

Documentação Completa do Sistema - Sistema de Gerenciamento de Serviços e Recursos para Concessionária

Desenvolvedores: Jenyffer Laura Prochno Pereira
Rafael Tagliaferro Galafassi

Objetivo: Desenvolver um sistema abrangente para a gestão eficiente de recursos e serviços em uma rede de concessionárias, contemplando as áreas de funcionários, clientes e serviços.

1. Introdução

O Sistema de Gerenciamento de Serviços e Recursos para Concessionária foi desenvolvido com o propósito de otimizar a gestão de recursos e serviços em concessionárias, abrangendo áreas cruciais como atendimento ao cliente, manutenção de veículos e gestão financeira. Este documento oferece uma visão detalhada do sistema, abordando desde sua arquitetura e modelo de dados até suas funcionalidades principais e o cronograma do projeto. Com uma metodologia ágil e a utilização de padrões de design modernos, o sistema fornecerá uma solução robusta e eficiente para as necessidades das concessionárias, visando melhorar a experiência tanto dos clientes quanto dos funcionários.

2. Metodologia

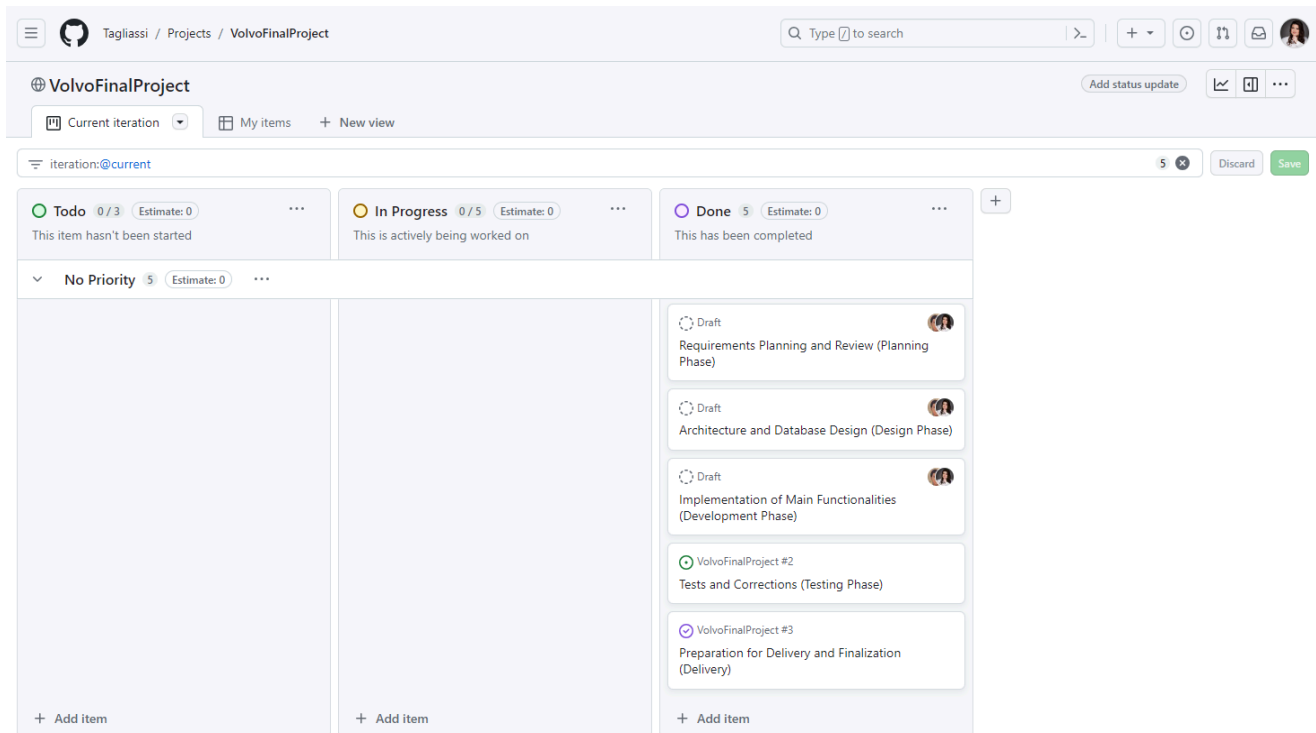
A metodologia Agile foi adotada para garantir flexibilidade e entrega rápida de valor, com ênfase nas funcionalidades fundamentais para uma implantação rápida e aprimoramentos contínuos. Utilizamos para o desenvolvimento dessa solução alguns artefatos do Scrum para facilitar e agilizar o processo como o MVP (Minimum Viable Product), o Kanban e o desenvolvimento de Personas.

