ESCOLA SENAI

“PROF. DR. EURYCLIDES DE JESUS ZERBINI”

Douglas Oliveira da Silva

Júlio César da Silva

Luís Henrique Ferreira de Souza

Valmir Soares

**REPAIRIQ:**

Aplicativo de gerenciamento de ordens de serviço

Campinas SP

2025

Douglas Oliveira da Silva

Júlio César da Silva

Luís Henrique Ferreira de Souza

Valmir Soares

**RepairIQ:**

Aplicativo de gerenciamento de ordens de serviço

Projeto apresentado à Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini” para obtenção do certificado de conclusão do Curso Técnico de Desenvolvimento de Sistemas.

Orientador: Douglas de Cassio Quinzani Gaspar, Renato de Mattos Onofre.

Campinas SP

2025

Douglas Oliveira da Silva

Júlio César da Silva

Luís Henrique Ferreira de Souza

Valmir Soares

**RepairIQ:**

Trabalho de conclusão de curso aprovado como requisito parcial para obtenção do grau de técnico, do curso Técnico de Desenvolvimento de Sistemas da Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini”

**BANCA EXAMINADORA**

1º Examinador

2º Examinador

3º Examinador

Campinas SP

2025

**DEDICATÓRIA**

Dedico este projeto a todos os profissionais de manutenção que fazem a diferença todos os dias e à equipe que acreditou na ideia e fez acontecer.

**AGRADECIMENTOS**

Agradecemos à nossa equipe pelo comprometimento, foco e entrega em cada etapa deste projeto.

Trabalhamos como um só, transformando ideias em soluções e superando desafios com colaboração e clareza.

Agradecemos também aos professores Renato de Mattos Onofre e Douglas de Cassio Quinzani Gaspar pelo conhecimento, orientação e incentivo constantes, fundamentais para a realização deste trabalho.

Este projeto é o reflexo da nossa capacidade de construir juntos, com propósito e excelência.

**EPÍGRAFE**

**“A inovação acontece.**

**Você tem duas opções:**

**Protagonizar ou testemunhar”.****Igor Marinelli**

**SUMÁRIO**

[1 INTRODUÇÃO 10](#_Toc14160040)

[2 JUSTIFICATIVA 11](#_Toc14160041)

[3 OBJETIVOS 12](#_Toc14160042)

[3.1. Objetivos Gerais 12](#_Toc14160043)

[3.2. Objetivos Específicos 12](#_Toc14160044)

[4 PRODUCT BACKLOG 13](#_Toc14160045)

[5 REQUISITOS NÃO FUNCIONAIS 13](#_Toc14160046)

[6 PREMISSAS 14](#_Toc14160047)

[7 RESTRIÇÕES 15](#_Toc14160048)

[8 ANÁLISE DE RISCOS DE UM PROJETO 16](#_Toc14160049)

[8.1. Nível e Planos de Ação para os Riscos 16](#_Toc14160050)

[8.2. Planos de ação 16](#_Toc14160051)

[9 SPRINTS 17](#_Toc14160052)

[9.1. Primeiro Sprint 17](#_Toc14160053)

[9.1.1. Product Backlog 17](#_Toc14160054)

[9.1.2. Sprint Backlog 17](#_Toc14160055)

[9.1.3. Burn Down Chart 17](#_Toc14160056)

[9.1.4. Diagramas 17](#_Toc14160057)

[9.1.5. Plano de testes 17](#_Toc14160058)

[9.1.5.1. Resultados 17](#_Toc14160059)

[9.1.6. Kanban e Retrospectiva 17](#_Toc14160060)

[10 Modelo de Dados 18](#_Toc14160061)

[10.1. Diagrama de Entidade e Relacionamento 18](#_Toc14160062)

[10.2. Modelo lógico do banco de dados 18](#_Toc14160063)

[10.3. Dicionário de dados 18](#_Toc14160064)

[11 PRINCIPAIS TELAS DO SISTEMA 19](#_Toc14160065)

[12 CONCLUSÃO 20](#_Toc14160066)

[12.1. Escreva os resultados obtidos 20](#_Toc14160067)

[12.2. Constatações 20](#_Toc14160068)

[12.3. Sugestões de possíveis aperfeiçoamentos técnicos 20](#_Toc14160069)

[13 REFERÊNCIAS 21](#_Toc14160070)

[14 GLOSSÁRIO 22](#_Toc14160071)

[15 ANEXOS 23](#_Toc14160072)

# INTRODUÇÃO

A manutenção eficiente é essencial para o bom funcionamento de qualquer empresa. Com isso em mente, desenvolvemos o RepairIQ, uma solução inovadora que revoluciona a forma como as ordens de serviço é gerenciada, tornando o processo mais ágil, preciso e inteligente.

Nosso aplicativo permite que funcionários abram ordens de serviço rapidamente, fornecendo detalhes completos sobre o equipamento e seus componentes. A equipe de manutenção recebe notificações em tempo real e tem acesso imediato às informações necessárias para realizar o reparo com eficiência.

Além disso, o RepairIQ vai além do gerenciamento básico: utilizando inteligência artificial, ele analisa todas as ordens de serviço já realizadas para identificar padrões, prever falhas futuras e recomendar ações preventivas, reduzindo o tempo de inatividade dos equipamentos e otimizando os custos operacionais.

Com uma interface intuitiva e recursos avançados, o RepairIQ transforma a manutenção corretiva, preventiva e preditiva em um processo mais estratégico e eficaz, garantindo maior produtividade e segurança para sua empresa.

O futuro da manutenção começa agora. Conheça o RepairIQ e leve sua gestão de manutenção para o próximo nível!

# JUSTIFICATIVA

O app REPAIRIQ, é um aplicativo voltado para a área de manutenção industrial, com as seguintes finalidades:

* Maior controle de Ordens de Serviços.
* Melhor ferramenta para os técnicos de campo.
* Ferramenta adequada para a supervisão monitorar atividades diárias.
* Atividades serão encaminhadas para cada área especifica.
* Tempo de resposta mais ágil.

