependente

Cargo

Estabelecimento

Acesso

Usuario

Senha

Perfil

Contato

Telefone

Email

Fonte: Autoral (2023)

10. CONCLUSÃO

NA CONCLUSÃO NÃO DEVE APARECER CITAÇÃO PORQUE É A PERCEPÇÃO DO GRUPO SOBRE O PROJETO REALIZADO, A SUA CONCLUSÃO DA PESQUISA.

Na conclusão deve haver um texto relacionando o problema que foi proposto e as soluções que foram sugeridas, de tal maneira a valorizar as opções adotadas e motivar o prosseguimento dos estudos, para que a empresa de locação receba mais propostas e contribuições da área de Análise e Desenvolvimento de Sistemas e suas aplicações.

11. REFERÊNCIAS

ACERVO LIMA O MAIOR ACERVO DE TUTORIAIS E REFERÊNCIAS Base de Dados Acervo lima. 2022. disponível em: <https://acervolima.com/vantagens-e-desvantagens-de-oop/> Acesso em: 10 maio 2023

ALVES, William P. Banco de Dados.: Editora Saraiva, 2014. E-book. ISBN 9788536518961. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536518961/>. Acesso em: 31 mai. 2023

ASP vantagens e desvantagens - 408 Palavras | Trabalhosfeitos. Disponível em: <https://www.trabalhosfeitos.com/ensaios/Asp-Vantagens-e-Desvantagens/31259653.html>. Acesso em: 13 abr. 2023.

BEUREH, He Maria. Gerenciamento de Informação: um recurso estratégico no processo de Gestão Empresarial. 2.ed. São Paulo: Atlas, 2000

BRASIL, E. M. Educa Mais Brasil - Bolsas de Estudo de até 70% para Faculdades – Graduação e Pós-graduação. Disponível em: <https://www.educamaisbrasil.com.br/cursos-e-faculdades/gestao-de-recursos-humanos/salario-de-gestor-de-recursos-humanos-carreira>. Acesso em: 20 abr. 2023.

CRM: O que é CRM e quais os benefícios? Disponível em: <https://www.salesforce.com/br/crm/#:~:text=CRM%20%C3%A9%20a%20sigla%20usada>.

EDUCAÇÃO, R. X. Flutter: o que é, para que serve e se vale a pena aprender. Disponível em: <https://blog.xpeducacao.com.br/flutter/#:~:text=Vantagens%20do%20Flutter&text=Multiplataforma%3A%20uma%20mesma%20base%20de>. Acesso em: 30 abr. 2023.

- **PMBOK® Guide – Sixth Edition. Project Management Institute, Inc., 2017.**
- **Schwalbe, K. (2018). Information Technology Project Management. Cengage Learning.**

- Heldman, K. (2018). **PMP: Project Management Professional Exam Study Guide**. Sybex.

PMI. **Um guia do conhecimento em gerenciamento de projetos. Guia PMBOK® 6a. ed. – EUA: Project Management Institute, 2017.**

Fowler, M. (2002). **Patterns of Enterprise Application Architecture**. Addison-Wesley.

VICTORINO, Marcio; BRÄSCHER, Marisa. **Organização da informação e do conhecimento, engenharia de software e arquitetura orientada a serviços: uma abordagem holística para o desenvolvimento de sistemas de informação computadorizados**. Revista de Ciência da Informação, v. 10, n. 3, 2009.

(Exemplos)

ALVES, L. **Educação a distância: conceitos e história no Brasil e no mundo**. Revista Brasileira de Aprendizagem Aberta e a Distância, v. 10, 2011. Disponível em: <http://www.abed.org.br/revistacientifica/Revista_PDF_Doc/2011/Artigo_07.pdf>. Acesso em: 05 fev. 2017.

ALVES, R. M.; ZAMBALDE, A. L.; FIGUEIREDO, C. X. **Ensino a distância**. Lavras: UFLA/FAEPE, 2004.

ARBACHE, F. S., SANTOS, A. G., MONTENEGRO, C., SALLES, W. F. **Gestão de logística, distribuição e trade marketing**. Rio de Janeiro: Editora FGV, 2014. 180p.

CARVALHO, D. T. de; NEVES, M. F. (Org.). **Marketing na nova economia**. São Paulo: Atlas, 2001.

COSTA, K. S.; FARIA, G. G. **EAD: sua origem histórica, evolução e atualidade brasileira face ao paradigma da educação presencial**. Disponível em: <<http://www.abed.org.br/congresso2008/tc/552008104927AM.pdf>>. Acesso em: 05 fev. 2017.

FARIA, M. A.; SILVA, R. C. S. **EAD: o professor e a inovação tecnológica**. Revista Brasileira de Aprendizagem Aberta e a Distância. São Paulo, dez. 2007.

GRAVES, P. **Por dentro da mente do consumidor**. Rio de Janeiro: Elsevier, 2011.

KOTLER, P., TRIAS de BES, F. **Marketing lateral – Uma abordagem revolucionária para criar novas oportunidades em mercados saturados**. Rio de Janeiro: Elsevier, 2004.

DENNIS, Alan; WIXOM, Barbara H.; ROTH, Roberta M. **Análise e Projeto de Sistemas**. Grupo GEN, 2014. 978-85-216-2634-3. <https://integrada.minhabiblioteca.com.br/#/books/978-85-216-2634-3/>

WAZLAWICK, Raul S. **Análise e Design Orientados a Objetos para Sistemas de Informação: Modelagem com UML, OCL e IFML** Grupo GEN, 2014. 9788595153653. <https://integrada.minhabiblioteca.com.br/#/books/9788595153653/>.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Grupo GEN, 2014. 9788535226270. <https://integrada.minhabiblioteca.com.br/#/books/9788535226270>

Grupo Nº _____ Ano _____ Período: _____ Orientador _____

Alunos:

[illegible][illegible]