**SPU – Sistema de Pagamento Uirapuru**

**Documento do Software**

**Versão 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Histórico de Revisões** | | | |
| **Versão** | **Data** | **Descrição** | **Autor** |
| 1.0 | 04/10/2024 | Primeiro formato do código pronto | Gabriela Galucio / Carlos Henrique / |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sumário

[1. Introdução 7](#_Toc183221079)

[1.1. Funcionalidades Principais 7](#_Toc183221080)

[1.2. Objetivo 8](#_Toc183221081)

[1.3. Escopo 8](#_Toc183221082)

[1.4. Limitações e Exclusões 9](#_Toc183221083)

[1.5. Objetivos do Sistema 9](#_Toc183221084)

[1.6. Publico-Alvo 10](#_Toc183221085)

[1.7. Definições, Acrônimos e Abreviações 10](#_Toc183221086)

[1.8. Referências 10](#_Toc183221087)

[1.9. Arquitetura do Sistema 10](#_Toc183221088)

[1.10. Requisitos de Hardware e Software 11](#_Toc183221089)

[2. Requisitos Funcionais 12](#_Toc183221100)

[2.1. Autenticação de Usuário: 12](#_Toc183221103)

[2.2. Dashboard Principal: 12](#_Toc183221104)

[2.3. Gestão de Pagamentos: 12](#_Toc183221105)

[2.4. Histórico de Pagamentos: 12](#_Toc183221106)

[2.5. Controle de Ponto: 13](#_Toc183221107)

[2.6. Tarefas e Cronogramas: 13](#_Toc183221108)

[2.7. Gerenciamento de Funcionários: 13](#_Toc183221109)

[2.8. Monitoramento de Serviços Prestados: 13](#_Toc183221110)

[2.9. Relatórios Gerenciais: 13](#_Toc183221111)

[2.10. Configurações e Auditoria: 14](#_Toc183221112)

[2.11. Suporte ao Usuário: 14](#_Toc183221113)

[3. Requisitos Não Funcionais 14](#_Toc183221114)

[3.1. Desempenho: 14](#_Toc183221116)

[3.2. Segurança: 14](#_Toc183221117)

[3.3. Escalabilidade: 15](#_Toc183221118)

[3.4. Usabilidade: 15](#_Toc183221119)

[3.5. Manutenibilidade: 15](#_Toc183221120)

[3.6. Disponibilidade: 15](#_Toc183221121)

[3.7. Compatibilidade: 15](#_Toc183221122)

[3.8. Internacionalização e Localização: 15](#_Toc183221123)

[4. Descrição das Telas 16](#_Toc183221124)

[4.1. Tela Principal: login.html 17](#_Toc183221126)

[4.2. Tela Principal: dashboard.html (ou dashboard.php) 17](#_Toc183221127)

[4.3. Tela Principal: gestaoPagamentos.html 17](#_Toc183221128)

[4.4. Subtela: registroNovoPg.html 17](#_Toc183221129)

[4.5. Subtela: comprovantePg.html 18](#_Toc183221130)

[4.6. Tela Principal: historicopg.html 18](#_Toc183221131)

[4.7. Subtela: detalhespagamento.html 18](#_Toc183221132)

[4.8. Tela Principal: controlheponto.html 18](#_Toc183221133)

[4.9. Subtela: registroponto.html 19](#_Toc183221134)

[4.10. Tela Principal: tarefas.html 19](#_Toc183221135)

[4.11. Tela Principal: gerenciamentoFuncionarios.html 19](#_Toc183221136)

[4.12. Subtela: cadastrofuncionario.html 19](#_Toc183221137)

[4.13. Subtela: documentosFuncionario.html 20](#_Toc183221138)

[4.14. Tela Principal: servicosPrestados.html 20](#_Toc183221139)

[4.15. Tela Principal: portal\_funcionario.html 20](#_Toc183221140)

[4.16. Subtela: historicoFuncionario.html (já descrita acima) 20](#_Toc183221141)

[4.17. Subtela: relatoriosPessoais.html 20](#_Toc183221142)

[4.18. Tela Principal: relatorios.html 21](#_Toc183221143)

[4.19. Tela Principal: configuracoes.html 21](#_Toc183221144)

[4.20. Subtela: auditoriaLogin.html 21](#_Toc183221145)

[4.21. Tela Principal: suporte.html 21](#_Toc183221146)

[4.22. Subtela: recuperacao.html 22](#_Toc183221147)

[5. Diagramas do Sistema 22](#_Toc183221148)

[5.1. Diagramas de casos de Uso 22](#_Toc183221150)

[5.2. Diagrama de Relacionamento de Usuários (Diagrama UR) 24](#_Toc183221151)

[5.3. Diagrama de Atividade 25](#_Toc183221152)

[5.4. Diagrama de Classes 27](#_Toc183221153)

[6. Banco de Dados (Entidades Principais e Atributos) 30](#_Toc183221154)

[6.1. Usuários 30](#_Toc183221156)

[6.2. Pagamentos 30](#_Toc183221157)

[6.3. Ponto 30](#_Toc183221158)

[6.4. Tarefas 30](#_Toc183221159)

[6.5. Serviços Prestados 31](#_Toc183221160)

[6.6. Funcionários e Documentos 31](#_Toc183221161)

[6.7. Relatórios Pessoais e Gerenciais 31](#_Toc183221173)

[6.8. Configurações e Auditoria 31](#_Toc183221186)

[6.9. Suporte e Recuperação de Senha 32](#_Toc183221187)

[6.10. Relacionamentos Entre Entidades 32](#_Toc183221188)

[6.11. Usuários e Ponto: Um usuário pode ter vários registros de ponto (entradas e saídas). 32](#_Toc183221204)

[6.12. Usuários e Tarefas: Um usuário pode ser designado para várias tarefas. 33](#_Toc183221205)

[6.13. Usuários e Serviços Prestados: Um usuário pode estar associado a vários serviços prestados. 34](#_Toc183221206)

[6.14. Usuários e Documentos: Um usuário pode ter vários documentos associados a ele. 34](#_Toc183221207)

[6.15. Usuários e Relatórios: Um usuário pode ter vários relatórios gerados para ele. 35](#_Toc183221208)

[6.16. Usuários e Auditoria: Cada tentativa de login é registrada, e pode haver múltiplas tentativas de login para um único usuário. 35](#_Toc183221209)

[6.17. Usuários e Suporte: Um usuário pode enviar várias mensagens de suporte. 36](#_Toc183221210)

[6.18. Usuários e Recuperação de Senha: Um usuário pode solicitar a recuperação de senha várias vezes, mas cada solicitação é registrada uma vez. 36](#_Toc183221211)

[6.19. Fluxo de Navegação Completo e Ações no Banco de Dados 37](#_Toc183221212)

[6.19.1. Gestão de Pagamentos: 37](#_Toc183221213)

[6.19.2. Controle de Ponto: 38](#_Toc183221214)

[6.19.3. Tarefas e Cronogramas: 38](#_Toc183221215)

[6.19.4. Monitoramento de Serviços Prestados 38](#_Toc183221216)

[6.19.5. Portal do Funcionário: 39](#_Toc183221217)

[6.19.6. Relatórios Gerenciais 39](#_Toc183221218)

[6.19.7. Configurações e Auditoria: 40](#_Toc183221219)

[6.19.8. Suporte e Recuperação de Senha: 40](#_Toc183221220)

[7. Criando o Banco de Dados 41](#_Toc183221221)

[7.1. Estrutura das Tabelas 41](#_Toc183221223)

[7.2. Tabela de Usuários 41](#_Toc183221224)

[7.3. Tabela de Pagamentos 42](#_Toc183221225)

[7.4. Tabela de Registros de Ponto 42](#_Toc183221226)

[7.5. Tabela de Tarefas 42](#_Toc183221227)

[7.6. Tabela de Serviços Prestados 43](#_Toc183221228)

[7.7. Tabela de Documentos dos Funcionários 43](#_Toc183221229)

[7.8. Tabela de Relatórios 43](#_Toc183221230)

[7.9. Tabela de Auditoria de Login 44](#_Toc183221231)

[7.10. Tabela de Suporte 44](#_Toc183221232)

[7.11. Tabela de Recuperação de Senha 44](#_Toc183221233)

[7.12. Procedimentos Armazenados 45](#_Toc183221234)

[7.13. Verificação de Login 45](#_Toc183221235)

[7.14. Inserção de Novo Pagamento 45](#_Toc183221236)

[7.15. Inserção de Novo Registro de Ponto 46](#_Toc183221237)

[7.16. Índices para Otimização 46](#_Toc183221238)

[7.17. Índice de Usuários por Email 46](#_Toc183221239)

[7.18. Índice de Pagamentos por Usuário 46](#_Toc183221240)

[7.19. Conclusão 46](#_Toc183221241)

[8. Arquitetura e Camadas 47](#_Toc183221242)

[8.1. Camada de Apresentação (Frontend) 47](#_Toc183221244)

[8.2. Camada de Lógica de Negócio (Backend) 47](#_Toc183221254)

[8.3. Camada de Persistência (Banco de Dados) 47](#_Toc183221255)

[8.4. Camada de Integração com Sistemas Externos 48](#_Toc183221256)

[8.5. Descrição das Camadas 48](#_Toc183221257)

[8.6. Fluxo de Trabalho: 48](#_Toc183221258)

[8.7. Camada de Lógica de Negócio (Backend) 48](#_Toc183221259)

[8.8. Fluxo de Trabalho: 48](#_Toc183221260)

[8.9. Camada de Persistência (Banco de Dados) 49](#_Toc183221261)

[8.10. Fluxo de Trabalho: 49](#_Toc183221262)

[8.11. Camada de Integração com Sistemas Externos 49](#_Toc183221263)

[8.12. Fluxo de Trabalho: 49](#_Toc183221264)

[9. CODIGOS 49](#_Toc183221265)

[10. Procedimentos de Instalação 50](#_Toc183221266)

[11. Testes 50](#_Toc183221267)

[12. Manutenção e Suporte 50](#_Toc183221268)

[13. Anexos 50](#_Toc183221269)

[14. Referências 50](#_Toc183221270)

[15. Aprovações 50](#_Toc183221271)

Documento do Software

# Introdução

Este sistema foi desenvolvido para gerenciar de forma eficiente e integrada as operações de uma empresa que lida com uma força de trabalho diversificada, incluindo funcionários internos e prestadores de serviços externos. Seu objetivo principal é otimizar e automatizar processos como o registro de ponto, gestão de pagamentos, controle de tarefas, monitoramento de serviços prestados, e a geração de relatórios financeiros e de desempenho.

A plataforma oferece uma interface intuitiva para os administradores, gerentes e funcionários, com acesso a funcionalidades específicas para cada tipo de usuário. Além disso, o sistema possui um painel de controle robusto, a **Dashboard**, que centraliza o gerenciamento de todas as operações, proporcionando fácil acesso às áreas mais críticas de operação da empresa.

## Funcionalidades Principais

**Gestão de Pagamentos**: Permite o registro, controle e emissão de comprovantes de pagamento para funcionários e prestadores de serviços, com opções de pagamento recorrente (diário, semanal, quinzenal, mensal).

**Controle de Ponto**: Facilita o registro de entradas e saídas dos funcionários, possibilitando o monitoramento de pontualidade, horas trabalhadas e geração de relatórios relacionados.

**Tarefas e Cronogramas**: Permite a atribuição de tarefas para os funcionários ou equipes, com acompanhamento do status de cada tarefa (pendente, em andamento, concluída) e a gestão de cronogramas.