Atualmente, a gestão de Ordens de Serviço (OS) na área de manutenção passa por uma transformação significativa impulsionada pela digitalização, automação e integração de novas tecnologias. As empresas estão cada vez mais adotando sistemas de gestão assistidos por computador (CMMS) para melhorar a criação, rastreamento e controle das Ordens de Serviço, oferecendo maior precisão e eficiência. Além disso, a manutenção preditiva, auxiliada por sensores IoT (Internet das coisas), permite a emissão de ordens com base em dados reais de desempenho dos equipamentos, ajudando a prevenir falhas antes que ocorram e a reduzir paradas não programadas.

A mobilidade também tem sido um fator-chave, com a utilização de aplicativos móveis permitindo que os técnicos recebam e atualizem ordens de serviço em tempo real, o que aumenta a produtividade e a comunicação. O uso de inteligência artificial e machine learning também está ajudando a otimizar o planejamento e a alocação de recursos de manutenção, criando uma gestão mais eficiente e econômica.

Outro avanço importante é a integração das OSs com outras áreas da empresa, como compras, inventário e finanças, proporcionando maior controle de custos e transparência. Além disso, a gestão de prioridades e a implementação de acordos de nível de serviço (SLAs) têm garantido um atendimento mais eficiente e a satisfação dos clientes.

No entanto, a área ainda enfrenta desafios, como a capacitação de profissionais qualificados para operar as novas tecnologias, a adaptação de sistemas complexos em empresas resistentes à mudança e o controle de custos operacionais sem comprometer a qualidade e segurança. Apesar desses desafios, a tendência é que a manutenção continue a se modernizar com soluções tecnológicas, buscando maior automação, eficiência e sustentabilidade.

# OBJETIVOS

Aqui vai o texto de objetivos

O **REPAIRIQ** é um aplicativo inovador voltado para a área de **manutenção industrial**, com o objetivo de otimizar e modernizar a gestão de **Ordens de Serviço (OS)**. Através de funcionalidades específicas, o app busca atender tanto às necessidades operacionais quanto estratégicas das empresas, oferecendo:

* **Maior controle de Ordens de Serviço**: Facilitando o acompanhamento e a gestão de tarefas de manutenção em tempo real, com maior precisão e agilidade.
* **Ferramenta otimizada para técnicos de campo**: Permitindo que os técnicos recebam, atualizem e concluam ordens de serviço diretamente em seus dispositivos móveis, aumentando a produtividade e a comunicação durante as operações.
* **Monitoramento eficaz para supervisores**: Proporcionando aos supervisores uma visão clara e em tempo real das atividades diárias, garantindo o cumprimento de prazos e a correta alocação de recursos.
* **Encaminhamento das atividades para áreas específicas**: As ordens de serviço são automaticamente direcionadas para as equipes ou áreas responsáveis, evitando erros de alocação e melhorando a eficiência operacional.
* **Tempo de resposta ágil**: O app garante que as ordens sejam respondidas rapidamente, com uma interface intuitiva e funcionalidades que reduzem o tempo entre a solicitação e a execução da manutenção.

Além disso, o **REPAIRIQ** se alinha com as tendências atuais da **manutenção digital**. Ele integra tecnologias emergentes como a **IoT** (Internet das Coisas), permitindo que a manutenção preditiva seja realizada com base em dados reais do desempenho dos equipamentos, o que minimiza falhas inesperadas e paradas não programadas. A **mobilidade** é um ponto-chave, facilitando o envio e o acompanhamento das ordens de serviço em tempo real, além de ser uma plataforma integrada que comunica diferentes áreas da empresa, como compras, inventário e finanças, promovendo maior controle e transparência nos processos.

Com a crescente digitalização e a introdução de tecnologias como **inteligência artificial** e **machine learning**, o **REPAIRIQ** contribui para um planejamento de manutenção mais eficiente e econômico, sempre com foco na **sustentabilidade** e **eficiência energética**. Em um cenário onde a adaptação tecnológica é cada vez mais necessária, o app oferece uma solução robusta, inteligente e intuitiva para a manutenção industrial, garantindo maior **automação** e **sustentabilidade** nas operações.

# Objetivos Gerais

Aqui vai o texto de objetivos gerais

 **Automatizar a Gestão de Ordens de Serviço (OS)**: Tornar o processo de criação, rastreamento e conclusão das OSs mais ágil e preciso, eliminando a gestão manual.

 **Implementar Manutenção Preditiva**: Integrar sensores IoT para gerar ordens de serviço com base no estado real dos equipamentos, prevenindo falhas e paradas inesperadas.

 **Facilitar o Monitoramento em Tempo Real**: Proporcionar aos supervisores visibilidade instantânea das atividades, status das tarefas e alocação de recursos.

 **Otimizar o Trabalho dos Técnicos de Campo**: Permitir que técnicos recebam e atualizem ordens de serviço diretamente em seus dispositivos móveis, aumentando a produtividade e comunicação.

 **Encaminhamento Automático de Atividades**: Direcionar automaticamente as ordens de serviço para as equipes corretas, garantindo alocação eficiente de tarefas.

 **Reduzir o Tempo de Resposta**: Agilizar o tempo entre a solicitação e a execução da manutenção, melhorando a eficiência operacional.

 **Integrar com Outras Áreas da Empresa**: Facilitar a comunicação entre manutenção, compras, inventário e finanças, proporcionando maior controle de custos e planejamento.

 **Acompanhar KPIs e SLAs**: Monitorar o cumprimento de prazos e padrões de qualidade, garantindo eficiência e satisfação.

 **Reduzir Erros e Retrabalho**: Minimizar falhas manuais e o risco de retrabalho, aumentando a eficiência e diminuindo custos operacionais.

 **Focar em Sustentabilidade**: Identificar oportunidades para melhorar a eficiência energética e reduzir impactos ambientais nos processos de manutenção.

