

# Sistema de Gerenciamento de Vidraçarias

Professor Dr. Sérgio Felipe Zirves  
Engenharia de Software N

Cristiano Medeiros	Fábio Beltrão	Juliano Gonçalves
173362	173361	172059

## 1 Introdução

Este trabalho busca abranger todos os conhecimentos adquiridos durante a disciplina de Engenharia de Software N, através da confecção de especificação formal para um software destinado à gerência da vidraçaria de propriedade do Sr. Rudinei Medeiros Hoff, cliente escolhido pelo grupo.

A vidraçaria se concentra principalmente no corte e venda de vidros e materiais auxiliares como maçanetas, dobradiças, etc. O modelo de negócio implantado requer a compra de placas de vidro de tamanho pré-definido e o posterior corte em dimensões de multiplicidade fixa para venda ao cliente final. Esse processo leva a um excesso de material, chamado de retalho, que deve ser reaproveitado de forma a produzir o mínimo de desperdício possível.

A situação atual da gerência da vidraçaria apresenta diversos problemas, segundo o que foi discutido inicialmente com o proprietário. Pelo fato de o processo ser totalmente manual, existe uma preocupação constante do responsável com a documentação das vendas efetuadas (incluindo controle de pagamento e agendamento de entrega de produtos), com cadastro de clientes e sua atualização, com o controle de estoque e com a reutilização de materiais. Esses problemas contribuem para perdas mensais perceptíveis de receita, e esse foi o ponto crítico no qual foi baseada a necessidade de informatizar o gerenciamento.

O cliente propôs a informatização parcial do estabelecimento, se concentrando principalmente no que engloba o cadastramento de clientes, a gerência do estoque, as operações de venda, e a análise financeira de custos e ganhos. Não foi percebida por ele a urgência de implementação de um sistema de RH completo com folha de pagamento, ou um esquema de comunicação entre filiais, por exemplo, dado o pequeno porte do comércio.

## 2 Metodologia

A especificação a seguir segue os princípios da análise e projeto orientados a objetos (APOO). Essa metodologia visa estabelecer um padrão de desenvolvimento de software de forma a obter melhores resultados, tanto em termos financeiros como em questões de tempo de desenvolvimento. Por apresentar um processo de especificação com diversos ciclos de desenvolvimento, tal técnica resiste de forma satisfatória à constante mudança das características esperadas do software (também chamadas de requisitos) por parte do cliente, fenômeno cada vez mais presente no ramo.

A APOO, por sua vez, enfatiza a utilização dos métodos e diagramas definidos pela UML (Unified Modelling Language), com o pretexto de facilitar a comunicação dos diversos aspectos e estágios de criação do software, tanto com o contratante da empresa de desenvolvimento (por vezes representado pelo usuário final, pelo diretor de uma companhia, etc), quanto com o time responsável pela codificação do programa final. O grupo utiliza, dentre todos os modelos contidos na UML, apenas um subconjunto, considerado uma base sólida para o desenvolvimento consistente de sistemas. Tal subgrupo engloba estruturas como a Tabela Hierarquica de Funções, o Diagrama Conceitual, a descrição dos Casos de Uso, os Diagramas de Sequência de Caso de Uso, e finalmente o Diagrama de Classes. Esses modelos serão apresentados nessa ordem no decorrer do trabalho (representando com certa precisão a ordem temporal e lógica da criação de cada um durante o processo de especificação), juntamente com uma breve explicação do funcionamento e da sua utilidade individual.

Obviamente, a especificação de um software precede a implementação do mesmo e, consequentemente, a cobrança de um valor justo a ser pago pelo cliente. Para tanto, são necessárias métricas precisas para um cálculo coerente do custo de desenvolvimento, já que esse deve satisfazer as necessidades financeiras da equipe de programadores, analistas e testadores enquanto, simultaneamente, se aproxima do valor que o cliente está disposto a pagar. Com o propósito de atingir tal objetivo, foi acordado o uso do método conhecido como Use Case Points que, baseado nos casos de uso definidos, nas necessidades do software e da equipe, e no tempo de duração para a confecção do programa, atribui valores reais para cada critério e finalmente produz um valor líquido total para o desenvolvimento completo do projeto.

