Documento de Requisitos do Sistema

Service

Versão1.0

|  |  |
| --- | --- |
| Integrantes | R.A. |
| Caio Henrique Segawa Tonetti | 79064 |
| João Almeida de Jesus Junior | 61440 |
| Thiago Rodrigo Bucalão | 63284 |

Sumário

[1. Introdução 3](#_Toc428813283)

[1.1 Objetivo do documento 3](#_Toc428813284)

[1.2 Escopo do Produto 4](#_Toc428813285)

[Convenções, termos, siglas e abreviações 4](#_Toc428813286)

[Descrição Documento de Requisitos 4](#_Toc428813287)

[2.1. Interfaces com o Produto 4](#_Toc428813288)

[2.2.1. Requisitos Funcionais 5](#_Toc428813289)

[2.2.2. Requisitos Não-Funcionais 15](#_Toc428813290)

[2.3.1. Identificação dos Casos de Uso: 15](#_Toc428813291)

[2.3.2. Priorização dos Casos de Uso: 16](#_Toc428813292)

[2.4. Diagrama de Casos de Uso Completo e em partes 16](#_Toc428813293)

[2.5. Detalhamento dos Casos de Uso 19](#_Toc428813294)

[2.6. Modelo de Análise: 23](#_Toc428813295)

[2.6.1. Gerar Ordem de Serviço 23](#_Toc428813296)

[2.6.2. Realizar Venda 24](#_Toc428813297)

[2.6.4. Faturar Serviço 25](#_Toc428813298)

[2.7. Planejamento de Interações: 25](#_Toc428813299)

[2.7.1. Interação 1 (cadastros básicos) 25](#_Toc428813300)

[2.7.2. Interação 2..N: 25](#_Toc428813301)

[2.8. Diagrama de Classes do Projeto (Completo e Dividido): 26](#_Toc428813302)

[3. Glossário: 29](#_Toc428813303)

[4. Análise de Viabilidade 29](#_Toc428813304)

[4.1. Contribuição para a empresa 29](#_Toc428813305)

[4.2. Viabilidade Técnica 30](#_Toc428813306)

[4.3. Viabilidade Legal 30](#_Toc428813307)

[5. Análise de Risco 30](#_Toc428813308)

[6. Diagrama de Objetos de Negócios: 31](#_Toc428813309)

[7. Diagrama de Pacotes: 32](#_Toc428813310)

[8. Arquiteturas de Software 33](#_Toc428813311)

[8.1. Arquiteturas não utilizadas 33](#_Toc428813312)

[8.1.1. Pipes e Filtros 33](#_Toc428813313)

[8.1.2. Quadro Negro 33](#_Toc428813314)

[8.1.3. Invocações Implícita 33](#_Toc428813315)

[8.1.4. Orientada a eventos 33](#_Toc428813316)

[8.1.5. Aplicações Distribuídas 33](#_Toc428813317)

[8.1.6. Ponto a Ponto 34](#_Toc428813318)

[8.1.7. Sistemas Orientados a Serviços 34](#_Toc428813319)

[8.1.8. Estilos Arquiteturais – Variações 34](#_Toc428813320)

[8.1.9. Cliente-Servidor de duas camadas 34](#_Toc428813321)

[8.1.10. Cliente-Servidor de múltiplas camadas para Web 34](#_Toc428813322)

[8.1.11. Sistema de tempo Real 35](#_Toc428813323)

[8.1.12. Em Nuvem 35](#_Toc428813324)

[8.2. Arquiteturas Utilizadas 35](#_Toc428813325)

[8.2.1. Objetos 35](#_Toc428813326)

[8.2.2. Sistema de Informação (Cliente Magro) 35](#_Toc428813327)

[8.2.3. MVC (Modelo Visão Controle) 35](#_Toc428813328)

[9. Frameworks: 36](#_Toc428813329)

[9.1. Vantagens e Desvantagens: 36](#_Toc428813330)

[9.2. Funcionamento do Framework: 36](#_Toc428813331)

[10. Diagrama de Classe Arquitetural: 38](#_Toc428813332)

[11. Padrão de Projeto: 38](#_Toc428813333)

[12. Diagrama de Componentes: 40](#_Toc428813334)

# Introdução

O documento expõe as características que serão implantadas no projeto *Service*, especificando o comportamento do sistema a ser desenvolvido para o auxílio dos desenvolvedores, podendo também ser usado como um contrato com o cliente.

*Service* é um *software* que visa à automatização dos setores que englobam cliente, fornecedor, pedido, estoque e financeiro da organização Cybercom.

# 1.1 Objetivo do documento

Ele tem como objetivo diminuir os erros provenientes pelo repasse de informações manuais, pois será armazenado em um banco de dados e as entradas serão processadas computacionalmente, eliminando repetições, com isso evitando também a duplicidade de informações. Contará ainda com uma interface simples e de fácil acesso as funções, permitindo integração entre os diferentes ramos organizacionais.

# 1.2 Escopo do Produto

Sistema criado com objetivo de organizar melhor o processo de manutenção de equipamentos. Sendo assim ele controlará a entrada e saída de máquinas, através de uma ordem de serviço a qual possuirá informações de máquinas, técnico e clientes. O processo se inicia com a entrada do equipamento e finaliza na entrega do mesmo. Existirão dois níveis de usuários que manusearão o produto sendo eles o técnico (mais avançado), e o atendente. O produto deverá controlar clientes e seus devidos equipamentos o qual poderá ser consultado sempre que necessário.

# Convenções, termos, siglas e abreviações

RF: Requisito funcional.

NF: Requisito não funcional.

HD: *HardDisk.*

RAM: Descrições de siglas e abreviações necessárias para interpretar apropriadamente o texto.

# Descrição Documento de Requisitos

# 2.1. Interfaces com o Produto

|  |  |
| --- | --- |
| **Ator** | **Descrição** |
| Cliente | Seria uma pessoa que tem algum problema em seu computador ou periférico e viria até a empresa para consertar seu equipamento. |
| Técnico | Responsável pela manutenção do equipamento, avisar o cliente os resultados tanto da avaliação técnica quanto conclusão do serviço. |
| Atendente | Funcionário responsável pela recepção do cliente e equipamento, cadastro de ambos caso necessário, e a geração da ordem de serviço para o equipamento. Será responsável também pela a entrega do equipamento e cobrar a fatura, além disso pela venda de peças que o cliente solicita como (teclado, *mouse,* modem e etc.). |
| Administrador | Funcionário da empresa responsável por fazer cadastros de funcionários e equipamentos. |

# 2.2.1. Requisitos Funcionais

[RF001] - Cadastrar Cliente: <Thiago>

Descrição: O sistema deve permitir o cadastro de novos clientes, no qual irá informar seus dados e esses serão salvos no banco de dados da empresa. Os campos para realizar o cadastro são: Nome completo (Obrigatório), Endereço (Obrigatório), RG (Obrigatório), CPF ou CNPJ (Obrigatório), Telefone para contato (Obrigatório), E-mail (Opcional).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF004* | | | | | |

[RF002] - Alterar Cliente: <Thiago>

O sistema deve permitir alterar os dados dos clientes já cadastrados. Os campos para realizar a alteração são: Nome completo (Obrigatório), Endereço (Obrigatório), RG (Obrigatório), CPF ou CNPJ (Obrigatório), Telefone para contato (Obrigatório), E-mail (Opcional).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF003* | | | | | |

[RF003] - Consultar Cliente: <Thiago>

O sistema deve permitir consultar os dados dos clientes já cadastrados. Os campos para realizar consulta são: Nome ou RG ou CPF (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF004* | | | | | |

[RF004] - Excluir Cliente: <Thiago>

O sistema deve permitir excluir clientes já cadastrados. Os campos para realizar exclusão são: Nome completo (Obrigatório), Endereço (Obrigatório), RG (Obrigatório), CPF ou CNPJ (Obrigatório), Telefone para contato (Obrigatório), E-mail (Opcional).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF001* | | | | | |

[RF005] - Cadastrar Máquina: <Caio>

O sistema deve permitir o cadastro de novas máquinas que clientes desejam fazer manutenção e essas serão salvas no banco de dados. Os campos desse cadastro serão: Código(Automático), Fabricante(Obrigatório), Modelo (Obrigatório), Dono(Obrigatório), Vídeo (Obrigatório), RAM (Obrigatório), Processador(Obrigatório), Placa Mãe (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF008* | | | | | |

[RF006] – Alterar Máquina: <Caio>

O sistema deve permitir a alteração de máquinas já cadastradas. Os campos para alteração são: Código(Automático), Fabricante(Obrigatório), Modelo (Obrigatório), Dono(Obrigatório), Vídeo (Obrigatório), RAM (Obrigatório), Processador(Obrigatório), Placa Mãe (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF008* | | | | | |

[RF007] - Excluir Máquina: <Caio>

O Atendente irá informar o código da máquina, verificar se o resultado é o que ele solicitou e então irá excluir a máquina do banco de dados. Os campos para exclusão serão: Código(Automático), Fabricante(Obrigatório), Modelo (Obrigatório), Dono(Obrigatório), Vídeo (Obrigatório), RAM (Obrigatório), Processador(Obrigatório), Placa Mãe (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF008* | | | | | |

[RF008] - Consultar Máquina: <Caio>

O sistema deve permitir a consulta de máquinas já cadastradas. Os campos para consulta são: Fabricante ou Modelo ou Dono (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF005* | | | | | |

[RF009] - Cadastrar Funcionário: <João>

O sistema deve permitir o cadastro de novos funcionários admitidos pela empresa, sendo esses classificados em atendentes e técnicos, e esses serão salvos no banco de dados. Os campos de cadastro serão: Código (Automático), Nome (Obrigatório), Login (Obrigatório), Senha (Obrigatório), Tipo (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF00* | | | | | |

[RF010] - Alterar Funcionários: <João>

O sistema deve permitir a alteração dos dados de funcionários já cadastrados. Os campos para alteração são: Código (Automático), Nome (Obrigatório), Login (Obrigatório), Senha (Obrigatório), Tipo (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF009* | | | | | |

[RF011] - Excluir Funcionário: <João>

O sistema deve permitir a exclusão de funcionários já cadastrados. Os campos para exclusão são: Código (Automático), Nome (Obrigatório), Login (Obrigatório), Senha (Obrigatório), Tipo (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF009* | | | | | |

[RF012] – Buscar Funcionário: <João>

Descrição: O sistema deve permitir a consulta de funcionário já cadastradas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF009* | | | | | |

[RF013] - Gerar Ordem de Serviço: <Thiago>

Caso o cliente e a máquina já estejam cadastrados no sistema será permitido gerar uma ordem de serviço com um prazo pré-estabelecido para que o produto seja avaliado pelo departamento técnico, portanto, será informado para o cliente o valor para se fazer o reparo na máquina. Caso o cliente autorize, será dado continuidade a manutenção do aparelho. Os dados serão salvos no banco de dados.