2.1 Minimum Viable Product (MVP)

Para assegurar a entrega de um sistema completo, com todas as funcionalidades necessárias, optamos por desenvolver um Produto Mínimo Viável.

2.2 Kanban

Utilizamos para a organização das ações no Kanban do Github:



2.3 Desenvolvimento de Personagens

Problema a ser Resolvido: Agilizar atendimento e manutenção de caminhões.

Como o Sistema Soluciona: Proporciona gestão eficiente, permitindo aos clientes solicitarem serviços e funcionários realizarem serviços.

- **Recepcionista:**

Atividades: Recepcionar clientes, abrir chamados para o Gerenciador de Inventário.

Necessidades: Verificar a disponibilidade do Gerenciador de Inventário e do Mecânico.

- **Gerenciador de Inventário:**

Atividades: Validar disponibilidade de peças.

Necessidades: Verificar disponibilidade do Mecânico e das peças para atendimento.

- **Mecânico:**

Atividades: Analisar pedido do serviço, realizar o serviço.

Necessidades: Verificar situação do caminhão para realizar o serviço.

- **Cliente:**

Atividades: Solicitar serviço, receber o serviço.

Necessidades: Receber atendimento dos funcionários.

3. Arquitetura

O sistema segue a arquitetura MVC (Model-View-Controller) para separar lógica de negócios, apresentação e manipulação de dados. O uso do padrão Repository and Unit of Work organiza e gerencia eficientemente o acesso a dados utilizando também os conceitos de Data Transfer Objects.

3.1 Componentes Principais

- **Model:** Representa entidades do sistema.
- **View:** Responsável pela interface do usuário.
- **Controller:** Gerencia interações do usuário, acionando lógica de negócios e atualizando a View.
- **Repository:** Manipula persistência de dados no banco de dados.
- **Unit of Work:** Coordena transações entre diferentes repositórios.
- **Middleware:** Software montado em um pipeline para lidar com solicitações e respostas.
- **DTO:** Garante a segurança da aplicação fazendo uma projeção das classes originais.
- **AutoMapper:** Mapeia as entidades para o DTO.

4. Modelo de Dados

O banco de dados é estruturado com base em entidades relacionadas. Abaixo estão as principais entidades:

1. Concessionária
2. Peças
3. Categoria Serviço
4. Serviço
5. Contatos
6. Funcionário
7. Veículo
8. Conta
9. Cliente

4.1 Modelo de Dados Detalhada

1. Tabela Contatos:

- IdContato: int;
- Número Endereço: int;
- Email: varchar;
- Número Telefone: varchar;
- Rua: varchar;
- Cidade: varchar;

- Bairro: varchar;
- Estado: varchar;
- CEP: varchar;
- Tipo Telefone: Enum

- 1- Celular,
- 2 - Residencial,
- 3 - Comercial.

2. Tabela Concessionária:

- IdConcessionaria: int;
- fkContato: int (referenciando tabela Contatos);
- CNPJ: varchar;
- Nome: varchar.

3. Tabela Funcionario:

- IdFuncionario: int;
- fkConcessionaria: int (referenciando tabela Concessionária);
- Contato: int (referenciando tabela Contatos);
- Salário: double;
- Salário Base: double;
- Comissão: double;
- CPF: varchar;
- Nome: varchar;
- Cargo: Enum

- 1 - Recepcionista,
- 2 - Gerenciador de Inventário,
- 3 - Engenheiro Mecânico.