**Relatórios Gerenciais**: Geração de relatórios detalhados sobre a produtividade dos funcionários, desempenho dos prestadores de serviços e resultados financeiros da empresa. esses relatórios auxiliam na análise de desempenho e na tomada de decisões estratégicas.

**Monitoramento de Serviços Prestados**: Registra e monitora os serviços realizados por prestadores externos (diaristas, empreiteiras, etc.), associando-os ao pagamento específico por serviço prestado.

**Portal do Funcionário**: Interface dedicada ao acesso de cada funcionário às suas informações pessoais, como histórico de ponto, pagamentos, desempenho e relatórios detalhados sobre suas tarefas.

**Configurações e Auditoria**: Permite a personalização do sistema e a auditoria das tentativas de login, garantindo segurança e controle sobre o acesso ao sistema.

**Suporte e Recuperação de Senha**: Sistema de faq para resolução de dúvidas frequentes e formulários de contato para solicitação de suporte, além de um módulo de recuperação de senha para usuários.

## Objetivo

O principal objetivo deste sistema é aumentar a eficiência operacional da empresa, automatizando processos e proporcionando uma visão detalhada de todas as áreas críticas. A gestão integrada de pagamentos, ponto, tarefas e serviços prestados, juntamente com relatórios gerenciais, permite que a empresa tome decisões baseadas em dados, otimize suas operações e melhore o desempenho geral.

Este sistema busca não apenas simplificar as atividades diárias, mas também oferecer uma plataforma segura e de fácil utilização para todos os usuários, garantindo a conformidade com as políticas internas da empresa e a satisfação de seus funcionários e prestadores de serviços.

## Escopo

O escopo deste sistema abrange o desenvolvimento e implementação de uma plataforma integrada para gerenciar e otimizar diversas operações empresariais, incluindo o controle de ponto, gestão de pagamentos, acompanhamento de tarefas e cronogramas, monitoramento de serviços prestados, e a geração de relatórios gerenciais. O sistema foi projetado para atender tanto as necessidades administrativas quanto operacionais da empresa, proporcionando uma interface amigável e funcionalidades avançadas de controle e análise.

**Funcionalidades Principais**

1. **Autenticação e Acesso**: O sistema possui uma tela de login com autenticação para garantir que apenas usuários autorizados acessem as funcionalidades. Após a autenticação, os usuários são direcionados à Dashboard, onde podem acessar as diferentes áreas do sistema de acordo com o perfil e permissões.
2. **Gestão de Pagamentos**: O módulo de pagamentos permite o registro de pagamentos de funcionários e prestadores de serviços, com suporte para pagamentos em diferentes periodicidades (diário, semanal, quinzenal, mensal). Ele também oferece a capacidade de gerar comprovantes de pagamento e controlar o histórico de transações.
3. **Controle de Ponto**: A funcionalidade de controle de ponto registra as entradas e saídas dos funcionários, seja manualmente ou por dispositivos automatizados. Além disso, o sistema permite visualizar e corrigir registros de ponto, garantindo um acompanhamento detalhado da jornada de trabalho dos funcionários.
4. **Tarefas e Cronogramas**: Os administradores podem criar, atribuir e acompanhar tarefas, separadas por categorias como pendente, em andamento ou concluída. O sistema permite o monitoramento de prazos e status das tarefas, com notificações automáticas para alertar sobre tarefas atrasadas.
5. **Relatórios Gerenciais**: O sistema oferece relatórios detalhados sobre produtividade, desempenho de funcionários, resultados financeiros e outros aspectos relevantes para a gestão. Esses relatórios ajudam os administradores a tomar decisões informadas e a otimizar as operações da empresa.
6. **Monitoramento de Serviços Prestados**: Para empresas que contratam prestadores de serviços externos, o sistema oferece o registro e acompanhamento de serviços realizados, incluindo a vinculação de tarefas específicas aos prestadores e o pagamento correspondente aos serviços executados.
7. **Portal do Funcionário**: Cada funcionário tem acesso ao seu portal pessoal, onde pode consultar seu histórico de ponto, pagamentos, desempenho e outros relatórios relacionados às suas tarefas e atividades dentro da empresa.
8. **Configurações e Auditoria**: A área de configurações permite aos administradores ajustar as definições gerais do sistema, enquanto a auditoria de login oferece uma camada extra de segurança, monitorando as tentativas de acesso ao sistema.
9. **Suporte e Recuperação de Senha**: O sistema inclui uma seção de suporte, com um FAQ detalhado para resolver as dúvidas mais frequentes, além de um formulário de contato para solicitações de assistência. Também oferece uma funcionalidade de recuperação de senha para usuários que esquecem suas credenciais de acesso.

## Limitações e Exclusões

Este sistema foi desenvolvido para cobrir os processos descritos acima, mas não inclui funcionalidades como:

* **Gestão de Estoque**: O sistema não possui controle sobre inventário ou gestão de estoques de produtos ou materiais.
* **Integração com Sistemas Externos**: A integração com outras plataformas de terceiros (por exemplo, sistemas de contabilidade ou CRM) não está prevista nesta versão.
* **Funcionalidades Avançadas de BI**: O sistema não oferece análise de dados avançada como ferramentas de Business Intelligence (BI), além dos relatórios gerenciais básicos.

## Objetivos do Sistema

* **Automatizar Processos**: O objetivo principal é reduzir o tempo gasto com tarefas administrativas e operacionais, automatizando o registro de ponto, o gerenciamento de pagamentos e a criação de tarefas.
* **Melhorar a Produtividade**: Fornecer ferramentas para monitorar a produtividade de funcionários e prestadores de serviços, garantindo que as metas sejam cumpridas e os resultados sejam facilmente acessíveis.
* **Facilitar a Tomada de Decisões**: Gerar relatórios e dados em tempo real que permitam a gestão da empresa tomar decisões informadas e estratégicas.
* **Proporcionar uma Experiência Intuitiva**: Criar uma interface de usuário fácil de navegar, que permita a todos os níveis de usuários acessarem suas respectivas funções de maneira simples e eficaz.

## Publico-Alvo

Este sistema é destinado a empresas que necessitam de um controle detalhado de suas operações relacionadas a funcionários e prestadores de serviços, incluindo setores de recursos humanos, finanças e operações. O sistema foi projetado para ser utilizado por administradores, gerentes, e funcionários, com funcionalidades e acessos definidos de acordo com o perfil de cada usuário.

## Definições, Acrônimos e Abreviações

• CLT: Consolidação das Leis do Trabalho

• LGPD: Lei Geral de Proteção de Dados

• eSocial: Sistema de Escrituração Digital das Obrigações Fiscais, Previdenciárias e Trabalhistas

## Referências

• Legislação LGPD

• Normas do eSocial

## Arquitetura do Sistema

O A arquitetura do sistema é projetada para garantir flexibilidade, escalabilidade e integração eficiente com outras plataformas existentes. A estrutura do sistema segue uma abordagem moderna e modular, permitindo que as diferentes funcionalidades interajam de forma coesa.

A **arquitetura baseada em API** permite que o sistema seja facilmente integrado com outros sistemas externos, como o sistema de ponto já existente na organização. A comunicação entre o sistema de controle de ponto e a plataforma é realizada por meio de **APIs RESTful**, facilitando a troca de dados entre diferentes sistemas de forma segura e eficiente.

Além disso, o sistema incorpora **componentes de inteligência artificial** para realizar a **análise de dados** e otimizar processos internos. As funcionalidades de IA podem incluir a análise de desempenho de funcionários, previsão de atrasos nas tarefas e a detecção de padrões nos dados de ponto e pagamento, com o objetivo de melhorar a gestão e tomar decisões informadas.

A arquitetura é dividida nas seguintes camadas:

* **Camada de Apresentação (Frontend)**: Responsável pela interface do usuário, onde são apresentadas as funcionalidades do sistema, como visualização de dados, formulários de registro e relatórios.
* **Camada de Lógica de Negócio (Backend)**: Gerencia as regras de negócios e executa as operações necessárias, como registro de ponto, gestão de pagamentos, atribuição de tarefas, e controle de usuários.
* **Camada de Dados (Banco de Dados)**: Armazena todas as informações do sistema, incluindo registros de ponto, históricos de pagamento, dados dos funcionários, relatórios gerenciais e configurações do sistema.

Essa arquitetura modular facilita a manutenção e atualização do sistema, pois novas funcionalidades podem ser adicionadas ou modificadas sem afetar as demais partes do sistema.

.

## Requisitos de Hardware e Software



**Requisitos de Hardware**

* **Processador**: Processador de 2 GHz ou superior (recomendado multi-core para desempenho otimizado).
* **Memória RAM**: 4 GB de RAM ou mais (para garantir o bom funcionamento do sistema com múltiplos usuários simultâneos).
* **Espaço de Armazenamento**: Pelo menos 10 GB de espaço livre para armazenar os dados do sistema e backups.
* **Conexão com a Internet**: Conexão estável de internet (recomenda-se uma velocidade mínima de 10 Mbps para garantir a comunicação eficiente com o servidor e a execução das funcionalidades do sistema).

**Requisitos de Software**

* **Sistema Operacional**: Windows 10 ou superior (compatível com servidores PHP e banco de dados MySQL).
* **Navegador Web**: O sistema é acessado via navegador, sendo compatível com as versões mais recentes dos seguintes navegadores:
  + Google Chrome (versão 85 ou superior)
  + Mozilla Firefox (versão 80 ou superior)
  + Microsoft Edge (versão 85 ou superior)
* **Servidor Web**: O sistema requer um servidor web PHP para o correto funcionamento das páginas dinâmicas e interativas. Recomendamos o uso do **Apache** ou **Nginx** como servidores web.
* **Banco de Dados**: MySQL 5.7 ou superior. O banco de dados é responsável por armazenar todas as informações cruciais, como registros de ponto, dados de funcionários, históricos de pagamento, tarefas e relatórios. O banco de dados deve ser configurado para suportar um alto volume de dados e permitir consultas eficientes.
* **Integração com Sistema de Ponto**: O sistema deve ser compatível com o **sistema de controle de ponto existente** da empresa. Isso pode incluir a integração por API ou outro método de comunicação que permita a sincronização de dados de entrada e saída de funcionários automaticamente.

Esses requisitos garantem que o sistema funcione de maneira eficiente e segura, proporcionando uma experiência de usuário de alto desempenho e facilidade de integração com as ferramentas e plataformas já utilizadas pela empresa.

# Requisitos Funcionais

Os **requisitos funcionais** especificam as funcionalidades que o sistema deve oferecer, ou seja, o que o sistema deve fazer para atender às necessidades do usuário. Abaixo estão os requisitos funcionais detalhados para o sistema de gestão:



## Autenticação de Usuário:

* O sistema deve permitir o login de usuários com diferentes perfis (Administrador, Funcionário).
* O sistema deve fornecer funcionalidade para recuperação de senha por e-mail.

## Dashboard Principal:

* Após o login, o usuário deve ser redirecionado para o painel de controle (Dashboard), onde poderá acessar todas as funcionalidades principais do sistema.
* O dashboard deve exibir informações resumidas sobre os pagamentos, ponto, tarefas e outros indicadores relevantes.

## Gestão de Pagamentos:

* O sistema deve permitir o registro de pagamentos para funcionários e prestadores de serviço (diários, semanais, quinzenais e mensais).
* O sistema deve permitir o upload de comprovantes de pagamento.
* O sistema deve permitir o acesso aos comprovantes de pagamento.