# Objetivos Específicos

Aqui vai o texto de objetivos específicos

# PRODUCT BACKLOG

Site:

* RF01 - Cadastro de ordem de serviço.
* RF01 - Atualização de ordem de serviço.
* RF01 - Remoção de ordem de serviço.
* RF01 - Busca de ordem de serviço.
* RF01 - Busca de ordem de serviço por data.
* RF01 - Busca de ordem de serviço por setor.
* RF01 - Busca de ordem de serviço por equipamento.
* RF01 - Cadastro de grupo de usuário.
* RF01 - Atualização de grupo de usuário.
* RF01 - Remoção de grupo de usuário.
* RF01 - Busca de grupo de usuário.
* RF01 - Cadastro de usuário.
* RF01 - Atualização de usuário.
* RF01 - Remoção de usuário.
* RF01 - Busca de usuário.
* RF01 - Cadastro de modelo de ordem de serviço.
* RF01 - Atualização de modelo de ordem de serviço.
* RF01 - Remoção de modelo de ordem de serviço.
* RF01 - Busca de modelo de ordem de serviço.
* RF01 - Cadastro de ocorrências.
* RF01 - Atualização de ocorrências.
* RF01 - Remoção de ocorrências.
* RF01 - Busca de ocorrências.
* RF01 - Cadastro de equipamentos.
* RF01 - Atualização de equipamentos.
* RF01 - Remoção de equipamentos.
* RF01 - Busca de equipamentos.
* RF02 – Notificar usuários sobre ocorrências de acordo com a responsabilidade deles;
* RF03 – Pesquisa personalizada de Ordens de Serviço, Grupos de Usuários, Usuários, Ocorrências;
* RF04 – Apresentar visões gerais sobre os equipamentos/ordens de serviço do cliente quando ele possuir dados o suficiente sobre as máquinas;
* RF05 – Permitir o gerenciamento das Ordens de Serviço, Grupos de Usuários, Usuários, Modelo de Ordem de Serviço, Ocorrências, Modelo de Ocorrências de acordo com as permissões do usuário cadastrado;
* RF06 – Permitir o agendamento de manutenções preventivas e periódicas de equipamentos ou serviços, com alertas para os técnicos e gestores.
* RF07 – Implementar um painel de métricas e KPIs (Key Performance Indicators) para monitorar o desempenho da equipe de manutenção, tempo médio de reparo, custos e disponibilidade de equipamentos.
* RF08 – Oferecer relatórios personalizáveis para análise de tendências de falhas, custos de manutenção e eficiência das ações preventivas/preditivas.
* RF09 – Permitir a exportação de dados em formatos como CSV, PDF e Excel para facilitar a análise externa.
* RF10 – Implementar um sistema de priorização automática de ordens de serviço com base na criticidade do equipamento e no impacto operacional.
* RF11 – Permitir que os usuários visualizem em tempo real o status de equipamentos em operação, com indicadores de desempenho e possíveis alertas para falhas iminentes, melhorando a antecipação de problemas.

Servidor:

* RF01 – Criar cinco tabelas (empresa\_usuarios, empresa\_modelo\_os, empresa\_os, empresa\_modelo\_ocorrencia, empresa\_ocorrencia) em nossos servidores programaticamente para a empresa que contratar nossos serviços;
* RF02 – Sistema de Login utilizando SHA-256 para salvar a senha do usuário, tokens para gerenciar os logins já feitos e sua vida útil;
* RF03 – Enviar todos os dados novos sobre uma empresa para uma IA (a ser definida) para dar sugestões nas ações corretivas, preditivas, preventivas;
* RF04 – Implementar um sistema de backup automático e recuperação de dados para garantir a segurança e disponibilidade das informações.
* ~~RF05 – Criar uma API RESTful para integração com outros sistemas e dispositivos IoT (Internet das Coisas) para coleta automática de dados de sensores e equipamentos.~~ (Necessário revisar e ver se vai dar tempo)
* RF07 – Permitir a configuração de regras personalizadas para notificações (ex.: notificar apenas falhas críticas fora do horário comercial).

Android:

* RF01 – Operações CRUD para as Ordens de Serviço, Ocorrências;
* RF02 – Notificar usuários sobre ocorrências de acordo com a responsabilidade deles;
* RF03 – Pesquisa personalizada de Ordens de Serviço, Ocorrências;
* RF04 – Permitir que os usuários possam anexar imagens, vídeos ou documentos nas ordens de serviço, para facilitar o registro e acompanhamento do processo.
* RF05 – Gerar QR Codes para cada máquinas, permitindo que os usuários acessem rapidamente os detalhes da ordem escaneando o código com o celular.
* ~~RF06 – Integrar a gestão de materiais necessários para cada ordem de serviço, com controle de estoque.~~ (Necessário revisar e ver se vai dar tempo)
* ~~RF07 – Incluir suporte a múltiplos idiomas no aplicativo Android, permitindo que usuários de diferentes localidades possa, utilizar o sistema na língua de sua preferência. (Necessário revisar e ver se vai dar tempo)~~
* RF08 – Implementar funcionalidade de geolocalização para identificar a localização do técnico e associá-la à ordem de serviço.
* RF09 – Oferecer um modo offline para criação e edição de ordens de serviço, com sincronização automática quando houver conexão com a internet.
* RF10 – ~~Permitir que os usuários visualizem em tempo real o status de equipamentos em operação, com indicadores de desempenho e possíveis alertas para falhas iminentes, melhorando a antecipação de problemas.~~
* RF11 – Permitir a assinatura digital do técnico ao finalizar uma ordem de serviço, garantindo a autenticidade do registro.