Os campos para gerar serão: Código (Automático), Código Máquina (Obrigatório), Problema (Obrigatório), Orçamento (Obrigatório), Data Entrada (Obrigatório), Previsão Saída (Obrigatório), Entregue (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF0005* | | | | | |

[RF014] - Consultar Ordem de Serviço: <Caio>

O sistema deve permitir a consulta de ordens de serviço já cadastradas. Os campos para consultar serão: Data Entrada (Obrigatório), Previsão Saída (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF013* | | | | | |

[RF015] - Encerrar Ordem de Serviço: <João>

Descrição: Após a avaliação técnica ou finalização do processo de manutenção, pode ocorrer a necessidade de encerrar a ordem de serviço e assim gerar a fatura para o cliente. Os campos para encerrar: Código (Automático), Código da Máquina (Obrigatório), Problema (Obrigatório), Data Entrada (Obrigatório), Previsão Saída (Obrigatório), Entregue (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF013* | | | | | |

[RF016] - Alterar Ordem de Serviço: <Thiago>

O sistema deve permitir a alteração de ordem de serviço já geradas. Os campos para alteração serão: Código (Automático), Código da Máquina (Obrigatório), Problema (Obrigatório), Data Entrada (Obrigatório), Previsão Saída (Obrigatório), Entregue (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF013* | | | | | |

[RF017] - Cadastrar produto em estoque: <Thiago>

Irá ser feito o cadastro dos produtos no qual estarão dispostos na loja para venda do cliente, tais como fonte, *HD,* memória RAM, Placas de Vídeo e etc. Os Campos para cadastro são: Código (Obrigatório), Nome do Produto (Obrigatório), Marca (Obrigatório), Modelo (Obrigatório), Valor (Obrigatório), Quantidade (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF00* | | | | | |

[RF018] – Alterar produto em Estoque: <João>

Descrição: Permite ao usuário alterar o Valor do produto e sua Quantidade em estoque, assumimos em nosso sistema que ao fazer o cadastro de um produto este existe na loja, ou seja, a quantidade mínima disponível em estoque será maior ou igual a 1. Tem se a ideia de utilizar a alteração para controlar a entrada e saída da quantidade do produto disponível no estoque, assim, quando for dar baixa em um produto iremos decrementar a quantidade disponível em estoque, caso seja adquirido novos produtos sua quantidade será incrementada.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF017* | | | | | |

[RF019] – Excluir Produto em Estoque: <Caio>

Descrição: Permite ao usuário que um determinado produto seja removida da lista de itens em estoque.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF017* | | | | | |

[RF020] – Consultar Produto em Estoque: <João>

Descrição: Permitir buscar por um determinado produto em estoque, informando assim o nome do produto e também a quantidade desse produto em estoque.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF017* | | | | | |

[RF021] – Emitir Relatório do Estoque: <Caio>

Descrição: Irá emitir um relatório com dados de entrada dos produtos e saída dos produtos do estoque (Os Campos existentes no relatório são: Código, Nome do Produto, Marca, Modelo, Valor, Quantidade.), o relatório pode ser exibido por um período de tempo, por exemplo, emitir relatório do mês de janeiro. Ou seja, tem por função manter um histórico do Estoque.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF020* | | | | | |

[RF022] - Realizar Venda: <João>

Caso o cliente queira comprar algum produto tal como uma fonte, HD, Teclado e etc. Para isso deverá buscar pelo produto e solicitar a venda. Os campos para realizar venda são: Código (Automático), Código do Cliente (Obrigatório), Nome do Atendente (Automático), Nome do Produto (Obrigatório), Código do Produto (Obrigatório), Valor do Produto (Obrigatório).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF001, RF009, RF017* | | | | | |

[RF023] - Cancelar Venda: <João>

Caso o cliente queira desistir da compra de um produto. Sendo assim o atendente cancelara a venda por meio da interface.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** |  | | | | | |
| **Requisitos associados:** |  | | | | | |

[RF024] - Devolver Venda: <João>

Caso o produto apresente uma avaria ou defeito de fabricação, o cliente poderá fazer a devolução do produto em questão. Mediante apresentação da embalagem do produto e a nota fiscal. O atendente irá fazer a devolução mediante interface e fazer uma descrição sobre a avaria.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | RF001, RF017 | | | | | |

[RF025] – Emitir boleto da fatura: <Caio>

Após gerar o valor da fatura, o atendente irá informar o código de Ordem de Serviço para imprimir o boleto de pagamento do serviço. O boleto deverá informar: Nome do Cliente, Nome do Atendente, Nome do Técnico, Código Serviço, Problema Serviço, Valor Total.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF022, RF001, RF017* | | | | | |

[RF026] – Controle de Devedores: <Caio>

Descrição: Caso o cliente atrase na data de pagamento de uma compra a prazo, seu nome irá para uma lista onde consta os devedores, quando o cliente quitar as dívidas ele será excluído da lista de devedores, a consulta da lista de devedores será usada para gerar relatórios.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF023* | | | | | |

[RF027] – Emitir Relatório de Devedores: <Caio>

Descrição: Será possível emitir um relatório com os nomes e dados dos clientes que estão em atraso com o pagamento.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Atendente | | | | | |
| **Requisitos associados:** | *RF00* | | | | | |

[RF028]- Controle de Finanças: <João>

Descrição: Será possível calcular o total de arrecadações com base nas vendas à vista, vendas à debito e faturas pagas em um determinado intervalo de tempo (data início, data fim).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Administrador |  |  |  |  |  |
| **Requisitos associados:** | *RF022, RF023* | |  |  |  |  |

[RF029] – Criar Orçamento: <Thiago>

Para que possa dar procedência ao serviço é necessário que um valor de orçamento seja criado, informado ao cliente e este permitindo o concerto da máquina.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Técnico |  |  |  |  |  |
| **Requisitos associados:** | RF013 | |  |  |  |  |

[RF030] – Gerenciar Orçamento: <Thiago>

Caso tenha verificado alguma nova falha na máquina do cliente que não foi cadastrado antes então será permitido fazer alteração no orçamento já criado.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Ator(es):** | Técnico |  |  |  |  |  |
| **Requisitos associados:** | RF029 | |  |  |  |  |

# 2.2.2. Requisitos Não-Funcionais

[NF001] Interface amigável:

Descrição: A interface deve ser simples e intuitiva, pois como o sistema será operado por pessoas que podem possuir pouco conhecimento sobre computação deve-se ter um acesso facilitado.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** | X | Essencial |  | Importante |  | Desejável |
| **Requisitos associados:** | *RF001, RF002, RF003, até o RF019.* | | | | | |

[NF002] Desempenho:

Descrição: Todas as funcionalidades referentes a relatórios e históricos devem gerar resposta em no máximo 60 segundos.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** |  | Essencial |  | Importante | X | Desejável |
| **Requisitos associados:** |  | | | | | |

[NF003] Manutenibilidade:

Descrição: O sistema deve ser fácil de ser alterado, pois pode haver interesse em aumentar as funcionalidades do mesmo.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prioridade:** |  | Essencial |  | Importante | X | Desejável |
| **Requisitos associados:** | *RF001, RF002, RF003, até o RF019.* | | | | | |

# 2.3.1. Identificação dos Casos de Uso:

1. Emitir fatura
2. Realizar venda
3. Gerenciar ordem de serviço
4. Gerenciar cliente
5. Gerenciar produtos
6. Gerenciar funcionarios
7. Emitir relatório de devedores
8. Emitir relatório de estoque (histórico de movimento de estoque)
9. Gerenciar devedores
10. Controle de Finanças
11. Gerar Orçamento

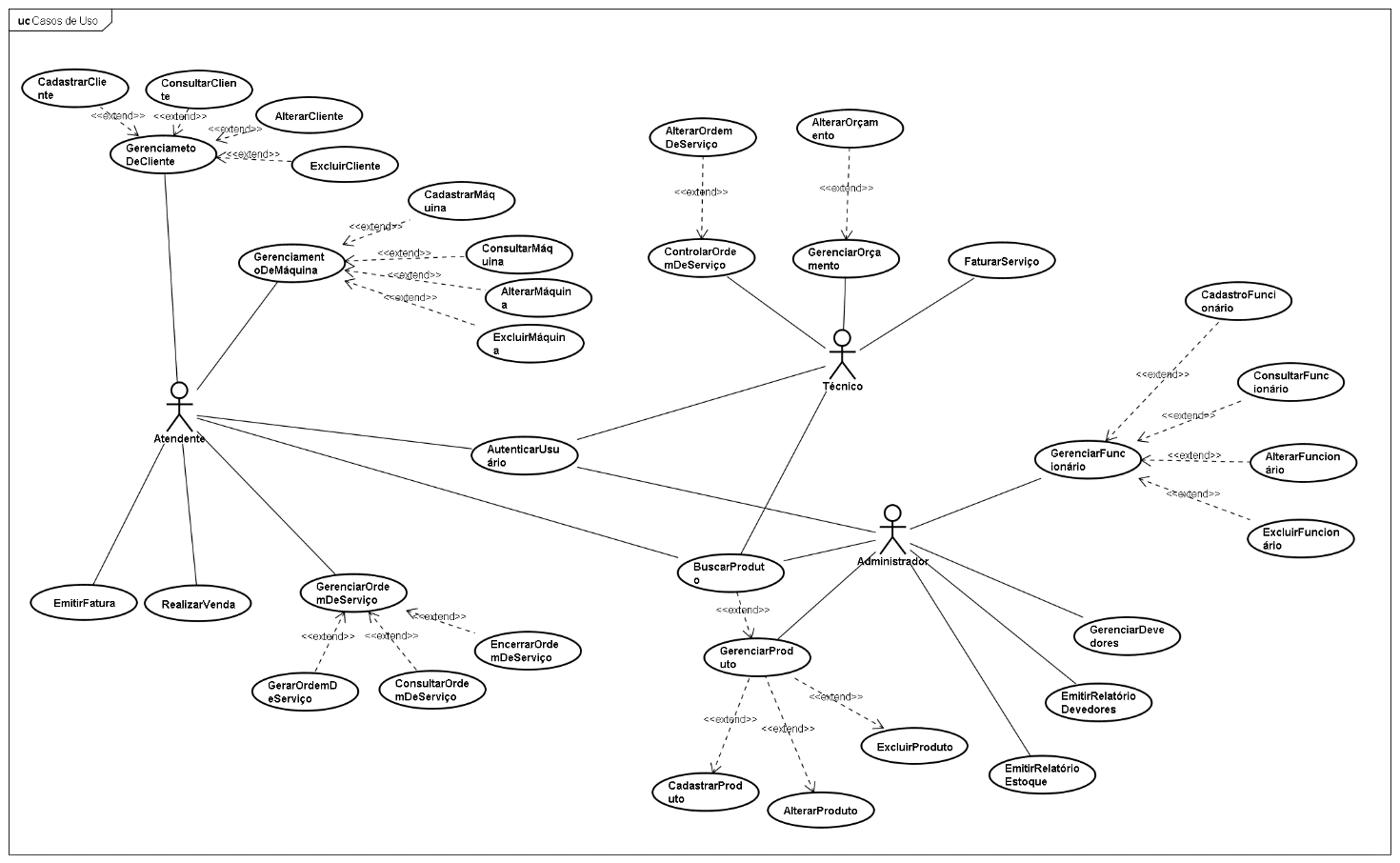
# 2.3.2. Priorização dos Casos de Uso:

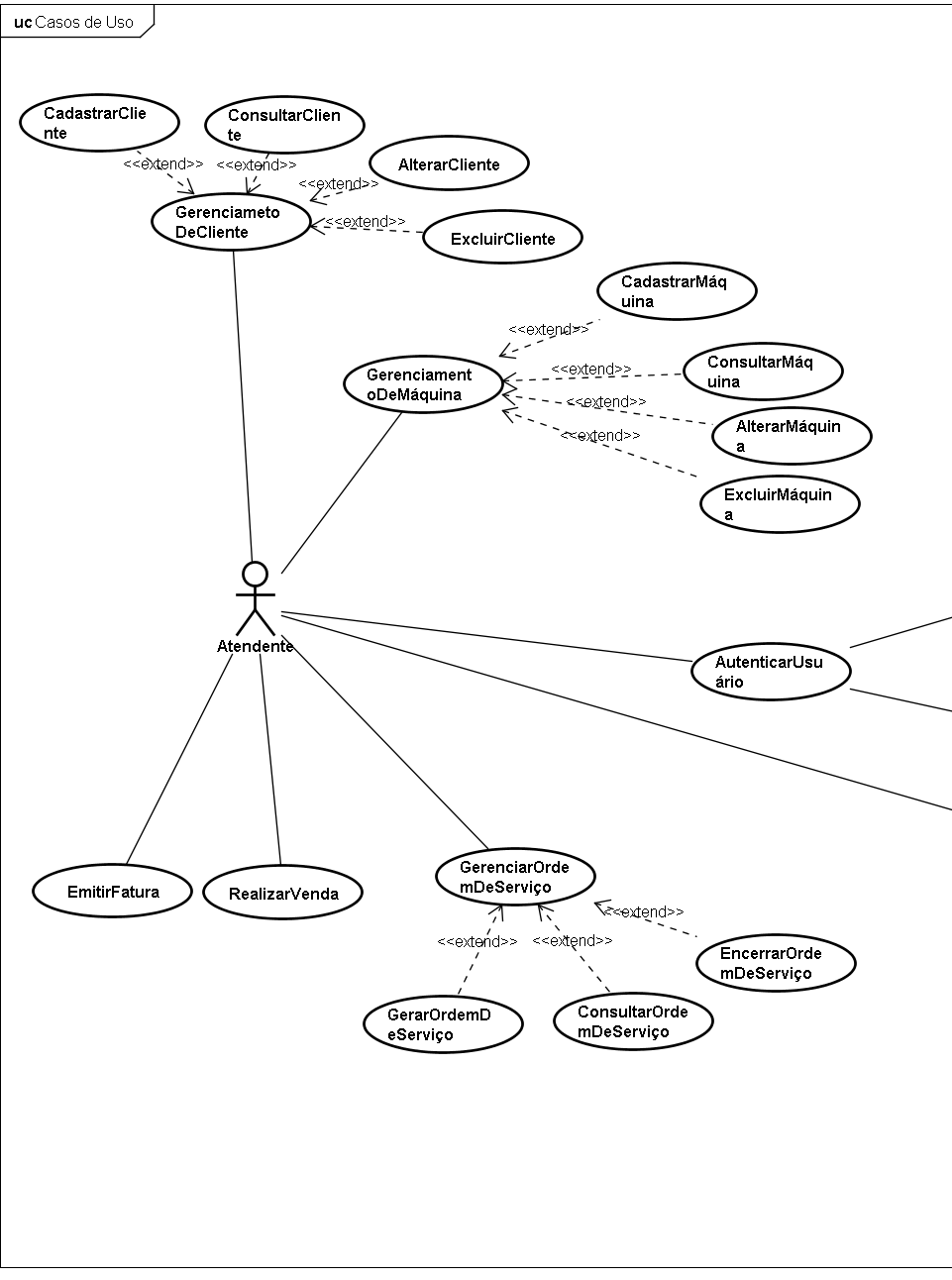
1. Gerenciar ordem serviço
2. Gerar Orçamento
3. Realizar venda
4. Emitir fatura
5. Emitir relatório de estoque(histórico de movimento de estoque)
6. Gerenciar funcionários
7. Gerenciar produtos
8. Gerenciar cliente
9. Emitir relatório de devedores
10. Gerenciar devedores
11. Controle de Finanças

# 2.4. Diagrama de Casos de Uso Completo e em partes

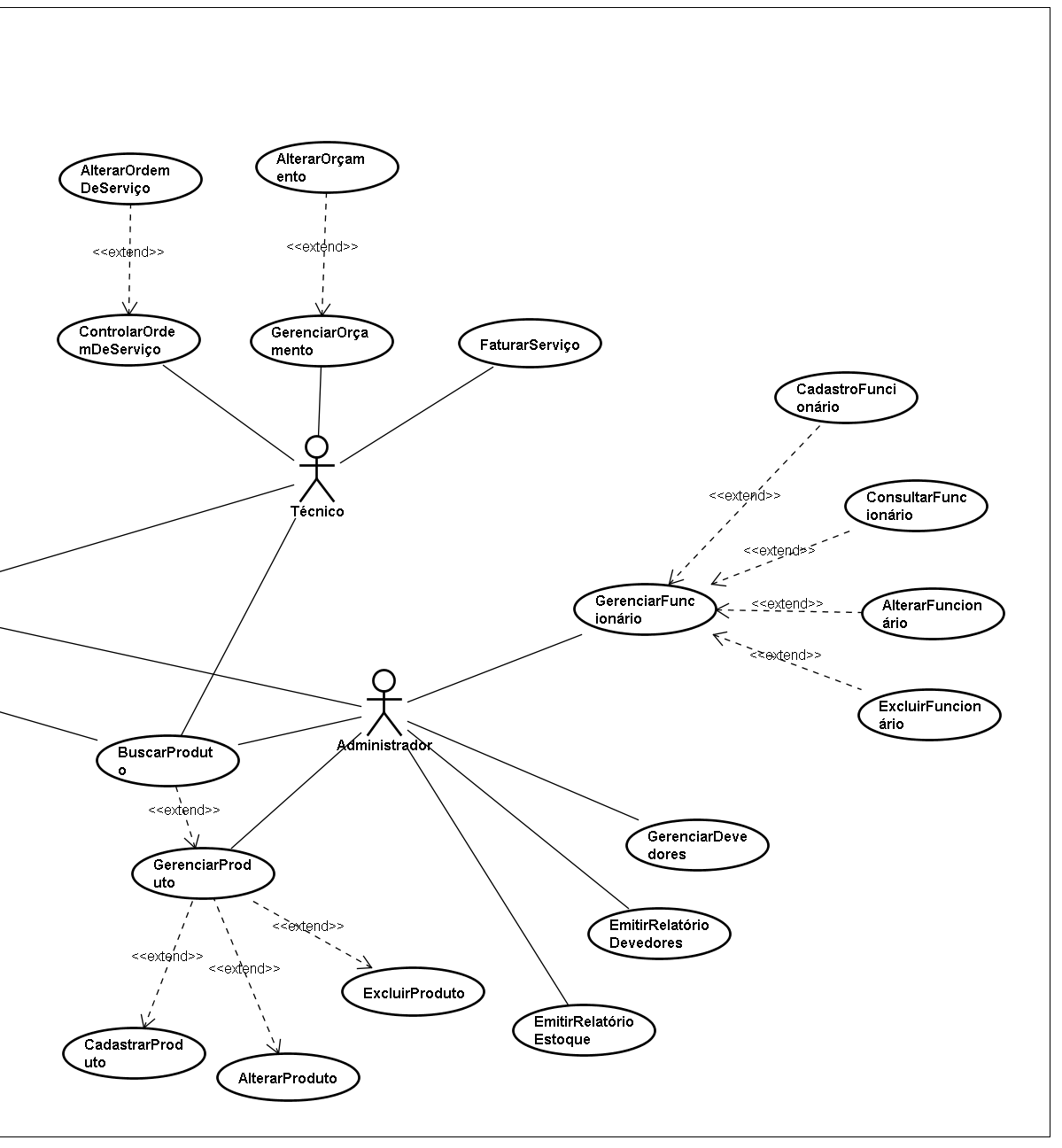
**Figura 3**

A figura 3 apresenta o modelo de caso de uso para o usuário Atendente. O objetivo desse digrama da UML é representar as principais funcionalidades do sistema, neste caso funcionalidades realizadas por todos os atores: Atendente, Técnico e Administrador.