O uso das ferramentas e técnicas indicadas acima foi completamente embasado no estudo feito pelo grupo através da disciplina de Engenharia de Software N. Elas foram aplicadas no caso corrente visando um aprendizado mais abrangente no que diz respeito ao processo de elaboração de um projeto realista, e em nenhum momento com o intuito de vender a solução. Nota-se que mesmo com essa peculiaridade, o grupo vê possibilidades reais de implementação do sistema num futuro bastante próximo, possivelmente num intervalo menor do que um ano.

## 3 Participantes

### 3.1 Cliente

O cliente é o Sr. Rudinei Hoff, proprietário da Vidraçaria Vidrex. O negócio é basicamente a venda de vidros cortados, para diversos clientes.

### 3.2 Analistas

Os analistas do sistema são Cristiano Medeiros Dalbem, Fábio da Fontoura Beltrão e Juliano Leal Gonçalves, alunos de Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Todos cursam a disciplina Engenharia de Software N, ministrada pelo professor Dr. Sérgio Felipe Zirbes. Esse é o primeiro contato dos alunos com uma metodologia formal de especificação de sistemas.

### 3.3 Glossário

**Chapa de Vidro:** É formato como o vidro é comprado do revendedor. A chapa virá em tamanhos pré-determinados (ver Multiplicade de Dimensão), que não necessariamente corresponderá ao formato de vidro que será vendido na vidraçaria.

**Multiplicidade de Dimensão:** No ramo de vidros cada tipo de material dos quais são feitas chapas implica numa multiplicade específica nas dimensões. Por exemplo, um certo tipo de vidro tem multiplicade 5 nas dimensões, ou seja, toda chapa que o vidraceiro for comprar terá que ter tamanho de 5x5, 10x10, 15x15, etc.

**Otimização de Corte:** É do interesse do vidraceiro que, ao fazer os cortes dos vidros que ele venderá nas chapas do estoque, tal seja feito de maneira que minimize as partes que sobram.

**Pecas Reutilizáveis:** Se diz das partes restantes dos cortes nas chapas. Esse conjunto ao longo do tempo adquire grandes proporções, sendo que em muitos casos os restos são de tamanhos que poderiam ser utilizados para novas vendas.

## 4 Cronograma

1.	Escolha do assunto e identificação do usuário	18 de março
2.	Definição do escopo (DHF)	30 de março
3.	Diagramas e Especificação dos Casos de Uso	22 de abril
4.	Modelo Conceitual	11 de maio
5.	Diagramas de Interação	01 de junho
6.	Modelo de Classes/Objetos	15 de junho
7.	Apresentação do trabalho	22 e 24 de junho
8.	Prazo final para entrega da especificação	01 de julho

## 5 Tabela Hierárquica de Funções

Baseado nas discussões com o usuário foram identificadas as principais funções do sistema, as quais foram a Tabela Hierárquica de Função, vista na Tabela 5. A tabela apresentada é a versão final concluída pelo grupo, lembrando que a tabela evolui conforme o cliente e os analistas vão se entendendo e entendendo exatamente o que deveria ser o sistema. Foi interessante o fato de que a primeira tabela confeccionada pelo grupo era *totalmente* diferente da atual.