## Histórico de Pagamentos:

* O sistema deve manter um histórico detalhado de todos os pagamentos realizados.
* O usuário deve ser capaz de visualizar detalhes completos de cada pagamento, incluindo período, tipo (funcionário/prestador), e método de pagamento.

## Controle de Ponto:

* O sistema deve permitir o registro manual de ponto (entrada e saída) de funcionários.
* O sistema deve permitir o registro automático de ponto para funcionários, baseado no horário do servidor.
* O sistema deve exibir um histórico completo de pontos registrados para cada funcionário.

## Tarefas e Cronogramas:

* O sistema deve permitir a definição de tarefas e cronogramas para equipes e indivíduos.
* O sistema deve permitir o acompanhamento do status das tarefas (pendente, em andamento, concluído).
* O sistema deve enviar notificações automáticas de tarefas atrasadas.

## Gerenciamento de Funcionários:

* O sistema deve permitir o cadastro de novos funcionários.
* O sistema deve permitir a visualização e a edição dos dados cadastrais de funcionários.
* O sistema deve permitir a visualização do histórico completo de ponto, pagamento e tarefas de cada funcionário.

## Monitoramento de Serviços Prestados:

* O sistema deve registrar os serviços prestados por funcionários e prestadores externos.
* O sistema deve vincular serviços realizados a tarefas e responsáveis.
* O sistema deve registrar pagamentos por serviços prestados.

## Relatórios Gerenciais:

* O sistema deve gerar relatórios detalhados sobre a produtividade e desempenho de funcionários e prestadores.
* O sistema deve gerar relatórios financeiros, com informações detalhadas sobre pagamentos e serviços prestados.
* O sistema deve permitir a exportação dos relatórios em formatos como PDF e Excel.

## Configurações e Auditoria:

* O sistema deve permitir configurações gerais de operação (idioma, fuso horário, permissões de usuário).
* O sistema deve registrar tentativas de login e fornecer relatórios de auditoria de acesso.

## Suporte ao Usuário:

* O sistema deve disponibilizar uma seção de FAQ para resolver as dúvidas mais comuns.
* O sistema deve permitir que o usuário envie mensagens de suporte através de um formulário de contato.

.

# Requisitos Não Funcionais

Os **requisitos não funcionais** especificam as qualidades e características que o sistema deve ter, mas que não estão diretamente relacionados às funcionalidades do sistema. Abaixo estão os requisitos não funcionais para o sistema de gestão:



## Desempenho:

* O sistema deve ser capaz de suportar até 1000 usuários simultâneos, com tempo de resposta de até 2 segundos para as páginas principais.
* O sistema deve carregar as páginas e interações de forma eficiente, mesmo com grandes volumes de dados.

## Segurança:

* O sistema deve usar criptografia SSL/TLS para garantir a segurança das informações transmitidas entre o cliente e o servidor.
* As senhas de usuários devem ser armazenadas de forma segura, utilizando técnicas de hashing e salting.
* O sistema deve implementar controle de acesso baseado em papéis (admin, funcionário) para restringir as permissões de usuários.
* O sistema deve permitir a autenticação multifatorial (2FA) para aumentar a segurança nas tentativas de login.

## Escalabilidade:

* O sistema deve ser projetado de maneira escalável, permitindo a adição de novas funcionalidades e aumento do número de usuários sem comprometer a performance.
* O banco de dados deve ser otimizado para suportar um grande volume de dados e permitir consultas rápidas.

## Usabilidade:

* O sistema deve ser intuitivo e fácil de usar, com uma interface amigável, com menus e botões bem definidos.
* A navegação entre as telas deve ser simples e fluida, com feedback adequado para ações realizadas pelo usuário.
* O sistema deve ser responsivo, funcionando corretamente em diferentes dispositivos (desktop, tablets, smartphones).

## Manutenibilidade:

* O código-fonte do sistema deve ser bem documentado e modular, facilitando a manutenção e a adição de novas funcionalidades.
* O sistema deve permitir a fácil atualização de suas partes, sem afetar outras funcionalidades do sistema.
* O sistema deve possuir logs detalhados para facilitar a resolução de problemas e a auditoria.

## Disponibilidade:

* O sistema deve estar disponível para acesso 99,9% do tempo, com o mínimo de tempo de inatividade para manutenção.
* O sistema deve ter backup automático diário para garantir a recuperação dos dados em caso de falhas.

## Compatibilidade:

* O sistema deve ser compatível com os navegadores modernos (Chrome, Firefox, Safari, Edge) e versões recentes de sistemas operacionais (Windows, macOS, Linux).
* O sistema deve ser compatível com dispositivos móveis (smartphones e tablets) através de uma interface responsiva.

## Internacionalização e Localização:

* O sistema deve ser capaz de suportar diferentes idiomas e fusos horários, permitindo que a empresa use a plataforma em diferentes locais geográficos.
* O sistema deve ter a capacidade de exibir datas, horários e formatos monetários de acordo com a localidade configurada.

Esses requisitos são essenciais para garantir que o sistema não apenas atenda às necessidades funcionais, mas também ofereça uma experiência de usuário eficiente, segura e agradável.

# Descrição das Telas

Abaixo está uma descrição detalhada de cada tela e subtela do sistema, incluindo suas funcionalidades e objetivos principais. Essa descrição ajuda a entender o fluxo de navegação do sistema e o que cada área específica realiza.



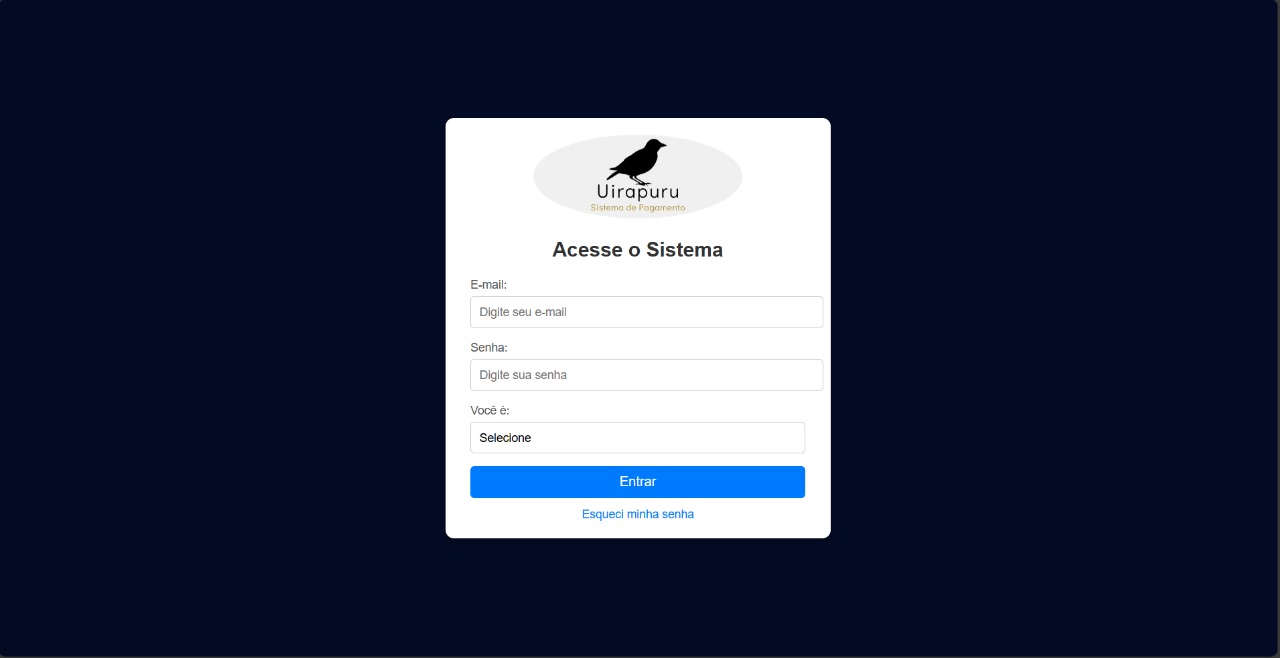
## ****Tela Principal: login.html****

**Objetivo**: Permite que o usuário faça a autenticação no sistema. Após o login, o usuário é redirecionado para a Dashboard.

**Funcionalidades**:

* + Formulário para entrada do usuário (login).
  + Recuperação de senha.
  + Validação das credenciais de usuário.

Imagem

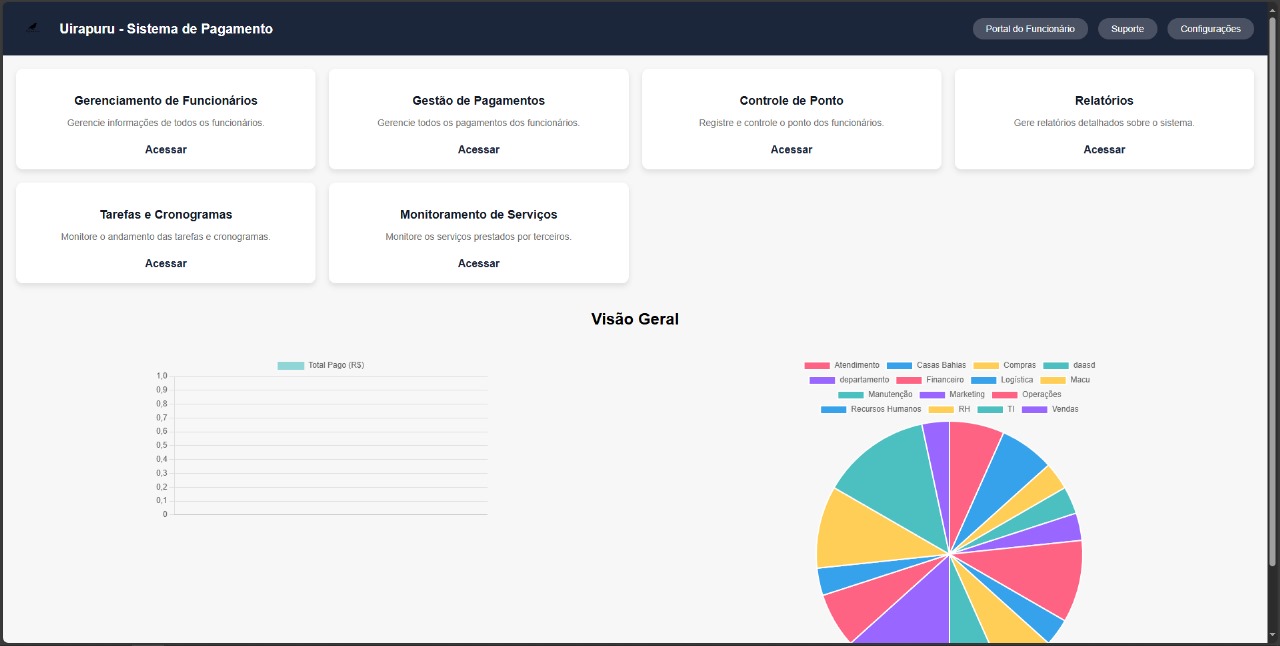


## ****Tela Principal: dashboard.html (ou dashboard.php)****

**Objetivo**: Serve como o painel de controle principal do sistema, de onde o usuário pode acessar as principais funcionalidades e telas do sistema.

**Funcionalidades**:

* + Exibição de resumos das informações do sistema, como pagamento, ponto, tarefas e outros indicadores.
  + Acesso direto às principais telas do sistema (Gestão de Pagamentos, Controle de Ponto, Tarefas, etc.).
  + Personalização do painel com widgets de fácil acesso.