**Figura 4.**

****

**Figura 5.**

# 2.5. Detalhamento dos Casos de Uso

|  |
| --- |
| **Caso de uso:** Gerar Ordem de Serviço |
| **Responsável:** Thiago Bucalão |
| **Identificação:** F1 |
| **Descrição:** O atendente irá realizar o *login* no sistemaem seguida realizaa busca do cliente no sistema através do CPF ou CNPJ do cliente. O sistema deverá listar todos as informações relacionadas a estes dados. O Cliente informará os dados da máquina, as quais serão verificados pelo sistema. Após isto será gerado uma ordem de serviço para o equipamento. |
| **Ator (es):** Atendente |
| **Pré-condição:** O sistema aguarda *login* do atendente e que cliente e máquina já estejam cadastrados. |
| **Pós-condição:** O sistema terá gerado uma Ordem de Serviço. |
| **Curso Normal:**   1. O atendente entra com as informações do cliente no sistema e verifica se está tudo correto. 2. Gera uma ordem de serviço com os dados da máquina e do cliente. 3. O Sistema informa na tela que a Ordem de Serviço foi gerada com sucesso. 4. O técnico entrega ao cliente um comprovante da ordem de serviço gerada. |
| **Fluxo Alternativo:** Cliente pendente.  No item 1, o sistema informa que o cliente informado tem pendências com a empresa, e por isso não há como prosseguir até que a dívida seja quitada. O sistema informa na página os dados do cliente e sua pendência.  **Fluxo Alternativo:** Ordem de serviço não é gerada.  No item 2, o sistema não irá gerar uma ordem de serviço caso haja alguma inconsistência nos dados ou campos obrigatórios não tenham sido preenchidos. Retorna a página de Gerar Ordem de Serviço. |

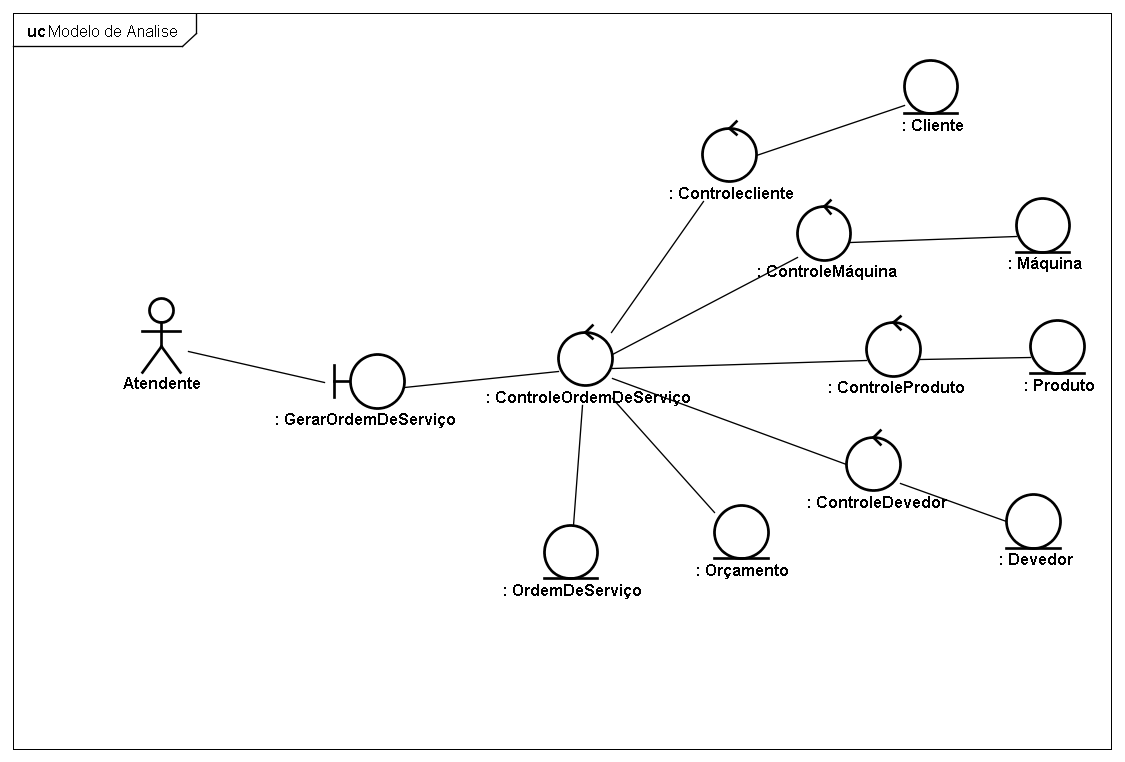
|  |
| --- |
| **Caso de Uso:** Realizar Venda. |
| **Responsável:** João |
| **Identificador:** F3 |
| **Descrição:** Realizar a venda de um produto, tal como mouse, teclado, etc a um cliente |
| **Ator (es):** Atendente |
| **Pré-condição:** O usuário (atendente) deverá estar logado no sistema, para, que pos ter o nível de acesso necessário para realizar venda. O produto deve constar no estoque. |
| **Pós-condição:** O usuario verifica a situação fisica do produto vendido. E emitido uma fatura e a quantidade dos itens é decrementada do estoque |
| **Curso Normal:**   1. O usuário executa o sistema service; 2. O usuário acessa a área do atendente (<<autenticar usuario>>) 3. Na tela inicial da sua area o usuário seleciona a opção realizar venda. 4. O usuário entra com o nome do produto e o sistema faz uma busca no estoque do mesmo; 5. Após a localização do produto no estoque e aprovação do cliente; 6. E feito um cadastro do cliente (se ele autorizar): com nome, RG, CPF, CNPJ, E-Mail, endereço, telefone. 7. O usuário escolhe a forma de pagamento. 8. O usuário seleciona a opção concluir venda; 9. O sistema emite uma fatura de venda do produto; 10. O sistema emite uma mensagem de “venda realizada com sucesso!”; 11. Finaliza a plicação. |
| **Fluxo alternativo:** No item 4 do fluxo principal, o produto não se encontra disponível do estoque:   1. O usuário informa ao cliente que o produto não consta como disponível em estoque; 2. O usuário seleciona cancelar venda; 3. O sistema informa um mensagem na tela, “se deseja confirmar a ação”; 4. O sistema retorna a página inicial do atendente.   No item 5 do fluxo principal, o cliente desiste compra do produto:   1. O usuário seleciona cancelar venda; 2. O sistema informa uma mensagem na tela, se deseja confirmar a ação; 3. O sistema informa uma mensagem na tela, “se deseja confirmar a ação”; 4. O sistema retorna a página inicial do atendente. |

|  |
| --- |
| **Caso de Uso:** Emitir relatório do estoque |
| **Responsável:** Caio |
| **Identificador:** F3 |
| **Descrição:** Verificar como o estoque se encontra no momento, exibindo a quantidade de produtos já cadastradas no estoque. Serão exibidos no relatório do estoque informações que variam de produtos unitários (memória RAM) até um notebook, também quantos produtos entraram e saíram em um determinado período. |
| **Ator(es):** Administados e Atendente |
| **Pré-condição:** A primeira condição para emitir um relatório é o usuário estar autenticado no sistema e seu nível de acesso permitir que ele tenha acesso a emissão de relatórios. A segunda condição é escolher o tipo de relatório desejado, nesse caso o usuário irá selecionar a opção EMITIR RELATÓRIO DO ESTOQUE. |
| **Pós-condição:** O usuário verificou qual a situação do estoque (entradas e saídas) em um determinado período. |
| **Curso Normal:**  1 – O usuário acessa a tela do sistema.  2 – O sistema solicita autenticação do usuário (<<include Autenticar Usuário>>).  3 – O usuário busca no menu a opção Relatório.  4 – O usuário escolhe a opção Relatório do Estoque.  5 – O sistema exibe as informações sobre a situação do Estoque.  6 – O usuário escolhe a opção Emitir Relatório do Estoque.  7 – O sistema imprime as informações.  8 – O cliente escolhe a opção encerrar Consulta. |
| **Fluxo Alternativo:**  1– O sistema informa que não possui informações inseridas.  2– O cliente escolhe a opção encerrar Consulta. |

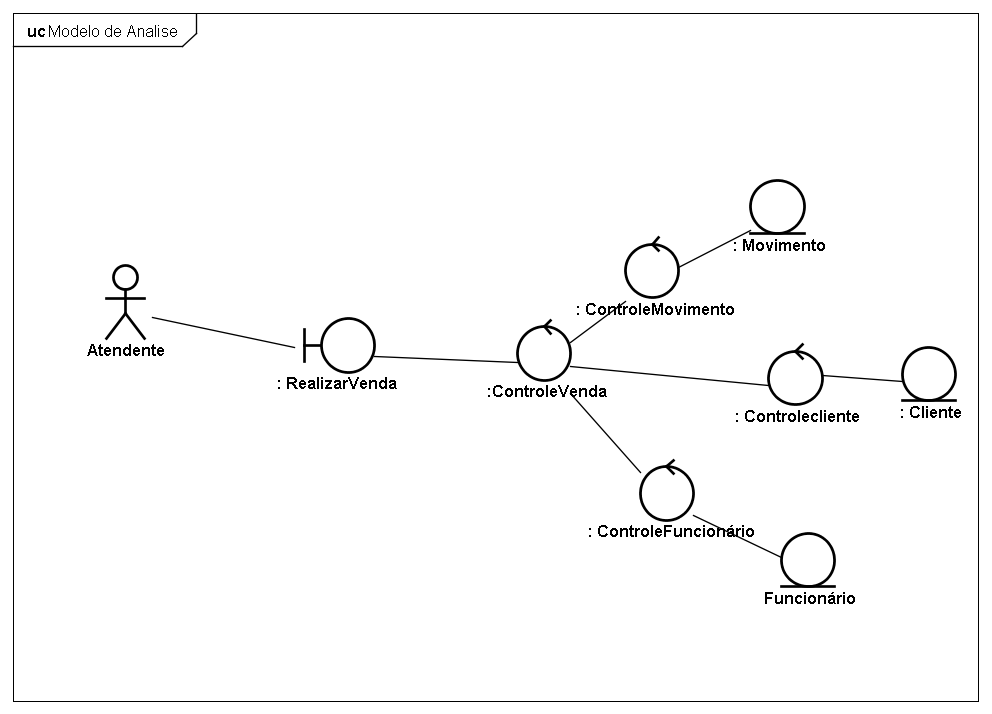
|  |
| --- |
| **Caso de uso:** Faturar serviço |
| **Responsável:** João |
| **Identificador:** F4 |
| **Descrição:** Com o produto do cliente pronto o técnico ira fatura o serviço, buscando a ordem de serviço do produto e gerando a fatura para ser paga pelo cliente. |
| **Ator (es):** Técnico |
| **Pré-condição:** O técnico deve ter efetuado login no sistema, e deve existir uma ordem de serviço para o produto. |
| **Pós-condição:** O serviço é faturado e armazenado no sistema para acesso do atendente posteriormente. |
| **Curso Normal:**   1. O Sistema mostra a tela de login. 2. O Técnico Insere login e senha. 3. O Sistema valida(OK) login e senha. 4. O Sistema mostra a tela inicial do Técnico. 5. O Técnico clica na opção faturar serviço. 6. O Sistema mostra a tela de faturar serviço. 7. O Técnico insere o nome do cliente. 8. O Sistema valida(OK) cliente. 9. O Técnico clica na opção inserir novo item. 10. O Sistema mostra tela do novo item. 11. O Técnico insere o valor de mão de obra. 12. O Técnico clica na opção de concluir inserção de item. 13. O Sistema mostra a tela de faturar serviço novamente. 14. As instruções (7,8) são executadas. 15. O Técnico seleciona os produtos usados no serviço de manutenção pelo menu de produtos cadastrados. 16. A instrução 10,11 são executadas. 17. O Técnico clica na opção imprimir fatura. 18. O Sistema salva os dados dessa fatura no banco de dados. |
| **Fluxo Alternativo:**  2.1. O Sistema mostra mensagem de erro de login ou senha.  2.2. O Técnico digita novamente login e senha.  2.1. O Sistema mostra mensagem de erro de usuário não cadastrado.  2.2. O Técnico solicita ao administrador do sistema o cadastro de usuário.  7.1. O Sistema mostra mensagem de erro de cliente não cadastrado.  7.2. O Técnico cadastra o cliente. |