Função	Categoria
1. Gerenciamento de Pessoas	
1.1 Gerenciar clientes	
1.1.1 Cadastrar cliente	Evidente
1.1.2 Editar dados do cliente	Evidente
1.3 Visualizar dados de cadastro	Opcional
1.4 Pesquisar por pessoa	
1.4.1 Pesquisar por nome	Evidente
1.4.2 Pesquisar por RG	Evidente
1.4.3 Pesquisar por CPF	Evidente
1.4.4 Pesquisar por CNPJ	Evidente
2. Gerenciamento de Vendas	
2.1 Nova venda	
2.1.1 Inserir item	Evidente
2.1.2 Remover item	Evidente
2.1.3 Editar item	Evidente
2.2 Confirmar Venda	Evidente
2.3 Cancelar Venda	Evidente
2.4 Aplicar multiplicidade de dimensão de vidro	Oculto
2.5 Aplicar algoritmo de otimização de corte	Oculto
2.6 Testar retalho de vidro	Oculto
2.7 Impressão de vias	
2.7.1 Imprimir relatório de venda completo	Evidente
2.7.2 Imprimir relatório de venda simplificado	Evidente

<b>Função</b>	<b>Categoria</b>
3. Gerenciamento de Estoque	
3.1 Visualizar estoque	Opcional
3.2 Gerenciar estoque	
3.2.1 Inserir no estoque	Evidente
3.2.2 Remover do estoque	Evidente
3.2.3 Editar itens do estoque	Evidente
3.2.4 Avisar quando item de estoque estiver acabando	Oculto
3.3 Gerenciar estoque de peças reutilizáveis	
3.3.1 Inserir no estoque	Evidente
3.3.2 Remover do estoque	Evidente
3.3.3 Editar itens do estoque	Evidente
3.4 Pesquisar no estoque	
3.4.1 Pesquisar por dimensão	Evidente
3.4.2 Pesquisar por tipo	Evidente
4. Histórico de Vendas	
4.1 Pesquisar no histórico	
4.1.1 Pesquisar por cliente	Evidente
4.1.2 Pesquisar por data	Evidente
4.1.3 Pesquisar por tipo de vidro	Evidente
4.2 Gerenciar dados de Histórico	
4.2.1 Adicionar registro	Oculto
4.2.2 Editar registro	Oculto
4.3 Calcular juros	Oculto
4.4 Visualizar dados de venda	Opcional
4.5 Visualizar entregas pendentes	Evidente
5. Finanças	
5.1 Controle de despesas	Evidente
5.2 Controle de ganhos	Evidente
5.3 Visualizar balanço	
5.3.1 Visualizar gráfico	Opcional
5.3.2 Visualizar tabela	Opcional
5.4 Login de Administrador	Evidente

Tabela 1: Tabela Hierarquica de Funções

## 6 Casos de Uso

Tendo a THF pode-se então descrever os Casos do Uso do sistema, que tem por objetivo indicar, de forma descritiva, as relações entre as funções definidas anteriormente, bem como as ações que os usuários do sistema e o próprio sistema irão executar. Um caso de uso é, basicamente, uma sequência de ações e reações de um ator (humano) ao sistema e vice-versa.

### 6.1 Descrição dos Casos de Uso

Segue abaixo as descrições dos casos de uso, que são de fundamental importância para a continuidade do projeto, pois é a partir delas que se derivam os demais diagramas necessários, afim de chegar ao diagrama de classes, a partir do qual já se tem praticamente as classes que serão implementadas em código.

#### 6.1.1 Caso de Uso: Vender Vidro

**Atores:** Cliente(Iniciador), Vidraceiro

**Propósito:** Vender vidros.

**Descrição:** O Vidraceiro lucra da venda de seus vidros. Este caso de uso representa a venda de vidros para algum cliente.

**Tipo:** Primário e Essencial

**Referência:** 1.1.1, 1.3, 1.4.1, 1.4.2, 1.4.3, 1.4.4, 2.1.1, 2.1.2, 2.1.3, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7.1, 2.7.2, 3.1, 3.2.4, 3.4.1, 3.4.2, 4.2.1