Inteligência Artificial:

* RF01 – Desenvolver um sistema de recomendação de peças de reposição com base no histórico de falhas e manutenções.
* RF02 – Criar um modelo preditivo para estimar a vida útil restante de equipamentos com base em dados históricos e padrões de uso.
* RF03 – Implementar um sistema de detecção de anomalias em tempo real, utilizando dados de sensores e histórico de manutenção.
* RF04 – Oferecer sugestões de melhorias nos processos de manutenção com base em análises de eficiência e custos.

# REQUISITOS NÃO FUNCIONAIS

Aqui vai o texto dos requisitos não funcionais do projeto

* RNF01 – O sistema deve suportar até 10.000 usuários simultâneos sem degradação significativa do desempenho.
* RNF02 – O sistema deve ter um tempo de resposta máximo de 2 segundos para a execução de operações de consulta e busca de dados.
* RNF03 – O sistema deve garantir 99,9% de disponibilidade, com tempo de inatividade máximo de 8 horas por ano.
* RNF04 – O sistema deve ser escalável horizontalmente para suportar aumento no número de usuários ou volume de dados.
* RNF05 – O sistema deve ser compatível com os principais navegadores da web, como Chrome, Firefox, Safari e Edge.
* RNF06 – O sistema deve garantir a integridade e segurança dos dados, utilizando criptografia SSL/TLS para todas as comunicações entre o servidor e os dispositivos de acesso.
* RNF07 – O sistema deve ser compatível com as versões mais recentes dos sistemas operacionais Android e iOS para garantir a maior base de usuários possível.
* RNF08 – O sistema deve ter um processo de autenticação seguro, com políticas de senhas fortes e verificação de múltiplos fatores (MFA) para usuários administrativos.
* RNF09 – O sistema deve ser capaz de realizar backups diários automáticos e garantir a restauração de dados de forma rápida e eficiente.

# PREMISSAS

Premissas são os fatores associados ao escopo do projeto que, para fins de planejamento, são assumidos como verdadeiros, reais ou certos sem a necessidade de prova ou demonstração. Ou seja, são hipóteses ou pressupostos.

Exemplos:

* O RepairIQ será desenvolvido e implantado utilizando infraestrutura em nuvem, garantindo disponibilidade e resiliência contra falhas de hardware e falta de energia.
* A equipe de desenvolvimento contará com no mínimo 4 desenvolvedores atuando em tempo integral durante todo o ciclo de desenvolvimento do projeto.
* O uso de metodologias ágeis será adotado, com reuniões regulares de acompanhamento do progresso e refinamento contínuo do backlog.
* Todos os desenvolvedores seguirão padrões de codificação bem definidos e realizarão code reviews obrigatórios para garantir a qualidade do código.
* O sistema será otimizado para alto desempenho, incluindo testes de carga e análise contínua da performance.
* Serão utilizados mecanismos de controle de versionamento (Git) e boas práticas de gerenciamento de branches para evitar problemas com código mal estruturado.
* A equipe receberá treinamentos sobre o uso adequado das ferramentas de desenvolvimento e metodologias ágeis para minimizar riscos operacionais.
* O RepairIQ contará com dashboards e relatórios para acompanhamento em tempo real das ordens de serviço e desempenho do sistema.
* Será implementado um plano de contingência para mitigar a dependência de um único desenvolvedor em partes críticas do código.

# RESTRIÇÕES

Restrições são limitações impostas interna ou externamente ao projeto. Restrições podem ser a obrigatoriedade de se executar determinadas ferramentas e a forma de trabalho da equipe.

Exemplos:

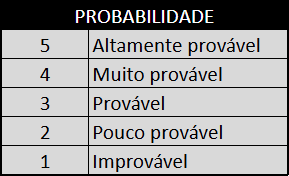
* RT001: Acesso à internet.
* RT002: Versão do Android Mínima 8.1.
* RT003: Solicitar ao usuário a permissão de acesso da câmera, armazenamento e localização do dispositivo durante a execução do aplicativo (Android):
* RT003.1: Acesso a câmera do dispositivo Android;
* RT003.2: Acesso ao armazenamento do dispositivo Android;
* RT003.3: Acesso à localização do dispositivo Android;
* RT004: Versão Android Studio Lady Bug 2024.2.1