# 2.6. Modelo de Análise:

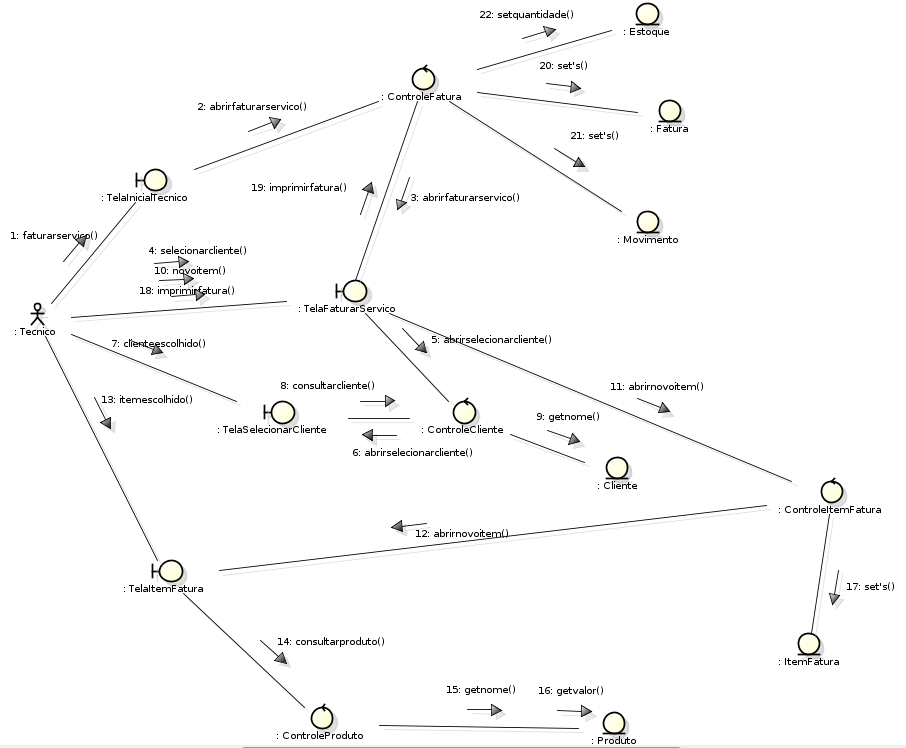
# 2.6.1. Gerar Ordem de Serviço

****

# 2.6.2. Realizar Venda



# 2.6.4. Faturar Serviço



# 2.7. Planejamento de Interações:

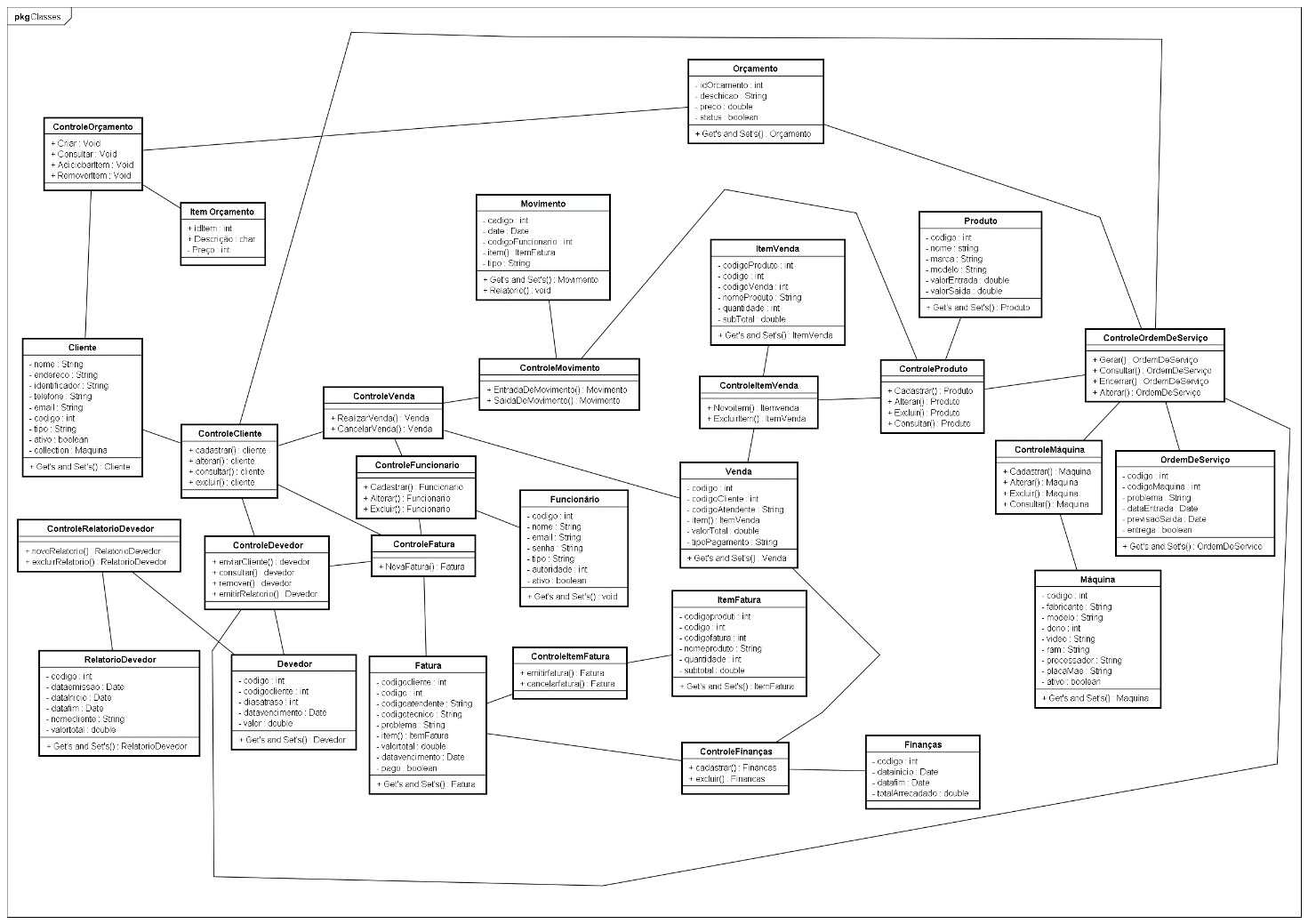
# 2.7.1. Interação 1 (cadastros básicos)

* Cadastrar Cliente
* Alterar Cliente
* Consultar Cliente
* Excluir Cliente
* Cadastrar Máquina
* Alterar dados da Máquina
* Consultar Máquina
* Excluir Máquina
* Cadastrar Funcionário
* Alterar Funcionário
* Buscar Funcionário
* Excluir Funcionário

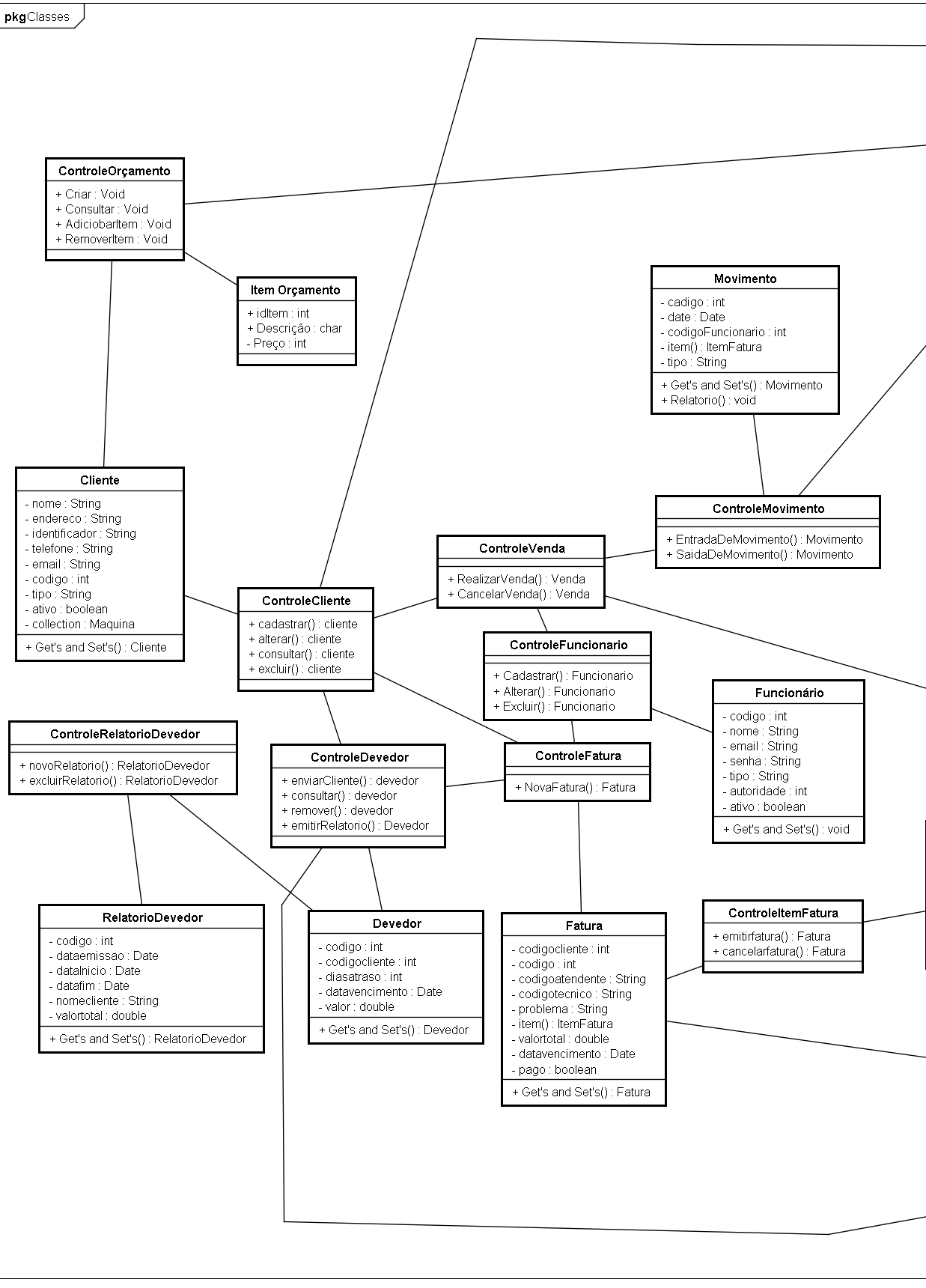
# 2.7.2. Interação 2..N:

* Gerar Ordem de Serviço
* Consultar Ordem de Serviço
* Encerrar Ordem de Serviço
* Alterar Ordem de Serviço
* Cadastrar Produto em Estoque
* Alterar Produto em Estoque
* Consultar Produto em Estoque
* Emitir Relatório de Estoque (histórico de movimento de estoque)
* Realizar Venda
* Faturar Serviço
* Gerar Valor Fatura
* Controle de devedores
* Emitir Relatório de Devedores
* Controle de Finanças

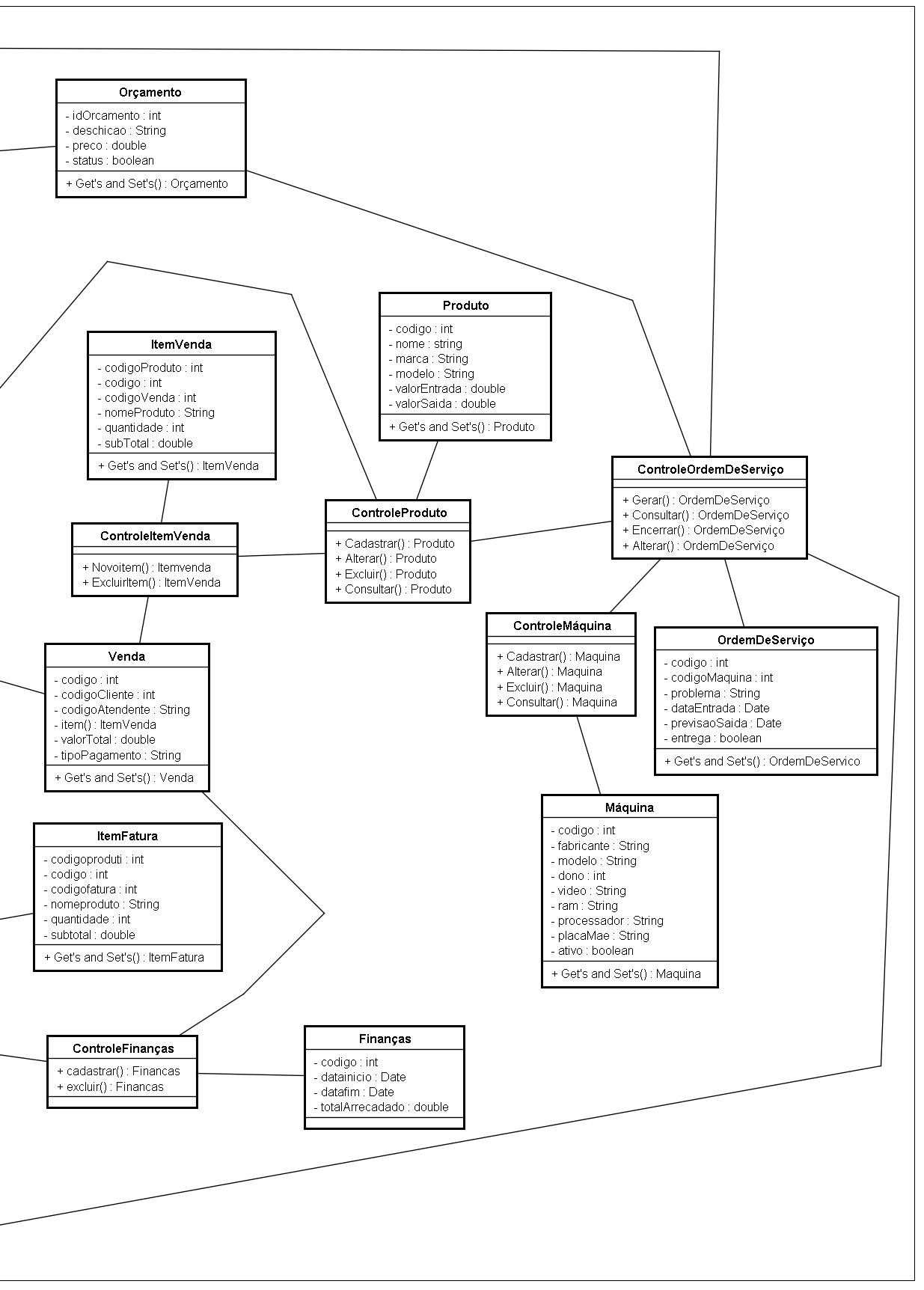
# 2.8. Diagrama de Classes do Projeto (Completo e Dividido):

****

**Figura 6.**

****

**Figura 7.**

****

**Figura 8.**

# 3. Glossário:

Fatura: Após encerrar a ordem de serviço, será possível emitir uma fatura para o cliente com o devido valor da manutenção.

Ordem de serviço: É o objetivo principal do *Service*. Nele ocorrerá todo o controle do processo quando todas as informações necessárias acima estiverem devidamente inclusas no sistema.

Cliente: Seria uma pessoa que tem algum problema em seu computador ou periférico e viria até a empresa para consertar seu equipamento.

Técnico: Responsável pela manutenção do equipamento, avisar o cliente os resultados tanto da avaliação técnica quanto conclusão do serviço.

Atendente: Funcionário responsável pela recepção do cliente e equipamento, cadastro de ambos casos necessários, e a geração da ordem de serviço para o equipamento. Será responsável também pela a entrega do equipamento e cobrar a fatura, além disso pela venda de peças que o cliente solicita como (teclado, *mouse,* mondem e etc.).

# Análise de Viabilidade

# Contribuição para a empresa

Algumas das contribuições ao qual o sistema irá trazer à empresa são:

* Aumento na velocidade na execução das operações da empresa, aumentando a produtividade.
* Formalização da informação no banco de dados e também na saída de dados.
* Redução de erros e custos (provenientes de retrabalho e falhas humanas).
* Aumento na confiabilidade dos dados, pois sempre haverá uma cópia dos dados no sistema, e não apenas em documentos impressos.
* Aumento na segurança das informações sigilosas, pois o sistema permitirá acesso às informações somente às pessoas autorizadas.

# Viabilidade Técnica

Tecnologia: o sistema pode ser implementado com a tecnologia atual, as linguagens e o sistema de banco de dados que serão utilizados estão disponíveis e são mantidos operacionalmente pelos distribuidores.

* Restrições de custo: As ferramentas utilizadas para o desenvolvimento são de licença gratuita.
* Restrições de prazo: Nosso prazo, até o final do semestre, é suficiente para o desenvolvimento do nosso sistema.
* Integração com outros sistemas existentes: Há princípio, não haverá integração com outros sistemas.

# Viabilidade Legal

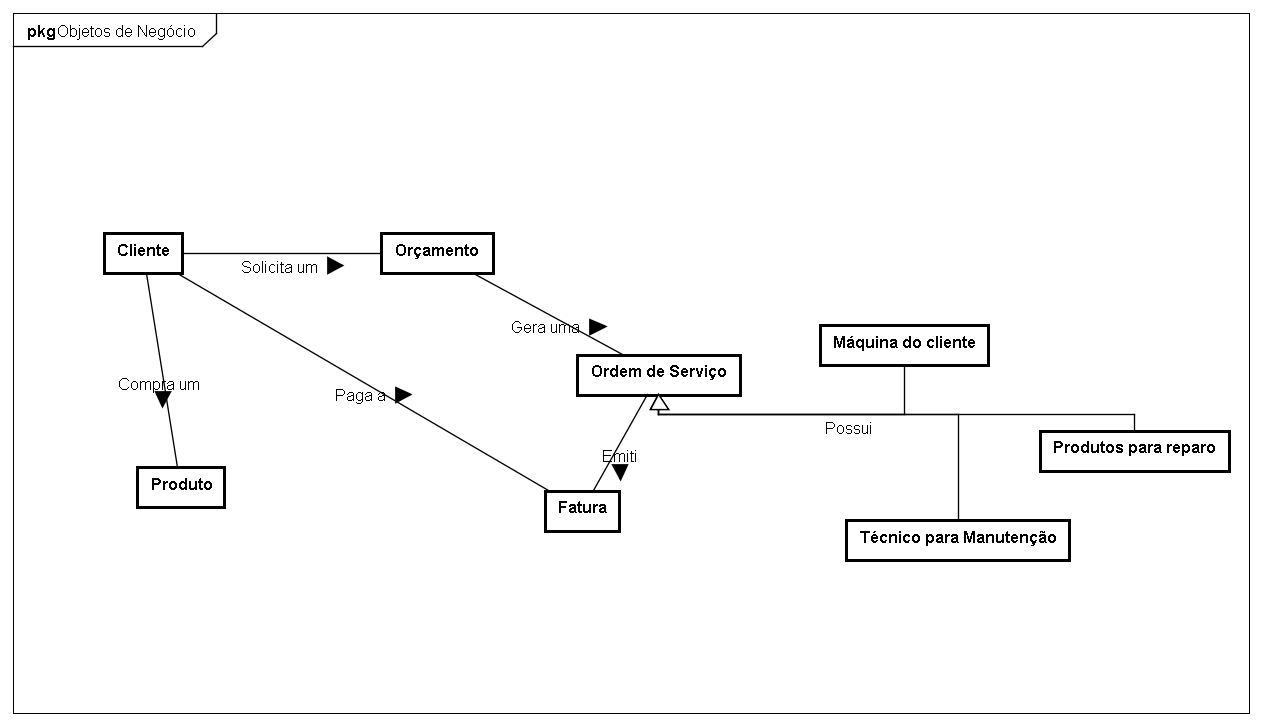
Um sistema de informações, como o descrito, não infringe ou viola a legislação as legislações federais, estaduais ou municipais.