<b>Típica Sequência de Eventos</b>	<b>Resposta do Sistema</b>
1. Este caso de uso inicia quando um cliente quer comprar um vidro. O Vidraceiro inicializa a venda no sistema.	2. O sistema instancia uma nova venda, solicitando uma identificação para o cliente.
3. O Vidraceiro entra no sistema um dado do cliente.	4. O sistema busca no banco de dados por clientes compatíveis com aquele dado, retornando-os
5. O Vidraceiro confirma com o cliente se ele é algum dos possíveis cadastrados. Caso o cliente já esteja cadastrado o Vidraceiro seleciona o cliente do banco de dados, vá para 9. Caso contrário, o Vidraceiro deve inserir um novo cliente no banco de dados.	6. O sistema abre um formulário para o Vidraceiro preencher com os dados do novo cliente.
7. O cliente informa os seus dados e o Vidraceiro os insere no sistema.	8. O sistema insere o cadastro do novo cliente no banco de dados.
	9. O sistema abre um formulário para preencher os dados da compra.
10. O cliente informa para o Vidraceiro quais as dimensões e tipo(s) do(s) vidro(s) que ele deseja comprar. O vidraceiro insere no sistema tais informações, assim como a(s) quantidade(s), data de entrega, prazo de pagamento.	11. Neste momento o sistema arredondará as medidas para o padrão e calcula o valor parcial da linha de venda. Se ainda houver mais vidros na compra vá para 10.
	12. Tendo terminada a entrada de vidros, o sistema: faz o planejamento do corte (cujo resultado será emitido no relatório de corte), calcula o valor total e consulta os estoques de vidros.
	13. O sistema verificará primeiramente no estoque de peças reutilizáveis. Se não encontrar um vidro compatível, fará então uma pesquisa do estoque de vidros novos. Se não encontrar neste também, o sistema emitirá um requerimento para compra de nova peça.
14. O Vidraceiro pode agora editar a venda, vá para 10. Ele pode cancelar a venda, encerrando o sistema. Ou ele pode finalmente confirmar a venda.	15. O sistema insere os dados da nova venda no histórico de vendas e emite três vias referentes à compra: uma para o cliente, uma para o próprio Vidraceiro (ambas contendo os valores da venda) e outra que vai para os funcionários encarregados de fazer o corte (esta última sem valores). Em seguida o sistema encerra a venda.



### 6.1.2 Caso de Uso: Confirmar Entrega

<p><b>Atores:</b> Operador, Entregador(Iniciador)</p> <p><b>Propósito:</b> Efetuar a confirmacao da entrega de uma venda no sistema.</p> <p><b>Descrição:</b> Para que uma venda seja concluída na vidraçaria, são necessários alguns pre-requisitos, sendo um deles a confirmação de entrega dos produtos. Esse caso de uso efetivamente confirma a entrega de um conjunto de mercadorias relacionadas com uma certa venda para um dado cliente.</p> <p><b>Tipo:</b> Primário, Essencial</p> <p><b>Referência:</b> 4.2.2, 4.4</p>	
Típica Sequência de Eventos	Resposta do Sistema
1. Este caso de uso inicia quando o entregador reporta ao operador o sucesso na entrega das mercadorias, informando o numero de pedido associado. O operador entao procede inserindo o numero no sistema.	2. O sistema retorna com as informações da venda solicitadas pelo operador.
3. O operador seleciona, dentre as opções possíveis, a confirmação da entrega.	4. O sistema verifica a situação atual da entrega para a venda em questão. Caso essa já esteja confirmada, produz um aviso ao operador indicando que a situação atual já esta correta e encerra a operação. Caso contrario, o sistema seta um estado interno indicando que a venda teve suas mercadorias entregues com sucesso.
	5. O sistema verifica se a venda já havia sido paga anteriormente. Caso sim, o sistema seta um estado interno de encerramento de venda encerra a operação. Caso não, encerra a operação sem mais modificações.