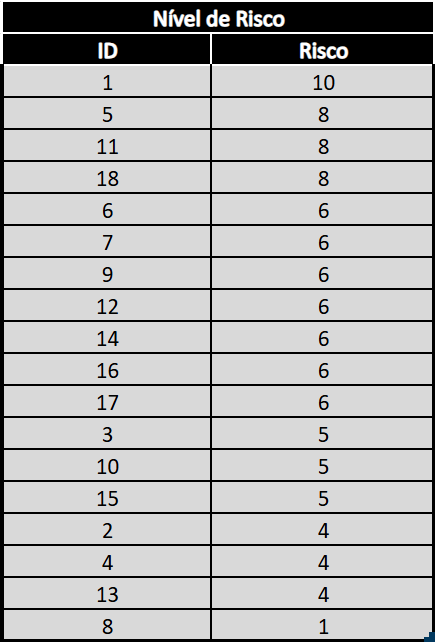
# ANÁLISE DE RISCOS DE UM PROJETO



# Nível e Planos de Ação para os Riscos

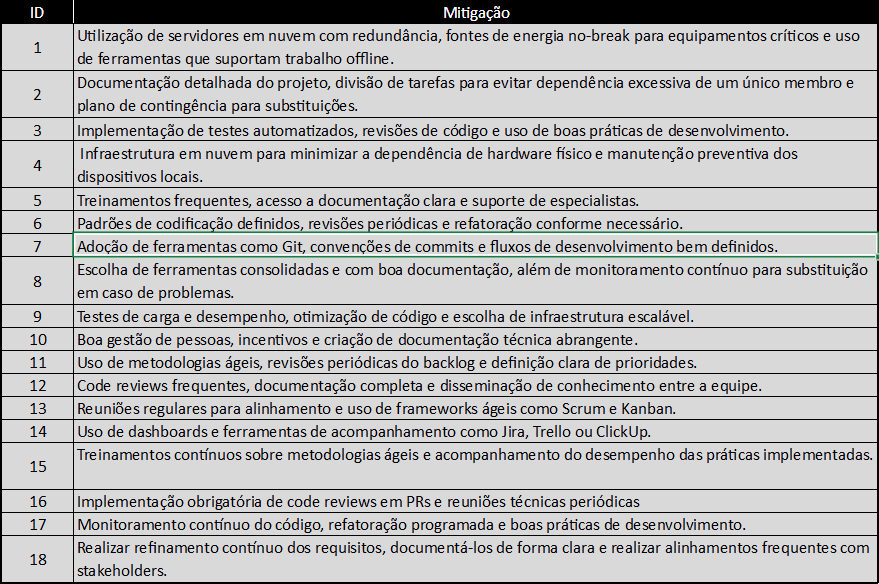
* 1. Definimos uma hierarquia do nível dos riscos, do mais grave para o menos grave. Assim, damos uma maior atenção ás ameaças com maior impacto e probabilidade de acontecer.



# Planos de ação

Planos de ação para os riscos referente à tabela de riscos gerais



# SPRINTS

# Primeiro Sprint

O primeiro Sprint foi idealizado principalmente para a realização da documentação referente ao projeto.

# Product Backlog

[Caso houver, apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) as alterações que foram feitas no product backlog.

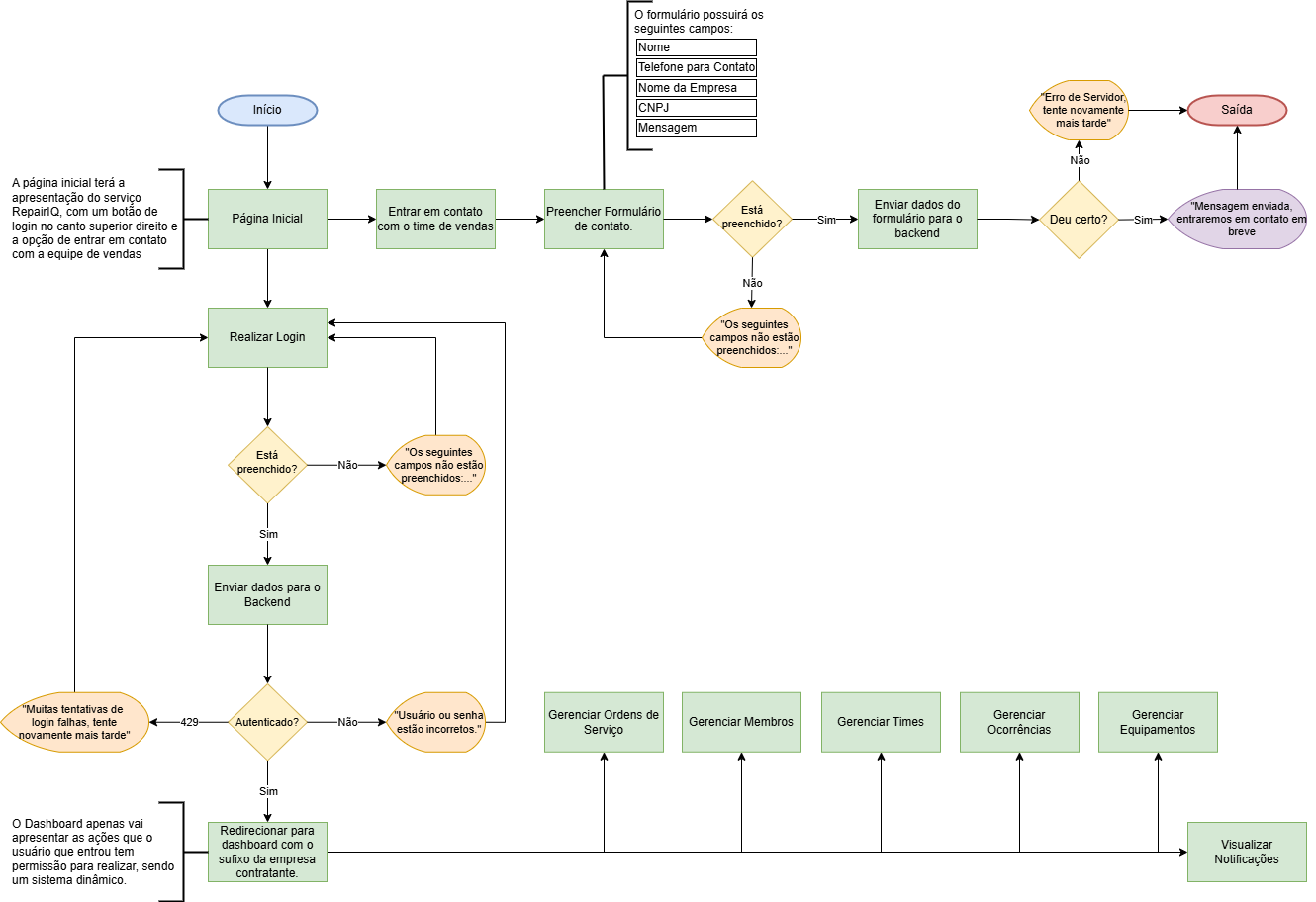
# Sprint Backlog

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128" \t "_blank) as histórias selecionadas para esse Sprint.