4. Tabela Servico:

- IdServico: int;
- fkConcessionária int (referenciando tabela Concessionária);
- IdFuncionario: int (referenciando tabela Funcionario);
- DataServico: date;
- Valor: double;
- Situação Servico: Enum

- 1 - Concluído,
- 2 - Em Andamento,
- 3 - Cancelado.

5. Tabela Categoria Serviço:

- IdCategoria: int;
- fkServico: int (referenciando tabela Serviço);

- Tempo Execução: int;
- Categoria: Enum.
 - 1 - Troca de Óleo,
 - 2 - Alinhamento,
 - 3 - Balanceamento,
 - 4 - Geometria,
 - 5 - Revisões Programadas,
 - 6 - Substituição de Peça,
 - 7 - Atualizações de software,
 - 8 - Atualizações de I-shift,
 - 9 - Serviço Extra,
 - 10 - Recepcionar o Cliente,
 - 11 - Validar a disponibilidade da peça.

6. Tabela Peças:

- IdPeca: int;
- fkCategoriaServiço: int (referenciado tabela Categoria Serviços);
- Quantidade: int;
- Valor: double;
- Disponibilidade: varchar;
- Nome: varchar;
- Locação: varchar.

7. Tabela Cliente:

- IdCliente: int;
- fkContato: int (referenciando tabela Contatos);
- fkVeículo: int (referenciando tabela Veículo);
- Nome: varchar;
- CPF: varchar.

8. Tabela Veículo:

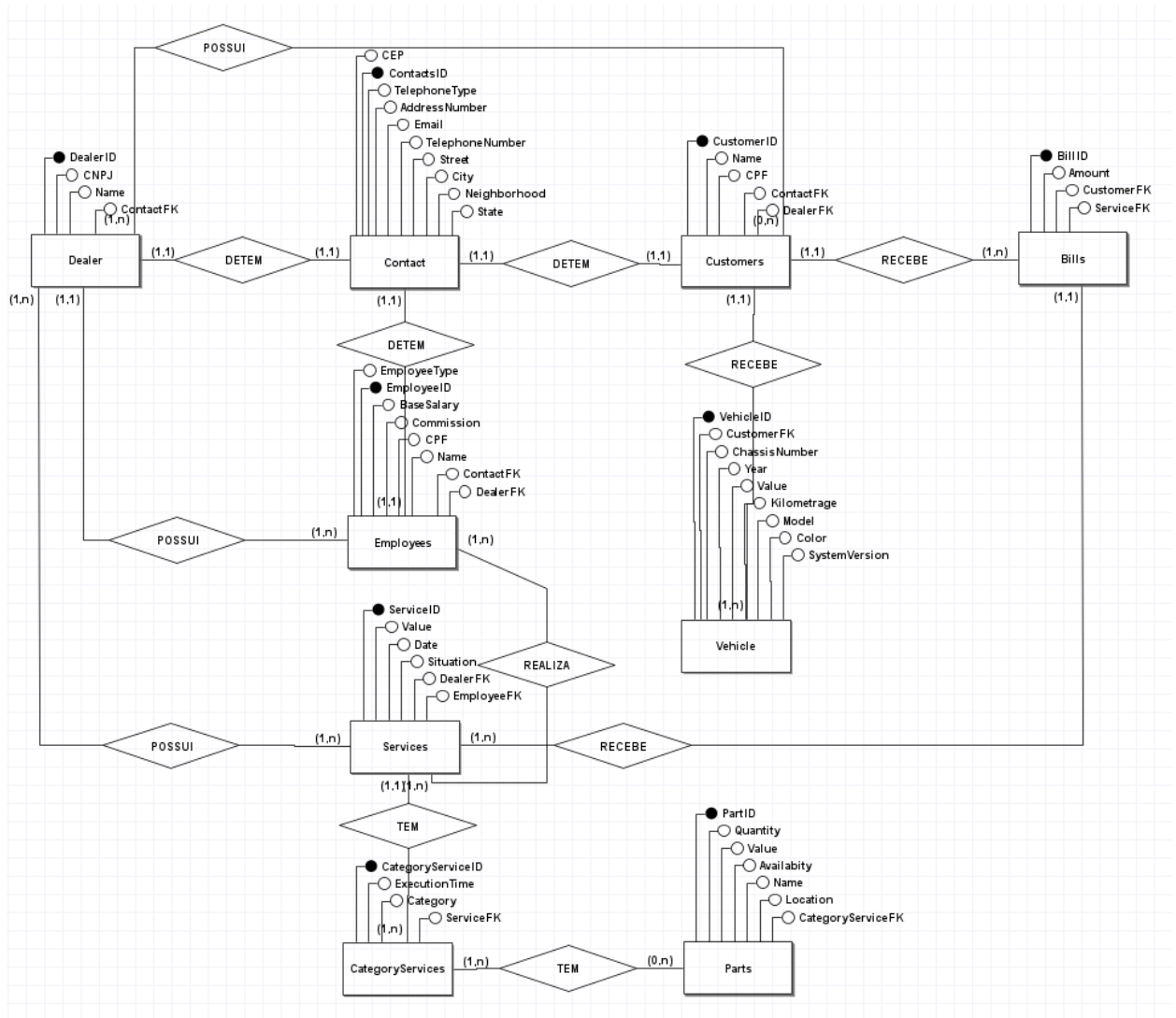
- IdVeiculo: int;
- fkCliente: int (referenciando tabela Cliente);
- Número Chassi: varchar;
- Ano: int;
- Valor: double;
- Quilometragem: double;
- Modelo: varchar;
- Cor: varchar;
- Versão Sistema: varchar.