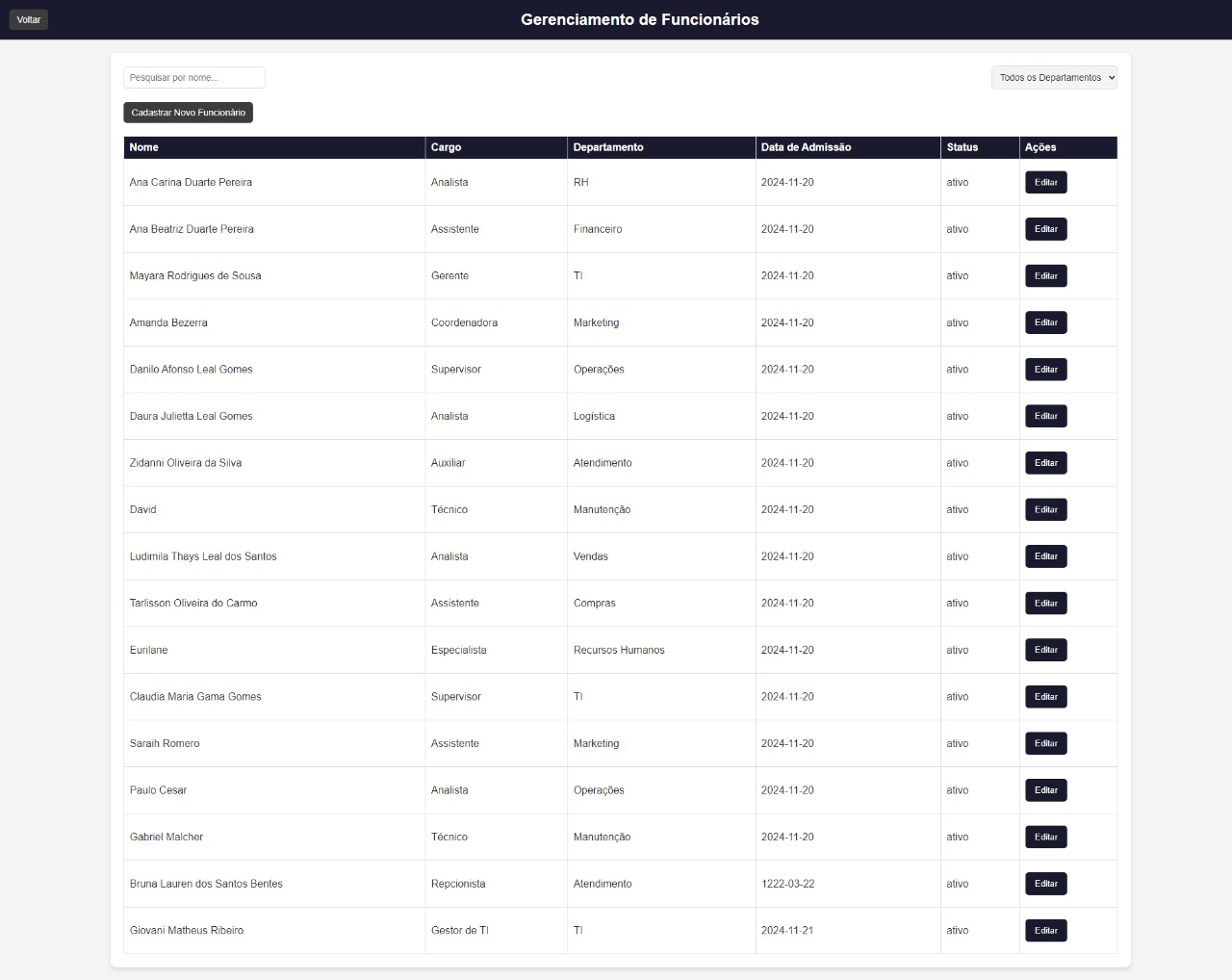


## ****Tela Principal: Gerenciamento de Funcionario****

**Objetivo**: Gerenciar todos os funcionários.

**Funcionalidades**:

* + Registro de novos pagamentos.
  + Acesso a registros de pagamentos anteriores.
  + Acesso a comprovantes de pagamento.

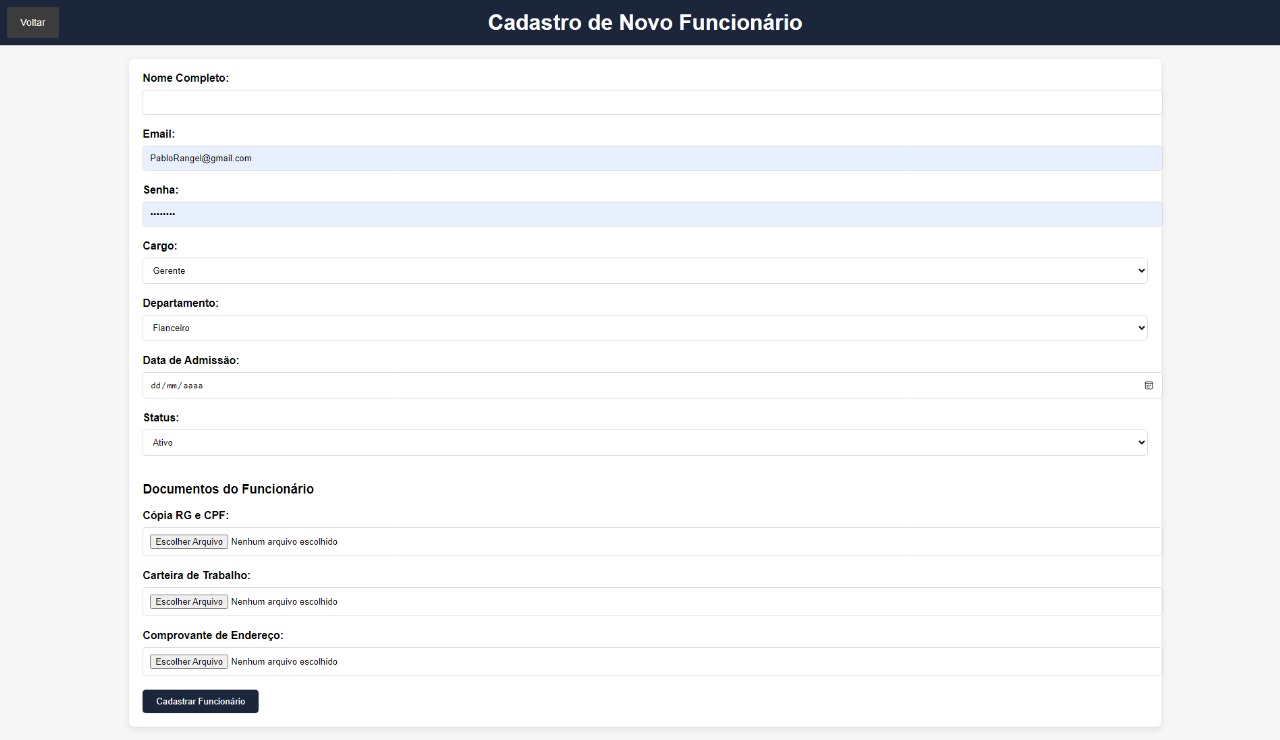


## ****Tela Principal: gestaoPagamentos.html****

**Objetivo**: Gerenciar todos os pagamentos dos funcionários e prestadores de serviços.

**Funcionalidades**:

* + Registro de novos pagamentos.
  + Acesso a registros de pagamentos anteriores.
  + Acesso a comprovantes de pagamento.



## ****Subtela: registroNovoPg.html****

**Objetivo**: Permite registrar um novo pagamento para funcionários ou prestadores de serviço.

**Funcionalidades**:

* + Seleção de tipo de pagamento (diário, semanal, quinzenal, mensal).
  + Escolha de funcionário/prestador.
  + Adição de informações sobre o pagamento (valor, método, etc.).

## ****Subtela: comprovantePg.html****

**Objetivo**: Permite o acesso aos comprovantes de pagamento.

**Funcionalidades**:

* + Exibição e download dos comprovantes de pagamento registrados no sistema.

## ****Tela Principal: historicopg.html****

**Objetivo**: Exibe todos os pagamentos realizados ao longo do tempo, com detalhes sobre o tipo e método de pagamento.

**Funcionalidades**:

* + Filtro por período, tipo de pagamento (funcionário ou prestador) e método.
  + Exibição de detalhes sobre cada pagamento realizado.

## ****Subtela: detalhespagamento.html****

**Objetivo**: Exibe informações detalhadas de um pagamento específico.

**Funcionalidades**:

* + Visualização de todos os dados do pagamento: data, valor, método, funcionário/prestador, etc.

## ****Tela Principal: controlheponto.html****

**Objetivo**: Gerenciar o controle de ponto dos funcionários, incluindo o registro de entrada e saída.

**Funcionalidades**:

* + Visualização do histórico de registros de ponto.
  + Opção para registrar ponto manualmente ou automaticamente.
  + Relatórios de tempo de trabalho por período.

## ****Subtela: registroponto.html****

**Objetivo**: Permite o registro manual de entrada e saída de funcionários.

**Funcionalidades**:

* + Formulário para registrar o horário de entrada e saída.
  + Exibição do horário registrado no banco de dados.

## ****Tela Principal: tarefas.html****

**Objetivo**: Permite o gerenciamento e acompanhamento de tarefas e cronogramas dos funcionários.

**Funcionalidades**:

* + Definição de tarefas para equipes ou indivíduos.
  + Acompanhamento do status da tarefa (pendente, em andamento, concluída).
  + Notificação de tarefas atrasadas.

## ****Tela Principal: gerenciamentoFuncionarios.html****

**Objetivo**: Gerencia as informações completas de todos os funcionários da empresa.

**Funcionalidades**:

* + Cadastro de novos funcionários.
  + Edição de dados de funcionários existentes.
  + Visualização do histórico de ponto, pagamentos e tarefas realizadas.

## ****Subtela: cadastrofuncionario.html****

**Objetivo**: Permite adicionar novos funcionários ao sistema.

**Funcionalidades**:

* + Formulário para inserir dados de funcionários (nome, cargo, salário, etc.).
  + Armazenamento dos dados no banco de dados.

## ****Subtela: documentosFuncionario.html****

**Objetivo**: Permite o upload e o gerenciamento de documentos importantes dos funcionários.

**Funcionalidades**:

* + Upload de documentos como RG, CPF, contratos, etc.
  + Visualização e download dos documentos.

## ****Tela Principal: servicosPrestados.html****

**Objetivo**: Gerencia os serviços prestados tanto por funcionários internos quanto por prestadores externos (ex: diaristas, empreiteiras).

**Funcionalidades**:

* + Cadastro de novos serviços prestados.
  + Vinculação de tarefas a responsáveis.
  + Registro de pagamentos para serviços prestados.
  + Histórico de serviços prestados por período.

## ****Tela Principal: portal\_funcionario.html****

**Objetivo**: Tela dedicada para que o funcionário acesse suas próprias informações.

**Funcionalidades**:

* + Acesso ao histórico de ponto, pagamentos e desempenho do funcionário.
  + Acesso aos relatórios pessoais.

## ****Subtela: historicoFuncionario.html**** (já descrita acima)

**Objetivo**: Visualização detalhada do histórico de ponto, pagamentos e tarefas do funcionário.

## ****Subtela: relatoriosPessoais.html****

**Objetivo**: Exibe relatórios detalhados sobre as atividades do funcionário.

**Funcionalidades**:

* + Relatório de pontualidade.
  + Relatório de tarefas concluídas e status.
  + Download e exportação dos relatórios.