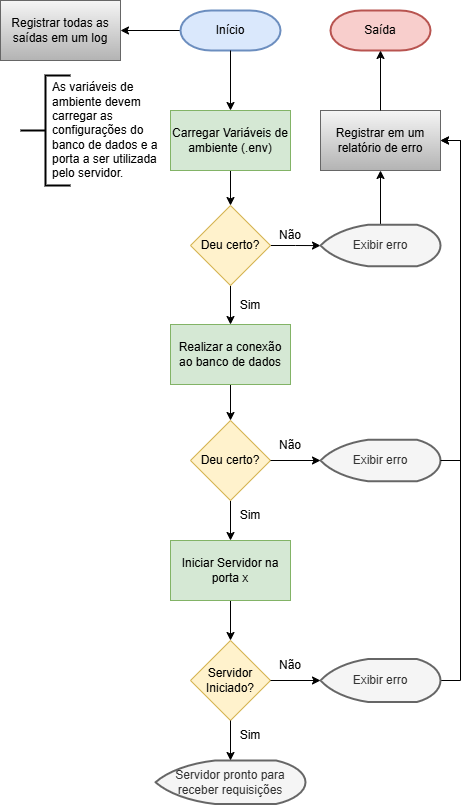
# Burn Down Chart

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128" \t "_blank) o burn down chart do Sprint

# Diagramas

Fluxograma Front-end.

Fluxograma Back-end.



# Plano de testes

O plano de teste é uma maneira de encontrar defeitos e bugs no sistema para serem futuramente corrigidos. Mesmo que durante a programação sempre foram realizados alguns, ainda sim no plano são realizados testes finais passo a passo.

# Resultados

Resultados apresentados nos planos de testes

# Kanban e Retrospectiva

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) imagens do quadro de Kanban e as conclusões da retrospectiva.

# Segundo Sprint

# Product Backlog

[Caso houver, apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) as alterações que foram feitas no product backlog.

# Sprint Backlog

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) as histórias selecionadas para esse Sprint.

# Burn Down Chart

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) o burn down chart do Sprint.

# Diagramas

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) os principais diagramas realizados.

# Plano de testes

O plano de teste é uma maneira de encontrar defeitos e bugs no sistema para serem futuramente corrigidos. Mesmo que durante a programação sempre foram realizados alguns, ainda sim no plano são realizados testes finais passo a passo.

# Resultados

Resultados apresentados nos planos de testes

# Kanban e Retrospectiva

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) imagens do quadro de Kanban e as conclusões da retrospectiva.

# Terceiro Sprint

# Product Backlog

[Caso houver, apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) as alterações que foram feitas no product backlog.

# Sprint Backlog

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) as histórias selecionadas para esse Sprint.

# Burn Down Chart

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) o burn down chart do Sprint.

# Diagramas

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) os principais diagramas realizados.

# Plano de testes

O plano de teste é uma maneira de encontrar defeitos e bugs no sistema para serem futuramente corrigidos. Mesmo que durante a programação sempre foram realizados alguns, ainda sim no plano são realizados testes finais passo a passo.

# Resultados

Resultados apresentados nos planos de testes

# Kanban e Retrospectiva

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) imagens do quadro de Kanban e as conclusões da retrospectiva.

# Quarto Sprint

# Product Backlog

[Caso houver, apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) as alterações que foram feitas no product backlog.

# Sprint Backlog

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) as histórias selecionadas para esse Sprint.

# Burn Down Chart

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) o burn down chart do Sprint.

# Diagramas

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) os principais diagramas realizados.

# Plano de testes

O plano de teste é uma maneira de encontrar defeitos e bugs no sistema para serem futuramente corrigidos. Mesmo que durante a programação sempre foram realizados alguns, ainda sim no plano são realizados testes finais passo a passo.

# Resultados

Resultados apresentados nos planos de testes

# Kanban e Retrospectiva

[Apresenta](http://www.devmedia.com.br/curso/introducao-a-uml/128) imagens do quadro de Kanban e as conclusões da retrospectiva.

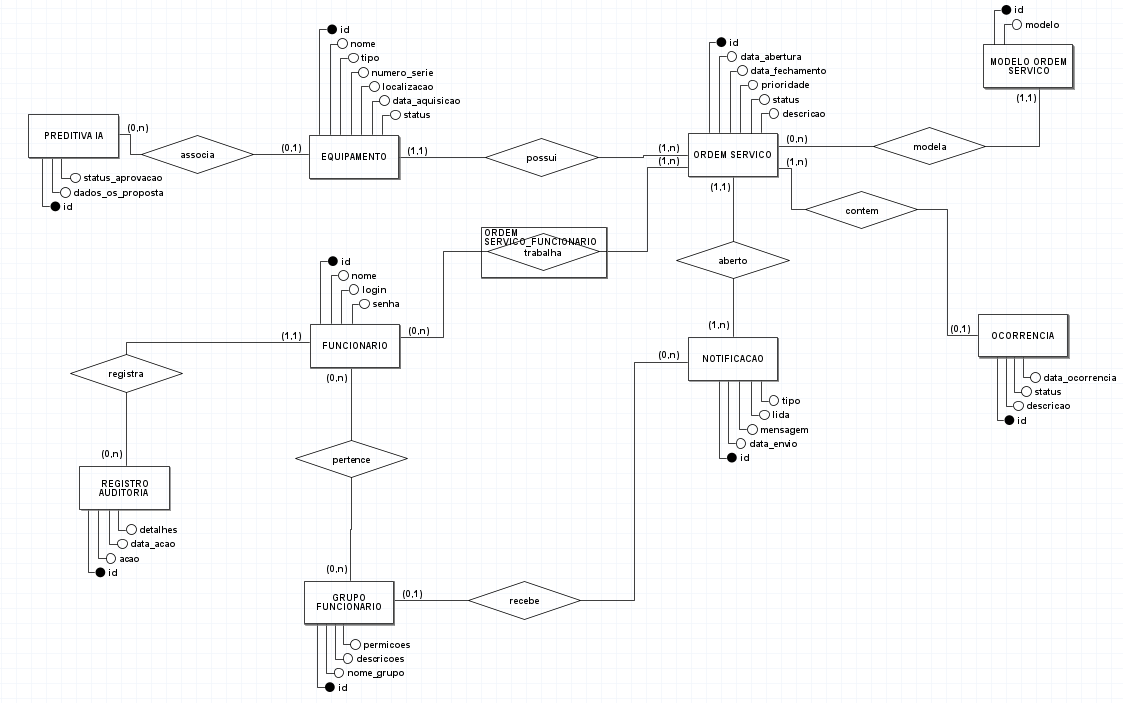
# Modelo de Dados

Esta parte do planejamento traz informações necessárias para a construção de um banco de dados para o Sistema de Gerenciamento de Acessos.