9. Tabela Conta:

- IdConta: int;

- fkCliente: int (referenciando tabela Cliente);
- fkServico: int (referenciando tabela Servico);
- Valor Total: double.

4.2. Modelo Conceitual:



4. Funcionalidades Gerais

1. Cadastro e Gestão de Concessionárias:

- Adicionar, editar, excluir e obter dados sobre concessionárias.
- Obter os funcionários de uma concessionária específica.

2. Cadastro e Gestão de Peças:

- Adicionar, editar, excluir e obter dados de peças.

3. Cadastro e Gestão de Categorias de Serviço:

- Adicionar, editar, excluir e obter categorias de serviço.
- Associar peças à categorias de serviço.

4. Cadastro e Gestão de Serviços:

- Adicionar, editar, excluir e obter dados dos serviços.
- Associar peças, funcionários, clientes, veículos e categorias de serviços a um serviço.

5. Cadastro e Gestão de Contatos:

- Adicionar, editar, excluir e obter dados de um contato.
- Associar funcionários, clientes, e concessionárias a um contato.

6. Cadastro e Gestão de Funcionários:

- Adicionar, editar, excluir e obter dados sobre funcionários.
- Associar funcionários a concessionárias, clientes e serviços.
- Calcular salários e comissões de acordo com a os serviços prestados.

7. Cadastro e Gestão de Veículos:

- Adicionar, editar, excluir e obter dados sobre veículos.
- Associar veículos a clientes, serviços e concessionárias.
- Obter dados sobre veículos de acordo com quilometragem e versão do sistema.

8. Cadastro e Gestão de Contas:

- Adicionar, editar, excluir e obter dados financeiros relacionados a serviços prestados na conta do cliente.
- Associar um cliente e serviços a uma conta.
- Calcular o valor total da fatura.

9. Cadastro e Gestão de Proprietários (Clientes):

- Adicionar, editar, excluir e obter dados sobre clientes.
- Associar clientes a concessionárias.

10. Listar Peças Conforme os Serviços Prestados:

- Criar uma funcionalidade que permite retornar as peças associadas a uma categoria de serviço.
- Ao acessar os dados de um serviço prestado por um funcionário, exibir os dados desse serviço.

11. Calcular o Salário de Cada Funcionário:

- Implementar um módulo de cálculo de salários que considere o salário-base de cada funcionário.
- Associar as comissões mensais de cada funcionário aos serviços prestados.
- Calcular o salário total somando o salário-base e as comissões.

12. Logs:

- Gerar logs de erros persistidos em um ou mais arquivos.