# Análise de Risco

Identificação: Um risco corrente ao projeto é o nível de experiência e comprometimento dos membros da equipe, assim devemos documentas as responsabilidades de cada envolvido, para que o resultado planejado, seja equivalente ao resultado final. Devemos adotar tal política organizacional pois muitas equipes falham por não terem pessoal comprometido com o objetivo comum à equipe. Além da necessidade de um gerente (líder) que tenha a habilidade de lidar com os diferentes perfis correntes ao grupo e também é necessário que todos os envolvidos tenham um domínio prévio das tecnologias usadas no sistema. Priorização: alta. Resolução: efetuar treinamento para capacitação dos usuários e documentar as tarefas. Monitoração: supervisão periódica do serviço dos usuários.

Identificação: as disponibilidades de determinados atores do sistema podem acarretar em restrições operacionais devido as áreas especificas de o sistema serem acessíveis apenas a um ator, em sua ausência estas funcionalidades ficam inutilizáveis. Priorização: médio. Resolução: solicitar que haja um responsável por cada área do sistema, ou seja, na ausência de um funcionário, outro possa realizar sua função, com restrição a área de atuação do gerente.

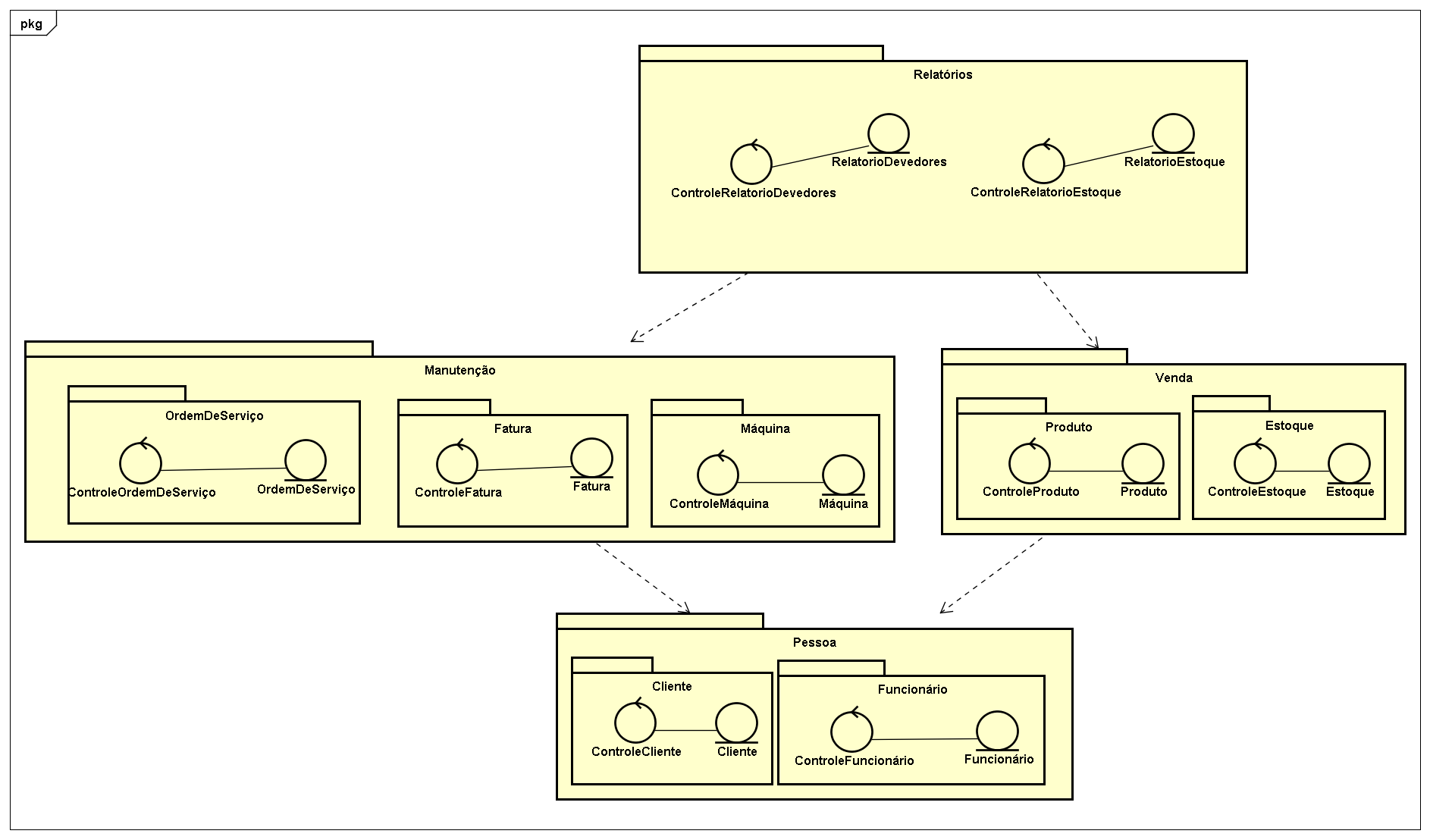
# Diagrama de Objetos de Negócios:



**Figura 1.**

O modelo de objeto de negócios conforme descrito na figura 1, também é representado por um diagrama de classes, temos uma visão de “alto nível” de quais são algumas das classes que irão fazer parte do sistema e como elas se relacionam.

# Diagrama de Pacotes:



**Figura 2.**

A figura 2 apresenta o diagrama de visão de negócio que é representado pelo diagrama de pacotes UML e por setas de dependência. Esse diagrama nos mostra que para que haja um gerenciamento da Ordem de Serviço isso dependerá das informações recebidas do Recursos Humanos (RH), do setor Financeiro da empresa e do Controle de Estoque.

# Arquiteturas de Software

# Arquiteturas não utilizadas

# Pipes e Filtros

Nosso sistema não requer vários estágios de processamento, esse estilo não é bem empregado divido ao processamento dos dados no sistema ser realizado de forma sequencial, ou seja, os processos não são independentes, logo, não há necessidade de aplicar esse estilo arquitetural no nosso projeto.

# Quadro Negro

Não há necessidade de um banco de dados onde todos os usuários acessem as informações, pois existem informações que são permitidas serem acessadas somente por uma das categorias de usuários. Portanto a arquitetura quadro negro traria baixa eficiência e alto esforço de desempenho assim, não existe há necessidade de utilizar este estilo arquitetural.

# Invocações Implícita

Os procedimentos (métodos) do nosso sistema serão invocados de forma direta e sequencial, além de não possuir um controle sobre quais componentes o sistema terá acesso, não sendo necessária a utilização desse estilo arquitetural.

# Orientada a eventos

Esse estilo não será usado na interface gráfica de nosso sistema, o sistema não cuida da manipulação de eventos, podendo causar conflito na manipulação de eventos, logo, este estilo não influenciará as decisões de projeto.

# Aplicações Distribuídas

Haverá um computador no qual estará o servidor de banco de dados e outros computadores que serão utilizados pelos funcionários para o manuseio do sistema. Como não usaremos em nosso sistema múltiplos processadores, nem multicomputares, as aplicações distribuídas não se enquadram em nosso sistema por ser um sistema simples.

# Ponto a Ponto

Apesar de ser uma aplicação distribuída, ela será centralizada, não havendo a possibilidade de utilizar o estilo arquitetural ponto a ponto semicentralizada, tão pouco a descentralizada.

# Sistemas Orientados a Serviços

Nosso sistema não disponibilizará serviços a terceiros, e temos o receio de utilizar tal arquitetura por ela possuir uma experiência limitada quanto a arquitetura orientada. Logo, não existe a necessidade deste estilo arquitetural.

# Estilos Arquiteturais – Variações

Baseado na arquitetura CORBA, não iremos utilizar tal arquitetura, pois cliente e máquina se comunicam usando a mesma máquina, o que é contrário a ideia da arquitetura CORBA, a qual se baseia na comunicação via rede.

# Cliente-Servidor de duas camadas

Não será utilizado a arquitetura cliente-servidor de duas camadas modelo cliente gordo, pois esta abrange sistemas robustos, além de demandar uma gama enorme de processamento e ser responsável por gerenciar sistemas complexos, opondo-se ao nosso sistema.

# Cliente-Servidor de múltiplas camadas para Web

Nosso sistema não disponibilizara servidor via Web, portanto não existe a necessidade desta arquitetura.

# Sistema de tempo Real

Nosso sistema não será implementado em linguagens de baixo nível e não necessita de uma resposta em tempo real, logo não existe o porquê de utilizar tal arquitetura.

# Em Nuvem

Nosso sistema não disponibilizara acesso à rede (internet) logo, não sendo possível fazer acesso remoto ao sistema. Portanto não existe a necessidade deste estilo arquitetural.

# Arquiteturas Utilizadas

# Objetos

Nosso sistema vai ser composto de várias partes comunicantes e também será desenvolvido com paradigma de análise, projeto e programação orientação a objetos, que está relacionado com esse estilo, também utiliza reuso como ferramenta, além de ter a manutenabilidade como um requisito não funcional, isto proporciona ao sistema que qualquer modificação nos requisitos irá afetar o menor número de objetos possíveis este estilo arquitetural será utilizado no nosso sistema.

# Sistema de Informação (Cliente Magro)

Dentre as subdivisões analisadas de sistema de informações, definimos como a mais adequada para nosso sistema a arquitetura cliente servidor de duas camadas modelo magro, pois trata sistemas de pequeno porte e o cliente ser responsável simplesmente por executar o software, assim o sistema faz todo o gerenciamento dos dados, está também possui a vantagem de ser flexível, permitindo mudanças frequentes.

# MVC (Modelo Visão Controle)

O padrão arquitetural Model-View-Controller (MVC) é uma forma de quebrar uma aplicação, ou até mesmo um pedaço da interface de uma aplicação, em três partes: o modelo, a visão e o controlador.

No padrão MVC, teríamos então o mapeamento de cada uma dessas três partes para o padrão MVC. Optamos utilizar esse padrão de projeto pois através da sua separação facilita manipular os objetos além de facilitar quando está sendo implementado.

# Frameworks:

O framework escolhido para ser utilizado em nosso projeto foi o Sails o qual possui uma tecnologia avançada, atualmente está surgindo duas importantes inovações:

* Primeiramente é a transição generalizada de uma abordagem puramente baseado em servidor para a construção de interfaces com o usuário para uma abordagem mais centrada no cliente;
* Segundo, é adoção generalizada de JavaScript fora do browser e, mais especificamente, no servidor via Node.js.