## ****Tela Principal: relatorios.html****

**Objetivo**: Geração de relatórios analíticos para gerentes e administradores.

**Funcionalidades**:

* + Geração de relatórios de produtividade dos funcionários.
  + Relatórios financeiros de pagamentos realizados.
  + Relatórios de desempenho de funcionários e prestadores.

## ****Tela Principal: configuracoes.html****

**Objetivo**: Permite o gerenciamento das configurações gerais do sistema.

**Funcionalidades**:

* + Ajustes de idioma, fuso horário e permissões de usuário.
  + Configuração do sistema de auditoria de login.

## ****Subtela: auditoriaLogin.html****

**Objetivo**: Exibe o registro de tentativas de login ao sistema.

**Funcionalidades**:

* + Visualização de logs de tentativas de login.
  + Acompanhamento de tentativas de acesso não autorizadas.

## ****Tela Principal: suporte.html****

**Objetivo**: Fornece uma área de ajuda para os usuários, com FAQ e formulário de contato.

**Funcionalidades**:

* + Exibição de perguntas frequentes (FAQ).
  + Formulário para enviar mensagens ao suporte técnico.

## ****Subtela: recuperacao.html****

**Objetivo**: Processo de recuperação e redefinição de senha para usuários.

**Funcionalidades**:

* + Envio de link de recuperação de senha para o e-mail do usuário.
  + Formulário para redefinir a senha de acesso.

# Diagramas do Sistema

Abaixo estão os diagramas necessários para ilustrar visualmente os casos de uso, a relação entre usuários e o sistema, o fluxo de atividades e a estrutura de classes do sistema.



## Diagramas de casos de Uso

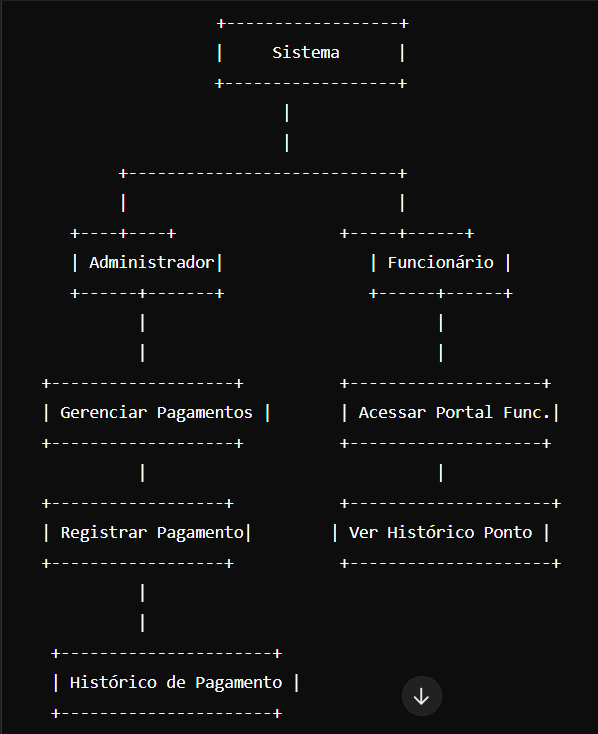
O **Diagrama de Casos de Uso** representa as interações entre os usuários (atores) e o sistema, mostrando quais funcionalidades cada ator pode acessar. Este diagrama ajuda a entender como os diferentes tipos de usuários interagem com o sistema.

**Atores**:

* Administrador
* Funcionário
* Prestador de Serviço

**Casos de Uso**:

1. **Login**: Realiza a autenticação no sistema.
2. **Acessar Dashboard**: Usuário é redirecionado para o painel de controle.
3. **Gerenciar Pagamentos**: Administra pagamentos para funcionários e prestadores.
4. **Acessar Histórico de Pagamentos**: Visualiza pagamentos anteriores.
5. **Registrar Ponto**: Registra a entrada e saída do funcionário.
6. **Gerenciar Tarefas**: Define, atribui e monitora tarefas.
7. **Gerenciar Funcionários**: Cadastra e edita dados de funcionários.
8. **Acessar Portal do Funcionário**: Funcionário acessa seu histórico e relatórios pessoais.
9. **Consultar Relatórios**: Geração de relatórios de produtividade e financeiros.
10. **Configurações e Auditoria**: Ajuste de configurações do sistema e auditoria de login.
11. **Suporte**: Acesso à FAQ e formulário de contato.



## Diagrama ****de Relacionamento de Usuários (Diagrama UR)****

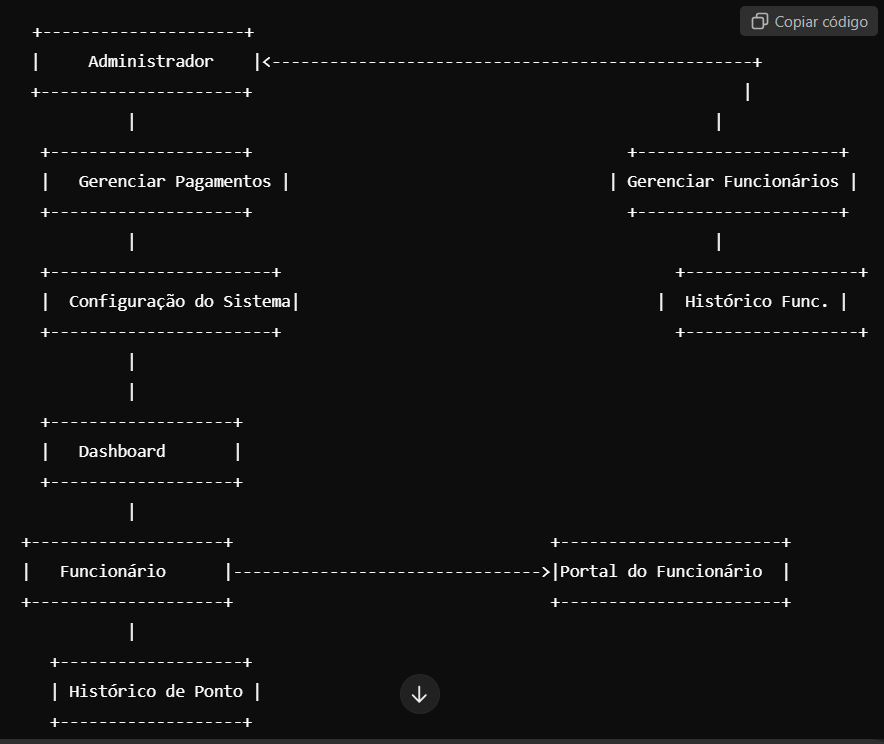
O **Diagrama de Relacionamento de Usuários (UR)** mostra a relação entre os usuários e o sistema, destacando quais usuários têm acesso a quais funcionalidades do sistema. Esse diagrama detalha como diferentes tipos de usuários interagem com os diversos módulos.

**Atores**:

* **Administrador**: Tem acesso a todas as funcionalidades e configurações do sistema.
* **Funcionário**: Acessa informações relacionadas a tarefas, ponto, pagamentos e relatórios pessoais.
* **Prestador de Serviço**: Acesso à visualização de serviços prestados e pagamentos.
* **Sistema**: Atua como o intermediário entre os usuários e os dados.

**Relacionamentos**:

* **Administrador** → Pode acessar todas as funcionalidades do sistema.
* **Funcionário** → Acessa o portal pessoal e histórico de tarefas e ponto.
* **Prestador de Serviço** → Acessa o histórico de serviços prestados e pagamentos.
* **Sistema** → Conecta todos os dados entre as funcionalidades e usuários.



## Diagrama de Atividade

O **Diagrama de Atividade** descreve o fluxo de trabalho em uma funcionalidade específica do sistema. Ele mostra como as ações ocorrem, os processos executados e os possíveis caminhos alternativos durante a execução de uma tarefa.

**Exemplo: Fluxo de Registro de Ponto**:

1. **Início**: O funcionário acessa a tela de registro de ponto.
2. **Escolha de Ação**: O sistema pergunta se o registro será manual ou automático.
   * Se manual, o funcionário insere a entrada e saída.
   * Se automático, o sistema registra o ponto automaticamente.
3. **Validação de Dados**: O sistema valida a entrada de dados.
   * Se válido, o sistema registra o ponto no banco de dados.
   * Se inválido, o sistema notifica o erro.
4. **Confirmação**: O sistema confirma que o ponto foi registrado corretamente.
5. **Fim**: O usuário é redirecionado para o histórico de ponto ou dashboard.

Início

|

V

[Funcionario Acessa Tela de Registro de Ponto]

|

V

[Escolhe Tipo de Registro]

|

|---> [Manual]

| |

| V

| [Funcionario Insere Hora de Entrada]

| |

| V

| [Funcionario Insere Hora de Saída]

| |

| V

| [Valida Dados]

| |

| V

| [Confirma Ponto]

| |

| V

| [Ponto Registrado]

| |

| V

| [Fim]

|

|---> [Automático]

|

V

[Sistema Registra Ponto]

|

V

[Ponto Registrado]

|

V

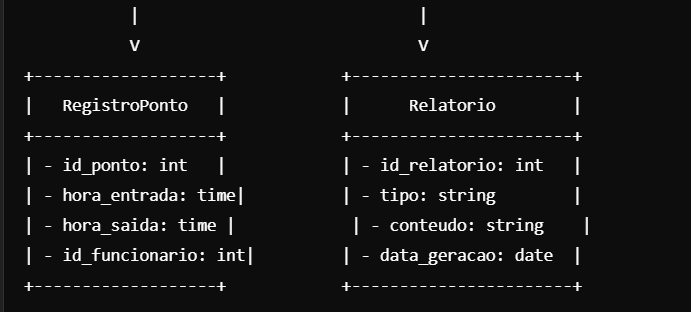
[Fim]

## Diagrama de Classes

O **Diagrama de Classes** descreve a estrutura do sistema, destacando as principais classes, seus atributos, métodos e os relacionamentos entre elas. Esse diagrama ajuda a entender como o sistema está organizado em termos de objetos e como eles se comunicam entre si.

**Exemplo de Classes**:

* **Classe: Usuario**
  + **Atributos**:
    - id\_usuario: int
    - nome: string
    - email: string
    - senha: string
    - tipo: string (Administrador, Funcionário, Prestador)
  + **Métodos**:
    - login()
    - logout()
    - alterarSenha()
* **Classe: Funcionario**
  + **Atributos**:
    - id\_funcionario: int
    - cargo: string
    - salario: float
  + **Métodos**:
    - registrarPonto()
    - visualizarRelatorio()
* **Classe: Pagamento**
  + **Atributos**:
    - id\_pagamento: int
    - valor: float
    - metodo: string (Cartão, Transferência, etc.)
    - data\_pagamento: date
  + **Métodos**:
    - registrarPagamento()
    - gerarComprovante()
* **Classe: Tarefa**
  + **Atributos**:
    - id\_tarefa: int
    - descricao: string
    - prazo: date
    - status: string (Pendente, Em andamento, Concluída)
  + **Métodos**:
    - atribuirResponsavel()
    - atualizarStatus()
* **Classe: Relatorio**
  + **Atributos**:
    - id\_relatorio: int
    - tipo: string (Produtividade, Financeiro, etc.)
    - conteudo: string
    - data\_geracao: date
  + **Métodos**:
    - gerarRelatorio()
    - exportarRelatorio()
* **Relacionamentos**:
  + Um **Usuario** pode ter múltiplos **Pagamentos** e **Tarefas**.
  + Um **Funcionario** pode ter múltiplos **Registros de Ponto** e **Relatórios**.
  + Um **Pagamento** é vinculado a um **Funcionario** ou **Prestador**.