4.1 Funcionalidades Principais: As funcionalidades essenciais do sistema incluem:

- Cadastro e Gestão de Concessionárias
- Cadastro e Gestão de Peças
- Cadastro e Gestão de Categorias de Serviço
- Cadastro e Gestão de Serviços
- Cadastro e Gestão de Funcionários
- Cadastro e Gestão de Veículos
- Cadastro e Gestão de Clientes
- Cadastro e Gestão de Contas
- Listar Peças Conforme as Categorias de serviços
- Calcular o valor total da fatura do cliente.
- Receber os funcionários de uma concessionária específica.
- Ao acessar as informações de um serviço prestado por um funcionário, exibir as informações do serviço.
- Calcular o Salário de Cada Funcionário.
- Registrar Logs de erros.

5. Design Patterns

O sistema incorpora padrões de design, incluindo Repository and Unit of Work, MVC Architecture e DTO pattern.

5.1. Repository Pattern:

O Repository Pattern permite um encapsulamento da lógica de acesso a dados, impulsionando o uso da injeção de dependência (DI) e proporcionando uma visão mais orientada a objetos das interações com a Data Access Layer (DAL).

5.2. Middleware Pattern:

Middleware em uma API refere-se a um componente intermediário que processa solicitações HTTP antes que elas alcancem o aplicativo principal e/ou após o aplicativo ter gerado uma resposta. No nosso sistema, o middleware desempenha um papel importante na captura e no tratamento de exceções ou erros que ocorrem durante o processamento das solicitações HTTP. Quando uma exceção não tratada ocorre na aplicação, o middleware pode intervir para capturá-la antes que ela cause uma falha no aplicativo e, em seguida, fornecer uma resposta apropriada ao cliente.

5.3. Data Transfer Object Pattern (DTO):

O padrão Data Transfer Object (DTO) em uma API refere-se a criação de objetos simples para transferir dados entre componentes de um sistema, como entre a

camada de aplicação e a camada de interface do usuário ou entre diferentes partes de uma API. O principal objetivo do padrão DTO é encapsular os dados em uma estrutura que pode ser facilmente transmitida entre diferentes partes do sistema, sem expor diretamente os detalhes internos dos objetos de domínio ou das entidades do banco de dados. Isso ajuda a promover uma separação clara de preocupações e a reduzir o acoplamento entre os diferentes componentes do sistema.

5.4. Unit of Work Pattern:

O padrão Unit of Work (Unidade de Trabalho) é um padrão de design que é comumente utilizado em aplicações que envolvem acesso a dados, como em APIs, para coordenar e gerenciar transações de banco de dados e operações CRUD (Create, Read, Update, Delete). O principal objetivo do padrão Unit of Work é fornecer uma maneira de garantir que todas as operações de banco de dados relacionadas a uma determinada transação sejam tratadas de forma consistente, garantindo que todas as operações sejam bem-sucedidas ou que sejam totalmente revertidas em caso de falha.

6. Ferramentas e Tecnologias

O sistema utiliza ferramentas como Entity Framework, Microsoft SQL Server, DOTNET C# e Microsoft Power Bi para garantir uma implementação robusta.

7. Qualidade do Código

O código segue padrões de qualidade, incluindo nomenclatura em inglês, camelCase e documentação adequada.

8. Logs

O sistema registra logs de erros persistentes em arquivos, fornecendo rastreamento detalhado de atividades.

9. Exemplo de Fluxo de Trabalho

Exemplo simplificado de um fluxo de trabalho no Sistema de Gerenciamento para Concessionária, envolvendo diferentes personas e funcionalidades principais do sistema.

9.1 Fluxo de Atendimento e Manutenção:

1. Cliente Solicita Atendimento:

O cliente entra em contato com a concessionária solicitando manutenção para seu caminhão.

2. Recepcionista Registra Serviço:

O recepcionista, utilizando a funcionalidade de Cadastro e Gestão de Serviços, registra um novo serviço associado ao cliente, veículo e categoria de serviço adequada e altera o status do serviço para em andamento.