# Diagrama de Entidade e Relacionamento

Nessa etapa se define: as entidades necessárias para a construção do Banco de Dados; Os relacionamentos e o seu grau, ou seja, a quantidade de entidades que estão ligadas ao relacionamento.

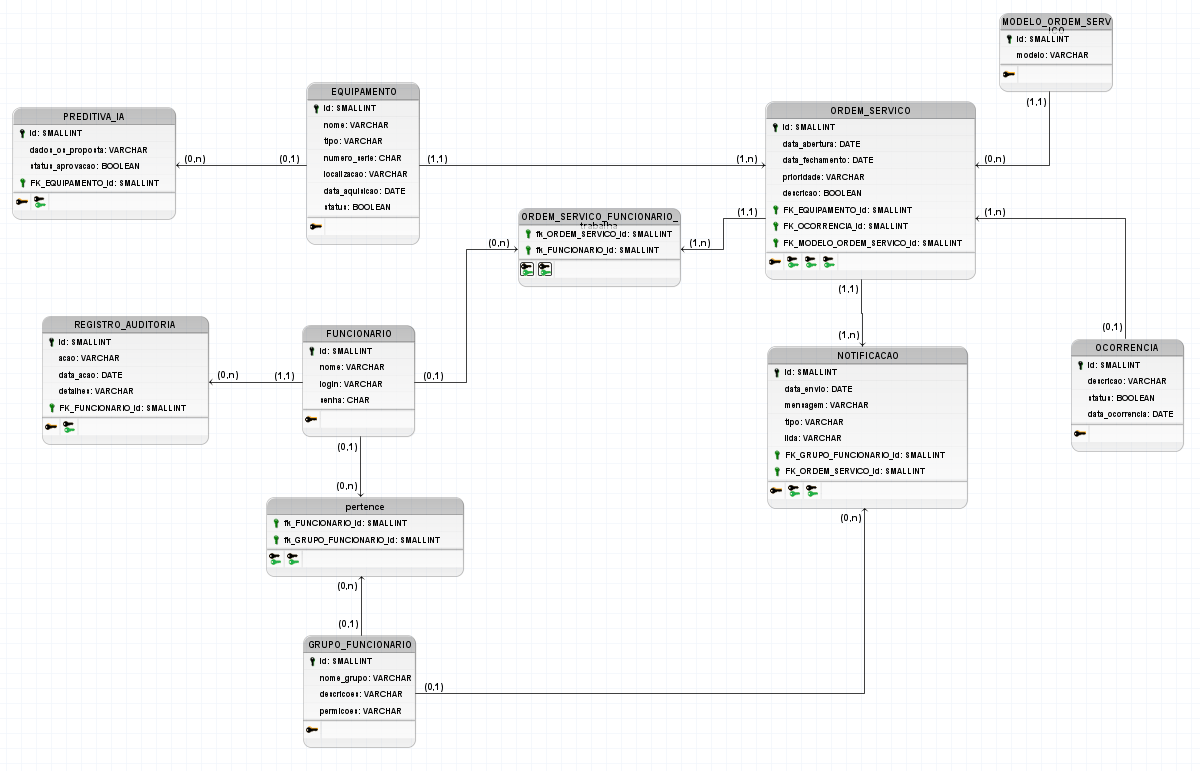
**MODELO CONCEITUAL**



# Modelo lógico do banco de dados

Nessa etapa se define: os atributos pertencentes a cada entidade; as chaves primárias e estrangeiras; o tipo de cada campo e valor de determinados campos.

**MODELO LÓGICO**



# Dicionário de dados

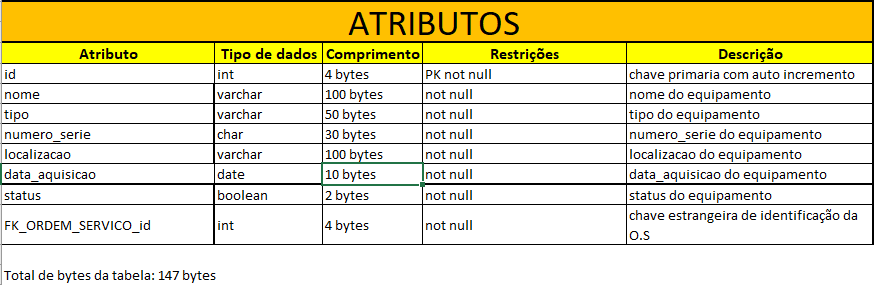
Nessa etapa é elaborada uma organização básica dos dados do banco. Aqui são informadas as entidades, com seus respectivos campos, tipos e descrições. O banco foi desenvolvido no servidor de banco de dados SQL Server 2012.