. 

# ****Banco de Dados (Entidades Principais e Atributos)****



## Usuários

**Usuários**: Armazena informações de login e dados básicos dos usuários.

* id\_usuario (PK)
* nome
* email
* senha
* tipo\_usuario (admin, funcionário, prestador)
* status (ativo/inativo)
* data\_criacao
* data\_ultimo\_login

## Pagamentos

**Pagamentos**: Armazena registros de pagamentos realizados a funcionários e prestadores.

* id\_pagamento (PK)
* id\_usuario (FK)
* data\_pagamento
* valor
* periodo (diário, semanal, quinzenal, mensal)
* metodo (transferência, cheque, etc.)
* tipo\_pagamento (funcionário, prestador)
* status (pago, pendente)

## Ponto

**Registros\_ponto**: Registra entradas e saídas dos funcionários.

* id\_ponto (PK)
* id\_usuario (FK)
* data
* hora\_entrada
* hora\_saida
* status (presente, ausente, atrasado)

## Tarefas

**Tarefas**: Armazena informações sobre tarefas atribuídas aos funcionários.

* id\_tarefa (PK)
* id\_usuario (FK)
* descricao
* data\_inicio
* data\_fim
* status (pendente, em andamento, concluída)
* prioridade (baixa, média, alta)

## Serviços Prestados

**Servicos\_prestados**: Armazena dados sobre serviços realizados por funcionários ou prestadores externos.

* id\_servico (PK)
* descricao\_servico
* id\_usuario (FK)
* data\_servico
* valor\_pago
* status\_pagamento (pago, pendente)

## Funcionários e Documentos



**Documentos\_funcionario**: Armazena documentos pessoais e contratuais dos funcionários.

* id\_documento (PK)
* id\_usuario (FK)
* tipo\_documento (RG, CPF, contrato, etc.)
* documento\_url
* data\_upload

## ****Relatórios Pessoais e Gerenciais****



**Relatorios: Armazena os relatórios gerados no sistema.**

* id\_relatorio (PK)
* id\_usuario (FK) *(relatório gerado para um usuário específico)*
* tipo\_relatorio (produtividade, financeiro, desempenho)
* data\_criacao
* conteudo\_relatorio (armazenado em formato JSON ou texto)

## ****Configurações e Auditoria****

**Auditoria\_login: Registra todas as tentativas de login no sistema.**

* id\_auditoria (PK)
* id\_usuario (FK)
* data\_tentativa
* status\_tentativa (sucesso, falha)
* ip\_tentativa

## ****Suporte e Recuperação de Senha****

**Suporte: Armazena mensagens de suporte enviadas pelos usuários.**

* id\_suporte (PK)
* id\_usuario (FK)
* assunto
* mensagem
* status (pendente, em andamento, concluída)
* data\_envio

**Recuperacao\_senha: Registra solicitações de recuperação de senha.**

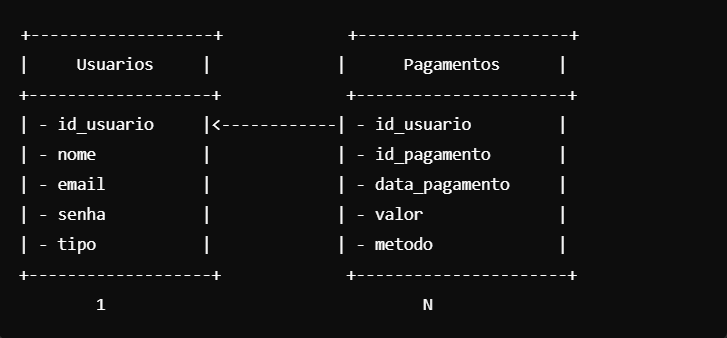
* id\_recuperacao (PK)
* id\_usuario (FK)
* data\_solicitacao
* status (pendente, concluído)

## ****Relacionamentos Entre Entidades****



**Usuários e Pagamentos**: Um usuário pode receber múltiplos pagamentos, mas cada pagamento é associado a apenas um usuário.

**Relacionamento 1** - Entre usuários e pagamentos.



## Usuários e Ponto: Um usuário pode ter vários registros de ponto (entradas e saídas).

**Relacionamento 1** - Entre usuários e registros\_ponto.



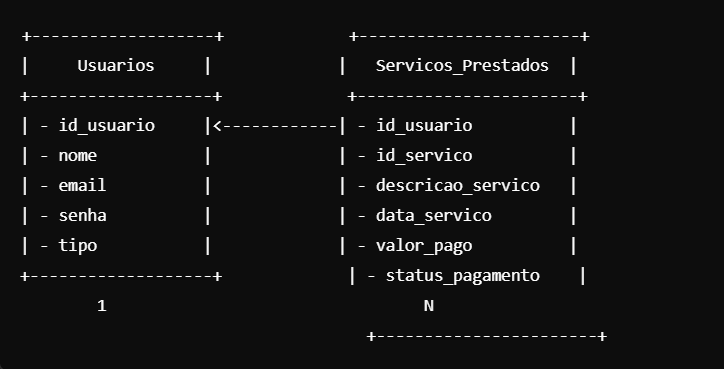
## Usuários e Tarefas: Um usuário pode ser designado para várias tarefas.

**Relacionamento 1** - Entre usuários e tarefas.



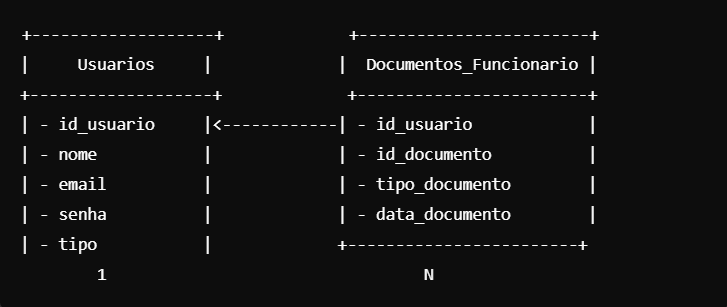
## Usuários e Serviços Prestados: Um usuário pode estar associado a vários serviços prestados.

**Relacionamento 1** - Entre usuários e servicos\_prestados.



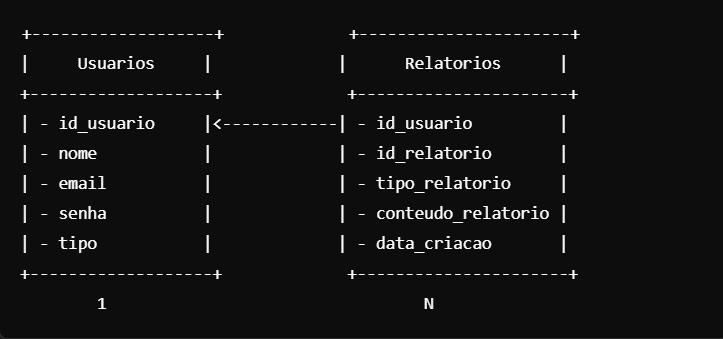
## Usuários e Documentos: Um usuário pode ter vários documentos associados a ele.

**Relacionamento 1** - Entre usuários e documentos\_funcionario.



## Usuários e Relatórios: Um usuário pode ter vários relatórios gerados para ele.

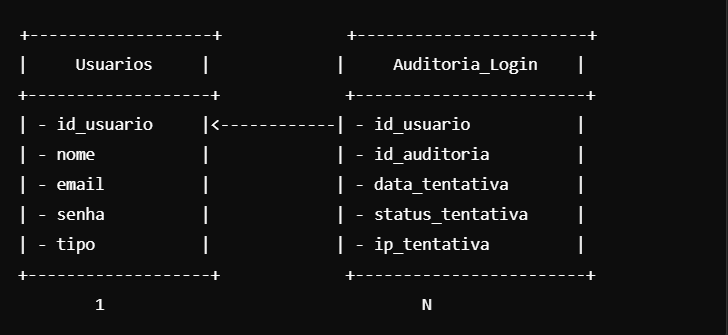
**Relacionamento 1** - Entre usuarios e relatorios.



## Usuários e Auditoria: Cada tentativa de login é registrada, e pode haver múltiplas tentativas de login para um único usuário.

**Relacionamento 1**

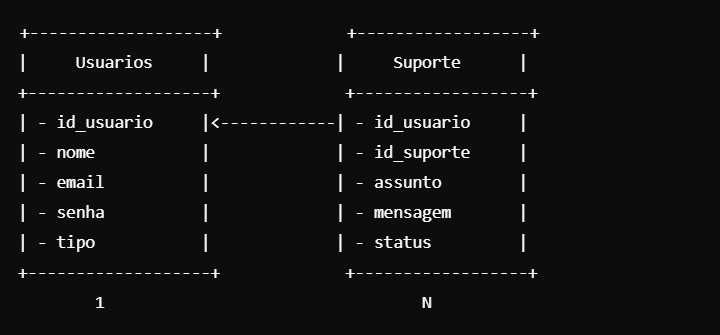
Entre usuarios e auditoria\_login.



## Usuários e Suporte: Um usuário pode enviar várias mensagens de suporte.

**Relacionamento 1**

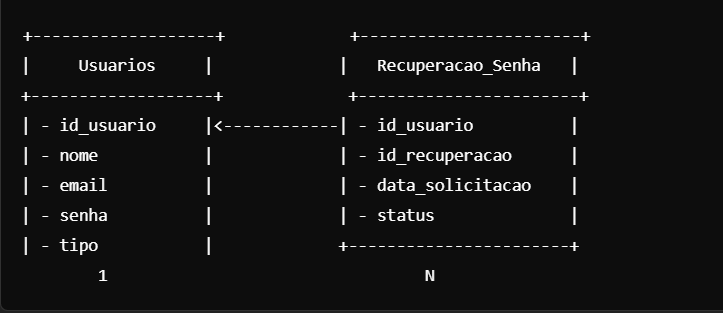
**Entre usuários e suporte.**



## Usuários e Recuperação de Senha: Um usuário pode solicitar a recuperação de senha várias vezes, mas cada solicitação é registrada uma vez.

**Relacionamento 1**

**Entre usuários e recuperacao\_senha.**

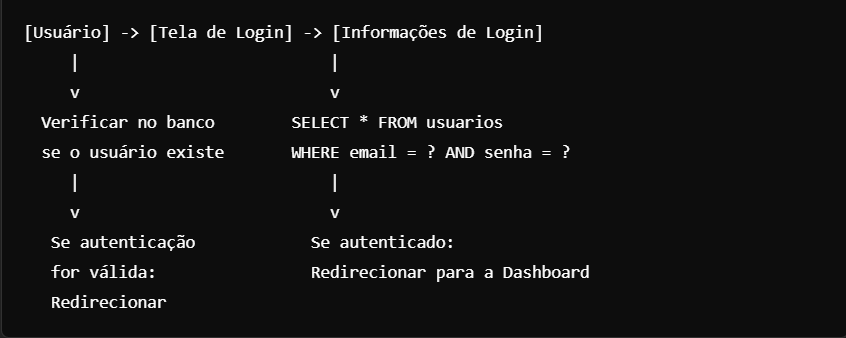


## ****Fluxo de Navegação Completo e Ações no Banco de Dados****

**Login**

O sistema verifica se o usuário existe na tabela usuários e valida a senha. Se autenticado, o usuário é redirecionado para a Dashboard.