3. O Serviço é passado ao Gerenciador de Inventário:

O Gerenciador de Inventário é avisado acerca do serviço para validar e preparar as peças necessárias, verificar se tem um funcionário mecânico disponível, e avisar o mecânico acerca do Serviço cadastrado.

4. Envio do Serviço ao Mecânico:

O Mecânico é avisado acerca do Serviço e recebe as peças para iniciar a manutenção.

5. Início da Manutenção:

O Mecânico inicia a manutenção e utilizando a funcionalidade de Cadastro e Gestão de Serviços e Peças ele verifica o que deve ser realizado.

6. Registro de Peças Utilizadas:

Durante a manutenção, o mecânico registra as peças utilizadas associadas ao serviço.

9. Conclusão da Manutenção:

Após a conclusão, o Mecânico registra a finalização do serviço utilizando a funcionalidade de Cadastro e Gestão de Serviços atualizando o status do serviço como concluído.

10. Cálculo de Comissão:

O sistema calcula automaticamente a comissão do Mecânico com base no serviço, utilizando a funcionalidade de Cadastro e Gestão de Funcionários.

11. Registro de Informações Financeiras:

Os dados financeiros relacionados ao serviço são registrados automaticamente utilizando a funcionalidade de Cadastro e Gestão de Contas.

12. Emissão de Relatório:

A equipe administrativa pode gerar relatórios sobre os serviços prestados, comissões e contas utilizando a funcionalidade de Relatórios.

10. Cronograma do Projeto: Sistema de Gerenciamento para Concessionária

Período: 01 de janeiro de 2024 a 07 de janeiro de 2024

1. Fase de Planejamento (01~02/02/2024):

- Revisão e validação dos requisitos do sistema.
- Detalhamento do escopo do projeto.
- Definição e alocação de recursos.

2. Fase de Design (03~04/02/2024):

- Elaboração e revisão da arquitetura do sistema.
- Projeto detalhado do banco de dados.

3. Fase de Desenvolvimento (05~06/02/2024):

- Implementação das funcionalidades principais do sistema.
- Desenvolvimento dos módulos de Cadastro e Gestão.
- Integração de peças, funcionários e serviços.
- Implementação dos padrões de design.

4. Fase de Testes (07/02/2024):

- Testes no sistema.
- Correção de bugs e otimizações.

5. Entrega (08/02/2024):

- Implementação final de logs.
- Preparação para a entrega oficial.
- Documentação final.
- Entrega do sistema.

Observações:

- Durante todas as fases, será mantida uma comunicação contínua com a equipe, revisando o progresso e ajustando o plano conforme necessário.
- Utilizaremos funcionalidades como Live Share do VSCode para realizarmos o desenvolvimento em par e nos manteremos em call durante todo o desenvolvimento.
- As datas específicas para cada fase podem ser ajustadas com base no progresso real e nas eventuais necessidades que surgirem durante o desenvolvimento.
- Haverá revisões regulares com a equipe para garantir a qualidade e aderência aos requisitos estabelecidos.

11. Conclusão

O Sistema de Gerenciamento de Serviços e Recursos para Concessionária é uma solução abrangente e estruturada que visa otimizar as operações em concessionárias, abrangendo desde o primeiro contato com o cliente até a gestão financeira dos serviços prestados. Nesse sentido, trouxemos uma solução que irá agilizar e aperfeiçoar o atendimento organizando todo o processo de atendimento.

A adoção das metodologias ágeis juntamente com a utilização de padrões de design e tecnologias modernas nos proporcionou flexibilidade, eficiência e agilidade no desenvolvimento desse projeto.

Além disso, a utilização de padrões como Repository and Unit of Work, Middleware e DTO garantiram uma abordagem consistente e segura na manipulação dos dados e interações com o usuário.

Diante do exposto acima, concluímos que esse projeto nos proporcionou acesso a fundamentações teóricas e colocando em prática tais ensinamentos tivemos a construção do conhecimento acerca da Metodologia Agile e Scrum, Arquitetura MVC e Padrões de Design, Ferramentas e Tecnologias (Entity Framework, Microsoft SQL Server, Microsoft Power BI e DOTNET C#), Planejamento e Gerenciamento de Projeto e Documentação.