**ENTIDADES E RELACIONAMENTOS**

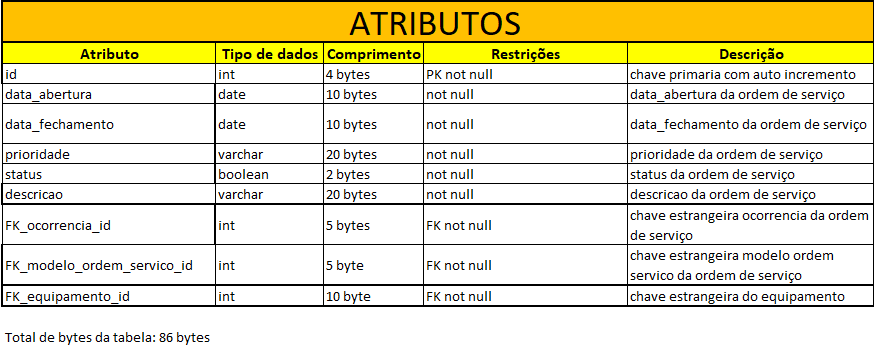


**ENTIDADES;**

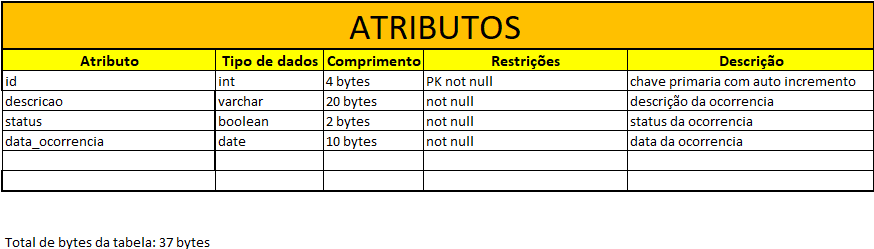
Equipamentos



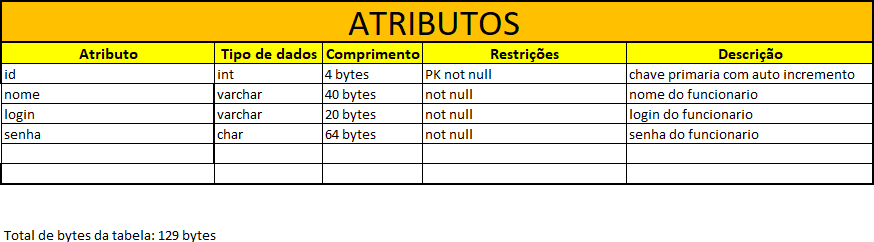
**Ordem de Serviço**



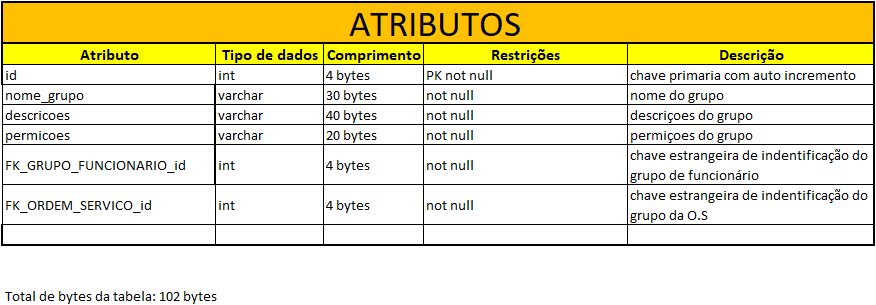
**OCORRÊNCIA**



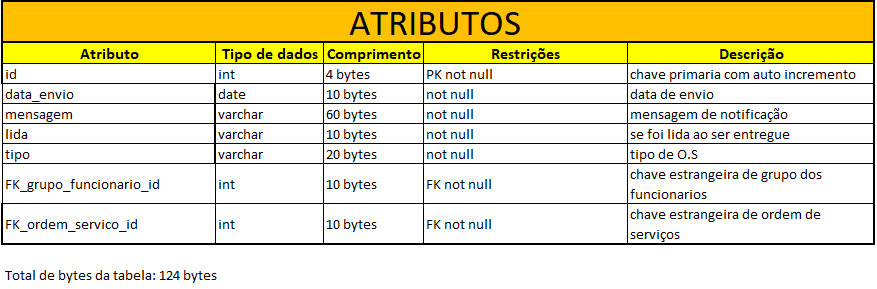
**FUNCIONÁRIO**



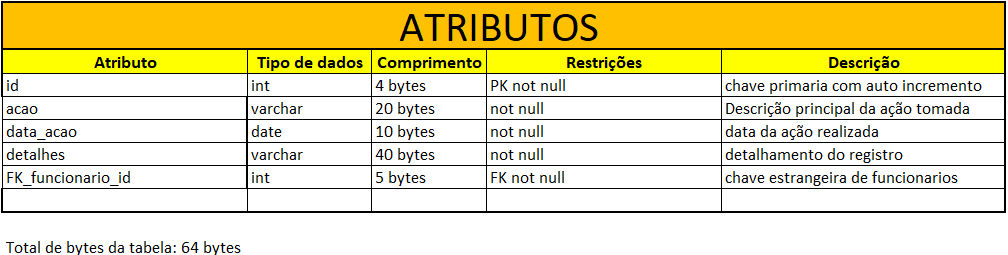
**GRUPO FUNCIONÁRIOS**



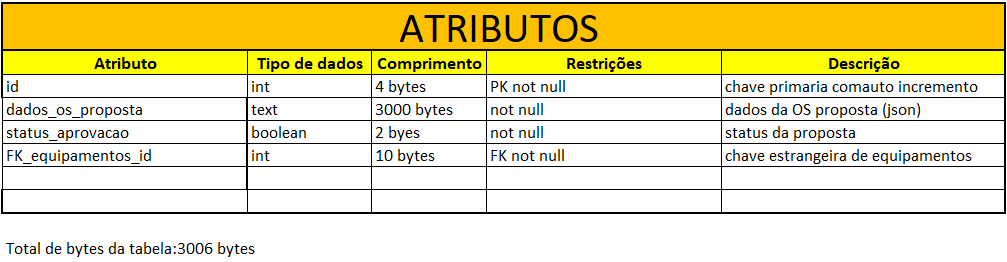
**NOTIFICAÇÃO**



**REGISTRO AUDITORIA**



**PREDITIVA IA**



PRINCIPAIS TELAS DO SISTEMA

Descreve de maneira simples as principais telas do sistema

# CONCLUSÃO

# Escreva os resultados obtidos

Resultados obtidos

# Constatações

Constatações

# Sugestões de possíveis aperfeiçoamentos técnicos

Sugestões

# REFERÊNCIAS

Aqui vai o texto de referências (MORE)

# GLOSSÁRIO

Se houver necessidade

# ANEXOS

Se houver necessidade