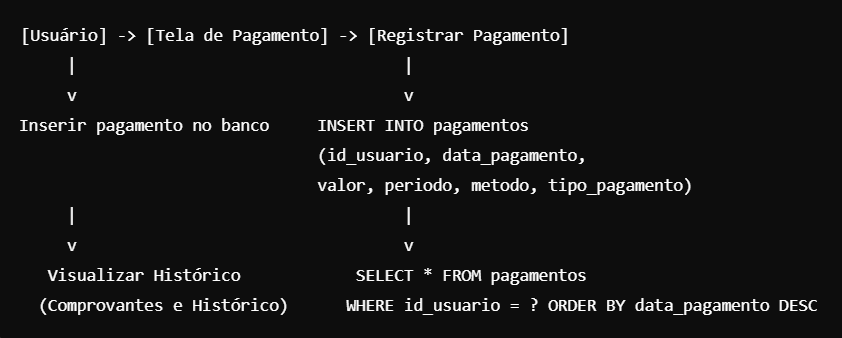
* + Ação no banco: SELECT \* FROM usuários WHERE email = ? AND senha = ?.



## ****Gestão de Pagamentos**:**

Permite registrar pagamentos, visualizar comprovantes e histórico de pagamentos.

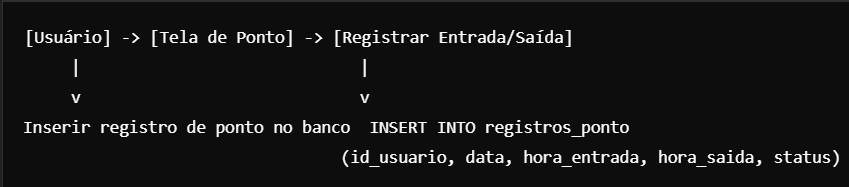
* + Ação no banco: INSERT INTO pagamentos (id\_usuario, data\_pagamento, valor, periodo, metodo, tipo\_pagamento) VALUES (?, ?, ?, ?, ?, ?).
  + Para histórico de pagamentos: SELECT \* FROM pagamentos WHERE id\_usuario = ? ORDER BY data\_pagamento DESC.



## ****Controle de Ponto****:

Registra entradas e saídas. Para cada registro, um novo dado é inserido.

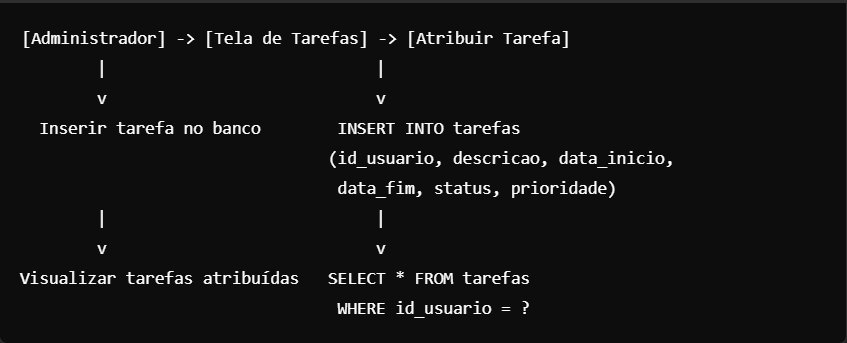
* + Ação no banco: INSERT INTO registros\_ponto (id\_usuario, data, hora\_entrada, hora\_saida, status) VALUES (?, ?, ?, ?, ?).



## ****Tarefas e Cronogramas****:

Os administradores podem atribuir tarefas aos funcionários e acompanhar o progresso.

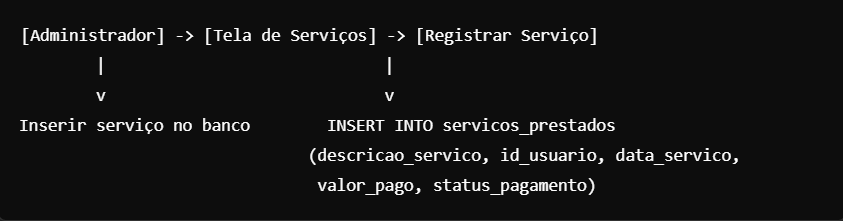
* + Ação no banco: INSERT INTO tarefas (id\_usuario, descricao, data\_inicio, data\_fim, status, prioridade) VALUES (?, ?, ?, ?, ?, ?).
  + Para visualizar tarefas: SELECT \* FROM tarefas WHERE id\_usuario = ?.



## ****Monitoramento de Serviços Prestados****

Registra serviços realizados e associa pagamentos a serviços.

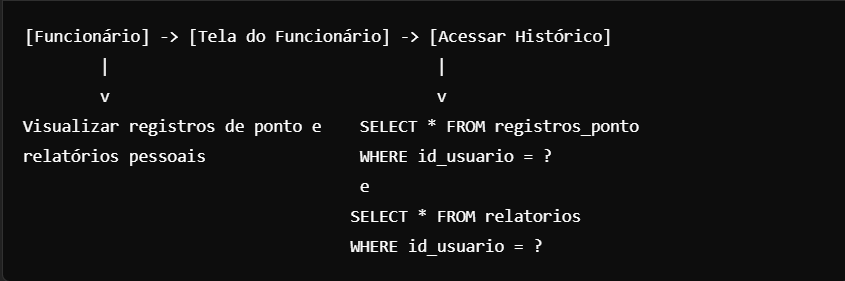
* + Ação no banco: INSERT INTO servicos\_prestados (descricao\_servico, id\_usuario, data\_servico, valor\_pago, status\_pagamento) VALUES (?, ?, ?, ?, ?).



## ****Portal do Funcionário**:**

**O funcionário pode acessar seu histórico e relatórios pessoais.**

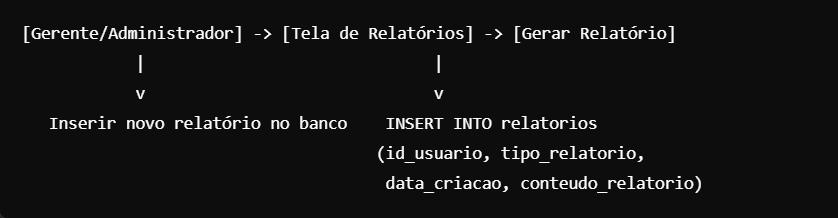
* + Ação no banco: SELECT \* FROM registros\_ponto WHERE id\_usuario = ? e SELECT \* FROM relatorios WHERE id\_usuario = ?.



## ****Relatórios Gerenciais****

Gerentes e administradores podem gerar relatórios financeiros, de produtividade e desempenho.

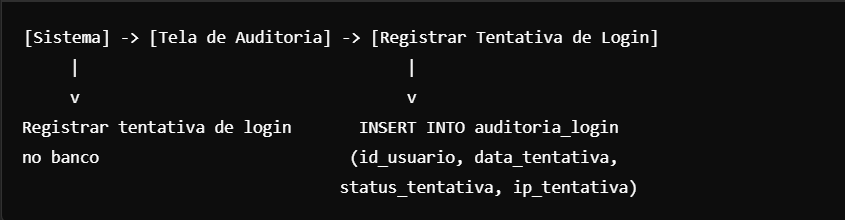
* + Ação no banco: INSERT INTO relatorios (id\_usuario, tipo\_relatorio, data\_criacao, conteudo\_relatorio) VALUES (?, ?, ?, ?).



## ****Configurações e Auditoria**:**

Registro das tentativas de login para auditoria.

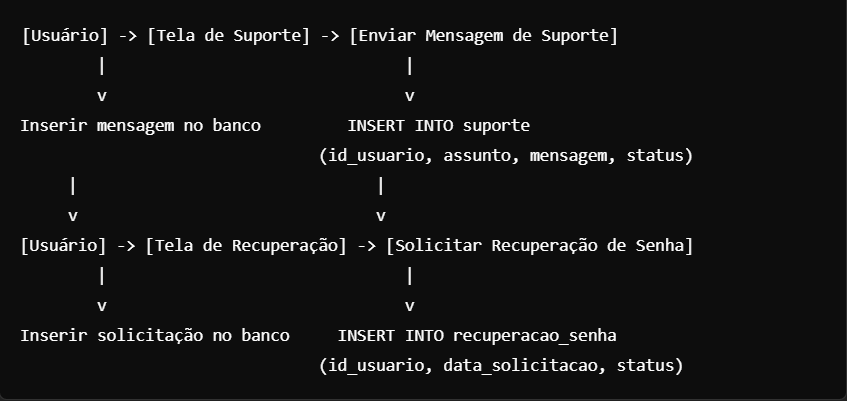
* + Ação no banco: INSERT INTO auditoria\_login (id\_usuario, data\_tentativa, status\_tentativa, ip\_tentativa) VALUES (?, ?, ?, ?).



## ****Suporte e Recuperação de Senha****:

Os usuários podem enviar mensagens de suporte e solicitar recuperação de senha.

* + Ação no banco para suporte: INSERT INTO suporte (id\_usuario, assunto, mensagem, status) VALUES (?, ?, ?, ?).
  + Para recuperação de senha: INSERT INTO recuperacao\_senha (id\_usuario, data\_solicitacao, status) VALUES (?, ?, ?).



# ****Criando o Banco de Dados****

Primeiro passo é criar o banco de dados que será utilizado para o sistema de gerenciamento.

sql

Copiar código

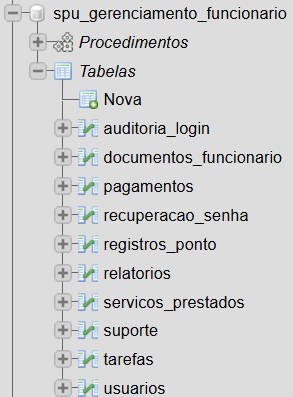
CREATE DATABASE sistema\_gerenciamento;

USE sistema\_gerenciamento;



## Estrutura das Tabelas

A seguir, estão as definições das tabelas que formam a base de dados do sistema. Cada tabela foi criada para suportar as funcionalidades de gestão de usuários, pagamentos, tarefas, controle de ponto e outros recursos.