# Vantagens e Desvantagens:

Abaixo temos as grandes vantagens desse framework:

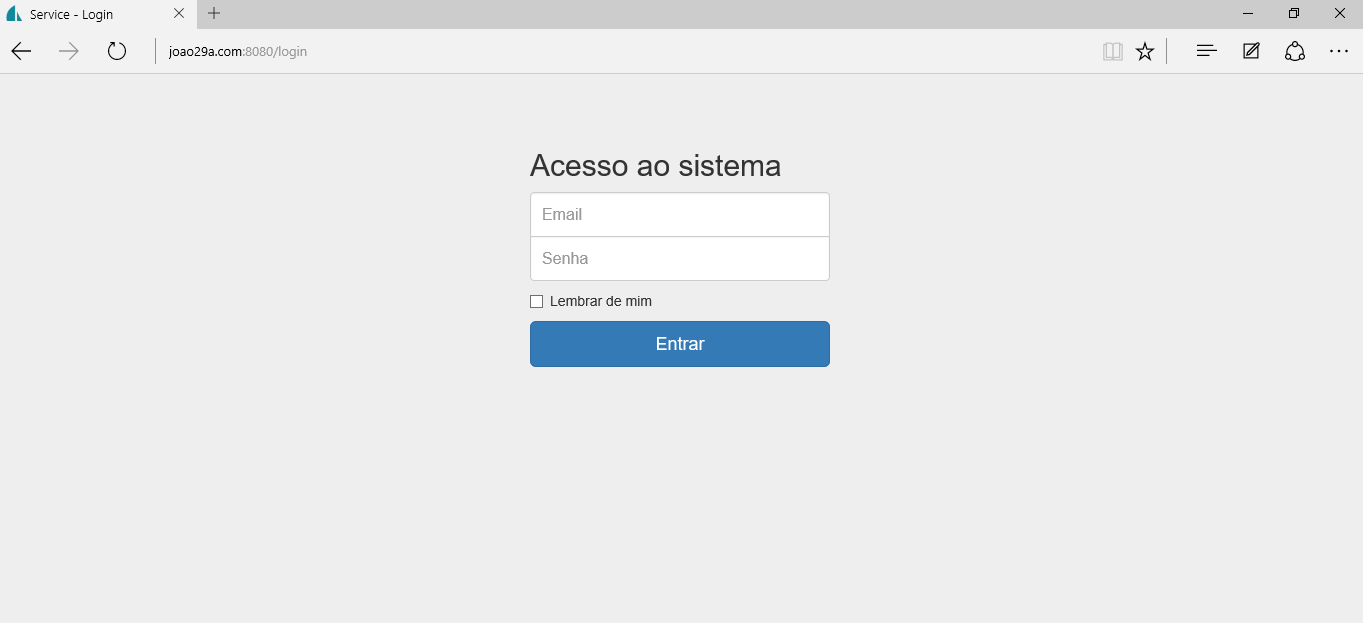
* Facilita a tarefa de construir verdadeiramente personalizado, aplicações de Node.js de nível empresarial;
* Sails fornece um caminho reconhecível para aqueles familiarizados com frameworks MVC, como Ruby on Rails, mas com o suporte para requisitos de novos aplicativos modernos: orientado a dados, APIs com uma arquitetura escalável, orientada a serviços. Isso faz com que o Sails seja uma estrutura ágil e rápida
* Criar o esqueleto da aplicação;
* O adaptador de campo determina onde os dados serão armazenados, sendo possível fazer a combinação entre MySQL e Redis
* Base de dados de agnóstico ORM que é simples, mas poderosa e flexível, bem
* Desenvolvimento de rotas / REST APIs é muito rápido;
* Diversas coisas “simplesmente funcionam" sem fazer o desenvolvedor pensar muito são funcionalidades quase automáticas, e o melhor é que o desenvolvedor sabe o que está acontecendo no Sails essas funcionalidades não são escondidas como "mágica";

Entretanto, o framework possui algumas desvantagens também:

* O fato de ser uma tecnologia recente, por vezes, problemas surgem que pode ser frustrante para lidar;
* Falta de documentação em alguns aspectos.

# Funcionamento do Framework:

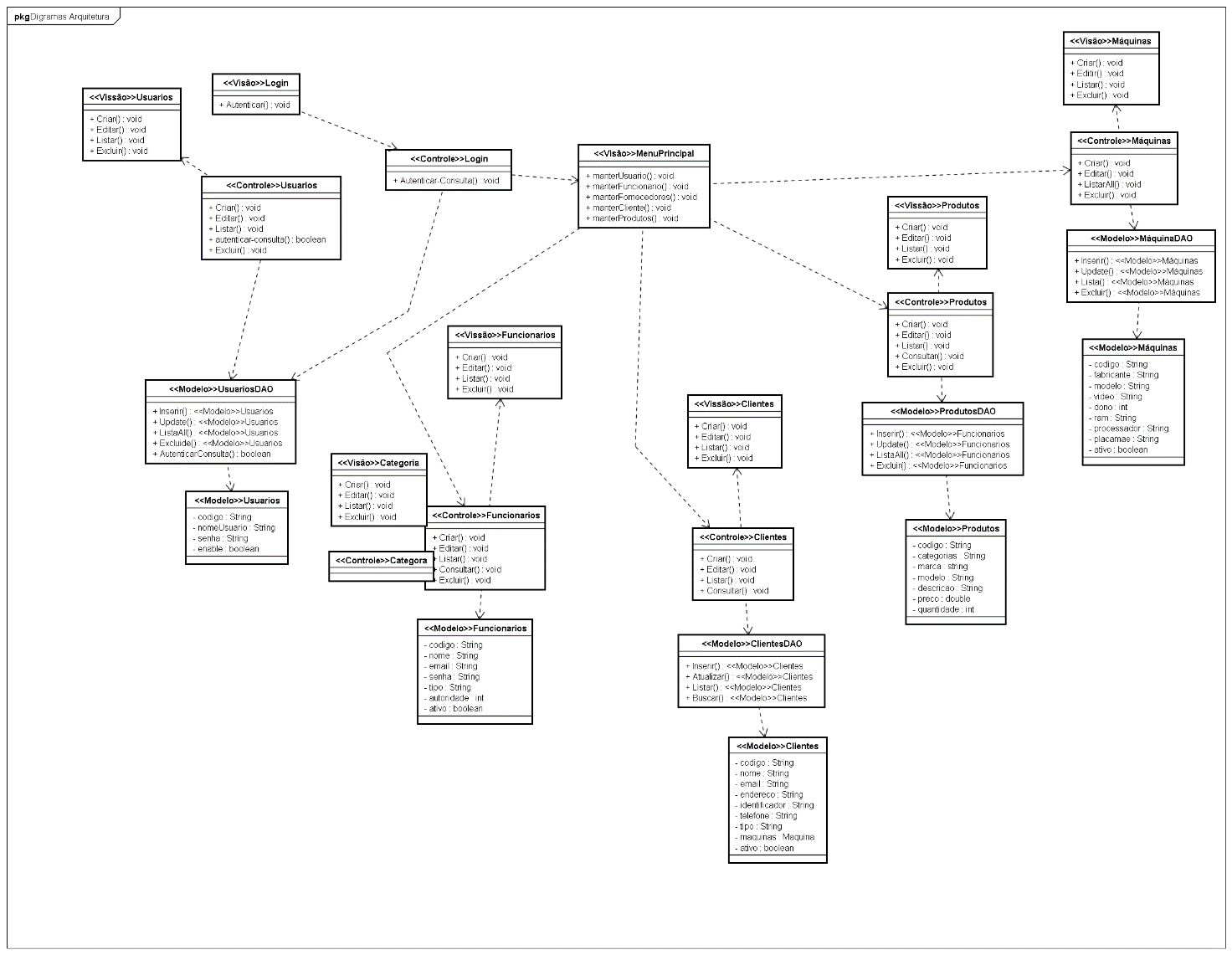
O Sails torna mais fácil para criar o servidor back-end que precisa de seu front-end para atingir a finalidade do aplicativo. Esses serviços podem incluir tudo de entregar do front-end para se comunicar com o banco de dados para interagir com 3 serviços de terceiros. Mais importante, Sails automatiza tarefas repetitivas, encontradas em todas as aplicações e combina-os com um poderoso conjunto de ferramentas que faz executar seu aplicativo mais rápido, repetitivo e mais confiável.



Um exemplo de como Sails age no back-end e front-end é a tela de Login do sistema. O objetivo dessa tela é permitir ao usuário provar ser quem diz, para isso ele irá fornecer *username* e *password* essas duas informações fazem parte do back-end para que possa executar a autenticação. Já o front-end é responsável por reconhecer se o botão de Login foi clicado e enviar o nome de usuário e senha para o back-end.

Em uma aplicação web como é o *Service* consiste simplesmente de um front-end que faz solicitações de um servidor back-end e esse por sua vez, o servidor back-end responde a essas solicitações de front-end com tudo o que foi solicitado.

# Diagrama de Classe Arquitetural:

****

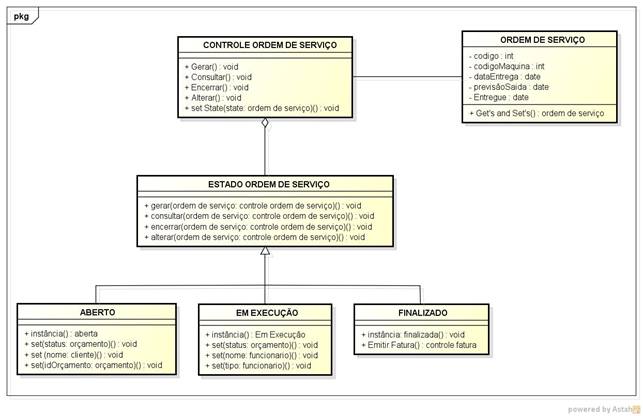
# Padrão de Projeto:

Padrão de projeto: STATE

É um padrão de projeto de software usado quando um componente de um objeto, dependendo do seu estado. Ou seja, defini uma interface para encapsular o comportamento com um estado particular do contexto.

A vantagem de se utilizar o padrão de projeto STATE é devido a fácil localização das responsabilidades de estados específicos, devido a que os encontram nas classes que correspondem a cada estado. Isso proporciona maior clareza no desenvolvimento, e na manutenção posteriormente. Pelo fato que diferentes estados são representados por um único atributo (estado) e não envolvido em diferentes variáveis e grandes condicionais.

A motivação encontrada em nosso sistema para a implementação desse padrão de projeto. E de encapsular diferentes comportamentos da classe Controle Ordem de Serviço. Em estados que ela pode assumir, com isso conseguimos delegar as responsabilidades dos estados específicos.

****

# Diagrama de Componentes:

1. 