## Tabela de Usuários

A tabela **usuarios** armazena as informações dos usuários, como nome, e-mail, senha, tipo de usuário e a data de criação do registro.

sql

Copiar código

CREATE TABLE usuarios (

id\_usuario INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

senha VARCHAR(255) NOT NULL,

tipo\_usuario ENUM('funcionario', 'administrador', 'prestador') NOT NULL,

data\_criacao TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

## Tabela de Pagamentos

A tabela **pagamentos** registra os pagamentos feitos aos usuários, associando-os a um usuário e registrando detalhes como data de pagamento, valor e tipo de pagamento.

sql

Copiar código

CREATE TABLE pagamentos (

id\_pagamento INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

data\_pagamento DATE,

valor DECIMAL(10, 2),

periodo ENUM('diario', 'semanal', 'quinzenal', 'mensal'),

metodo ENUM('dinheiro', 'transferencia', 'cartao'),

tipo\_pagamento ENUM('funcionario', 'prestador'),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Registros de Ponto

A tabela **registros\_ponto** armazena os registros de entrada e saída dos funcionários, com a data e o status da presença.

sql

Copiar código

CREATE TABLE registros\_ponto (

id\_registro INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

data DATE,

hora\_entrada TIME,

hora\_saida TIME,

status ENUM('presente', 'ausente', 'atrasado'),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Tarefas

A tabela **tarefas** é utilizada para gerenciar as tarefas atribuídas aos usuários, incluindo descrições, datas de início e fim, e status das tarefas.

sql

Copiar código

CREATE TABLE tarefas (

id\_tarefa INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

descricao TEXT,

data\_inicio DATE,

data\_fim DATE,

status ENUM('pendente', 'em andamento', 'concluida'),

prioridade ENUM('baixa', 'media', 'alta'),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Serviços Prestados

A tabela **servicos\_prestados** armazena informações sobre os serviços prestados pelos usuários e os valores pagos por cada serviço.

sql

Copiar código

CREATE TABLE servicos\_prestados (

id\_servico INT AUTO\_INCREMENT PRIMARY KEY,

descricao\_servico TEXT,

id\_usuario INT,

data\_servico DATE,

valor\_pago DECIMAL(10, 2),

status\_pagamento ENUM('pendente', 'pago', 'não pago'),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Documentos dos Funcionários

A tabela **documentos\_funcionario** é usada para armazenar documentos relacionados aos funcionários, como contratos e identificações.

sql

Copiar código

CREATE TABLE documentos\_funcionario (

id\_documento INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

tipo\_documento VARCHAR(100),

caminho\_arquivo VARCHAR(255),

data\_upload TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Relatórios

A tabela **relatorios** armazena os relatórios gerenciais e pessoais gerados para os usuários.

sql

Copiar código

CREATE TABLE relatorios (

id\_relatorio INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

tipo\_relatorio ENUM('financeiro', 'produtividade', 'desempenho'),

data\_criacao TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

conteudo\_relatorio TEXT,

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Auditoria de Login

A tabela **auditoria\_login** registra as tentativas de login dos usuários para fins de segurança e auditoria.

sql

Copiar código

CREATE TABLE auditoria\_login (

id\_auditoria INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

data\_tentativa TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status\_tentativa ENUM('sucesso', 'falha'),

ip\_tentativa VARCHAR(45),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Suporte

A tabela **suporte** registra as mensagens enviadas pelos usuários solicitando suporte.

sql

Copiar código

CREATE TABLE suporte (

id\_suporte INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

assunto VARCHAR(100),

mensagem TEXT,

status ENUM('pendente', 'em andamento', 'resolvido'),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Tabela de Recuperação de Senha

A tabela **recuperacao\_senha** armazena as solicitações de recuperação de senha feitas pelos usuários.

sql

Copiar código

CREATE TABLE recuperacao\_senha (

id\_recuperacao INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT,

data\_solicitacao TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status ENUM('solicitada', 'concluida'),

FOREIGN KEY (id\_usuario) REFERENCES usuarios(id\_usuario) ON DELETE CASCADE

);

## Procedimentos Armazenados

Os procedimentos armazenados são utilizados para facilitar a interação com o banco de dados e encapsular a lógica de negócios de maneira eficiente.

## Verificação de Login

Este procedimento é responsável por verificar se o e-mail e senha fornecidos por um usuário são válidos.

sql

Copiar código

DELIMITER $$

CREATE PROCEDURE verificar\_login(IN p\_email VARCHAR(100), IN p\_senha VARCHAR(255))

BEGIN

SELECT \*

FROM usuarios

WHERE email = p\_email AND senha = p\_senha;

END $$

DELIMITER ;

## Inserção de Novo Pagamento

Este procedimento insere um novo pagamento realizado a um usuário.

sql

Copiar código

DELIMITER $$

CREATE PROCEDURE inserir\_pagamento(

IN p\_id\_usuario INT,

IN p\_data\_pagamento DATE,

IN p\_valor DECIMAL(10, 2),

IN p\_periodo ENUM('diario', 'semanal', 'quinzenal', 'mensal'),

IN p\_metodo ENUM('dinheiro', 'transferencia', 'cartao'),

IN p\_tipo\_pagamento ENUM('funcionario', 'prestador')

)

BEGIN

INSERT INTO pagamentos (id\_usuario, data\_pagamento, valor, periodo, metodo, tipo\_pagamento)

VALUES (p\_id\_usuario, p\_data\_pagamento, p\_valor, p\_periodo, p\_metodo, p\_tipo\_pagamento);

END $$

DELIMITER ;

## Inserção de Novo Registro de Ponto

Este procedimento insere um novo registro de ponto para um usuário (funcionário).

sql

Copiar código

DELIMITER $$

CREATE PROCEDURE inserir\_registro\_ponto(

IN p\_id\_usuario INT,

IN p\_data DATE,

IN p\_hora\_entrada TIME,

IN p\_hora\_saida TIME,

IN p\_status ENUM('presente', 'ausente', 'atrasado')

)

BEGIN

INSERT INTO registros\_ponto (id\_usuario, data, hora\_entrada, hora\_saida, status)

VALUES (p\_id\_usuario, p\_data, p\_hora\_entrada, p\_hora\_saida, p\_status);

END $$

DELIMITER ;

## Índices para Otimização

Além da criação das tabelas, os índices são importantes para otimizar o desempenho das consultas mais comuns.

## Índice de Usuários por Email

Este índice melhora a performance das buscas por e-mail dos usuários.

sql

Copiar código

CREATE INDEX idx\_email\_usuario ON usuarios(email);

## Índice de Pagamentos por Usuário

Este índice otimiza as consultas para verificar pagamentos de um usuário específico.

sql

Copiar código

CREATE INDEX idx\_usuario\_pagamentos ON pagamentos(id\_usuario);

## Conclusão

Com as tabelas e procedimentos armazenados configurados, o sistema está pronto para realizar operações de gerenciamento de usuários, pagamentos, controle de ponto, tarefas e muito mais. A segurança e a integridade dos dados são garantidas por meio de relacionamentos e restrições apropriadas entre as tabelas.

# ****Arquitetura e Camadas****



## Camada de Apresentação (Frontend)

O sistema foi projetado com uma arquitetura **baseada em API** para permitir a integração com o sistema de ponto existente e outros componentes, como módulos de inteligência artificial para análise de dados. O modelo de arquitetura inclui as seguintes camadas:



A camada de **apresentação** é responsável pela interação do usuário com o sistema. É onde os dados são exibidos ao usuário e as interações com o sistema são realizadas. Ela é composta por páginas web que podem ser acessadas por qualquer navegador moderno.

* **Tecnologias Usadas:** HTML, CSS, JavaScript (frameworks como React ou Vue.js podem ser usados aqui).
* **Responsabilidade:** Exibir informações de forma amigável, como dashboards, históricos de pagamentos, registros de ponto, entre outros. Os dados apresentados são obtidos por meio de chamadas à API backend.



## Camada de Lógica de Negócio (Backend)

A camada **backend** é o coração da lógica do sistema. Ela processa as requisições do usuário, realiza operações de leitura e gravação no banco de dados, e retorna os resultados para a camada de apresentação. A comunicação entre o frontend e o backend ocorre via **APIs RESTful**.

* **Tecnologias Usadas:** PHP (com framework como Laravel ou Slim), Node.js (caso prefira JavaScript no backend), entre outras tecnologias de servidor.
* **Responsabilidade:** Receber requisições HTTP do frontend, processar dados, e executar a lógica de negócios, como cálculo de pagamento, controle de ponto, geração de relatórios e auditoria. Também faz validação dos dados de entrada e retorna respostas adequadas.

## Camada de Persistência (Banco de Dados)

A camada de **persistência** é onde os dados são armazenados. Ela é composta pelo banco de dados, que pode ser acessado diretamente pela camada de backend por meio de SQL.

* **Tecnologias Usadas:** MySQL (ou outro banco relacional, dependendo da necessidade).
* **Responsabilidade:** Armazenar de forma persistente as informações relacionadas aos usuários, pagamentos, tarefas, ponto, serviços prestados, entre outros. O sistema de banco de dados é projetado para garantir a integridade dos dados e otimizar as consultas através do uso de índices e relacionamentos.

## Camada de Integração com Sistemas Externos

Esta camada é responsável por integrar o sistema com **outros sistemas** e serviços externos, como o **sistema de ponto** e módulos de **inteligência artificial** para análise de dados.

* **Tecnologias Usadas:** APIs REST, SOAP, Webhooks, etc.
* **Responsabilidade:** Integrar a aplicação com sistemas de ponto (para registros de entrada/saída dos funcionários), sistemas de pagamento externos (se houver), e integrar módulos de IA para análise de produtividade, segurança, ou outras funcionalidades que dependem de dados externos.

## Descrição das Camadas

**Camada de Apresentação (Frontend)**

A camada de apresentação lida com a interação direta com os usuários. Ela é composta por páginas dinâmicas que exibem informações em tempo real para o usuário e coletam dados de entrada através de formulários, botões, etc. O frontend comunica-se com o backend através de requisições HTTP (AJAX, por exemplo), usando REST APIs para obter dados e enviar comandos de volta ao sistema.

## Fluxo de Trabalho:

1. O usuário interage com a interface gráfica.
2. O frontend envia uma requisição para a API do backend.
3. O backend processa os dados, interage com o banco de dados e retorna uma resposta.
4. O frontend exibe a resposta ao usuário.

## Camada de Lógica de Negócio (Backend)

A camada de backend é responsável pela execução das regras de negócios do sistema. Aqui, são processadas as informações vindas do frontend e do banco de dados, sendo feitas verificações de segurança e validações de dados. Também é responsável por orquestrar a lógica de processos como autenticação de usuários, cálculos de pagamentos, controle de ponto, etc.

## Fluxo de Trabalho:

1. O backend recebe as requisições do frontend.
2. A lógica de negócios é executada (por exemplo, validação de login ou cálculo de pagamento).
3. O backend interage com o banco de dados para armazenar ou recuperar dados.
4. O backend envia uma resposta para o frontend, que a processa e exibe para o usuário.

## Camada de Persistência (Banco de Dados)

A camada de persistência é o sistema de armazenamento de dados, onde todas as informações são armazenadas de forma organizada e estruturada. Os dados podem ser acessados e manipulados através de comandos SQL.

## Fluxo de Trabalho:

1. O backend solicita dados ao banco de dados (consultas SQL).
2. O banco de dados executa as operações solicitadas (inserção, atualização, exclusão, etc.).
3. O banco de dados retorna os dados ou confirma a operação para o backend.
4. O backend processa a resposta e envia para o frontend.

## Camada de Integração com Sistemas Externos

Essa camada é dedicada às interações com outros sistemas, como o sistema de ponto existente ou módulos de análise de dados que utilizam inteligência artificial. A comunicação com esses sistemas é feita através de APIs RESTful ou outras tecnologias de integração, como SOAP ou Webhooks.

## Fluxo de Trabalho:

1. O sistema recebe dados de sistemas externos via APIs.
2. O sistema processa esses dados (por exemplo, calcula o ponto ou faz análise de dados de produtividade).
3. O sistema envia dados de volta para os sistemas externos ou os apresenta ao usuário.

.

# CODIGOS

1. Estrutura de organização dos códigos:

Imagem

# Procedimentos de Instalação

# Testes

# Manutenção e Suporte

# Anexos

# Referências

# Aprovações

|  |  |  |
| --- | --- | --- |
| **Aprovações** | | |
| **Participante** | **Assinatura** | **Data** |
| Gabriela Galucio Cardoso |  |  |
| Carlos Henrique Farias Malcher |  |  |
| Bruna Lauren |  |  |
| Liria Repolho |  |  |
| Proprietario da Empresa |  |  |
| Professor Responsavel |  |  |