### 6.1.3 Caso de Uso: Confirmar Pagamento

<p><b>Atores:</b> Operador(Iniciador), Cliente</p> <p><b>Propósito:</b> Confirmar pagamento de uma venda já iniciada.</p> <p><b>Descrição:</b> O cliente irá até a vidraçaria e fará o pagamento de uma venda já feita, podendo já ter sido entregue. Se tiver passado de um prazo estipulado no momento da venda, o sistema calculará juros.</p> <p><b>Tipo:</b> Primário e Essencial</p> <p><b>Referência:</b> 4.2.2, 4.3, 4.4</p>	
Típica Sequência de Eventos	Resposta do Sistema
1. Este caso de uso se inicia quando um cliente chega na filial da vidraçaria e expressa seu desejo de realizar o pagamento de uma venda, ou após a definição da venda, ou após a confirmação por parte do operador de que um pagamento foi realizado por depósito bancário.	2. O sistema pedirá informações que identificam a venda e então irá procurá-la no banco de dados.
3. O operador fornece as informações necessárias.	4. Se a venda já tiver sido paga o sistema acusa erro. Caso contrário, é checado se o pagamento está sendo feito com atrasado. Se sim, é calculado juros sobre o preço registrado. É informado o preço a ser pago.
5. O operador confirma a operação.	6. O sistema marca que a venda foi paga. Se ela já tiver sido entregue, a venda é marcada como finalizada. Encerra-se a operação.

#### 6.1.4 Caso de Uso: Visualizar Estoque

<b>Atores:</b> Operador(Iniciador) <b>Propósito:</b> Obter informações atuais do estoque. <b>Descrição:</b> A vidraçaria mantém um relatório atualizado de todo material presente no estoque. Esse caso de uso trata da visualização do estoque. <b>Tipo:</b> Secundário <b>Referência:</b> 3.1, 3.4.1, 3.4.2	
Típica Sequência de Eventos	Resposta do Sistema
1. O caso de uso inicia quando o Operador do sistema solicita acesso para visualização corrente do estoque da loja.	2. O sistema responde a solicitação requisitando do Operador que informe qual dos estoques deseja visualizar, através de um menu de opções.
3. O Operador seleciona qual dos estoques deseja visualizar, dentre o estoque de material novo e o estoque de peças reutilizáveis.	4. O sistema retorna com as informações correntes dos materiais no estoque solicitado, e encerra a operação.

### 6.1.5 Caso de Uso: Inserir no Estoque

<p><b>Atores:</b> Operador(Iniciador)</p> <p><b>Propósito:</b> Atualizar o estoque de produtos.</p> <p><b>Descrição:</b> A vidraçaria efetua compras de novas placas de vidro e as recorta para venda aos seus clientes. Esse caso de uso trata da inserção das informações das novas placas de vidro adquiridas pela vidraçaria através de pedidos a fornecedores.</p> <p><b>Tipo:</b> Secundário</p> <p><b>Referência:</b> 3.2.1, 3.2.3, 3.3.1, 3.3.3, 3.4.1, 3.4.2</p>	
Típica Sequência de Eventos	Resposta do Sistema
1. O caso de uso inicia quando o Operador do sistema solicita acesso para insercao de materiais no estoque da loja.	2. Como sao inseridos apenas materias novos, o sistema solicita ao Operador qual o tipo de material, suas dimensoes e a quantidade.
3. O Operador informa os dados solicitados atraves de um formulario na tela e solicita insercao no estoque ao sistema.	4. O sistema valida as informacoes, e caso algum dos dados seja inconsistente, retorna ao passo anterior de solicitacao de informacao e emite uma mensagem de erro ao Operador.
	5. O sistema verifica se ja existe um material do mesmo tipo e com as mesmas dimensoes no estoque. Caso afirmativo, a quantidade em estoque do material e atualizada somando-se a quantidade corrente com a quantidade solicitada. Caso contrario, o sistema cria uma nova entrada no estoque com o item solicitado. A operacao é então encerrada.

### 6.1.6 Caso de Uso: Remover do Estoque

<b>Atores:</b> Operador(Iniciador) <b>Propósito:</b> Atualizar o estoque de produtos. <b>Descrição:</b> A vidraçaria utiliza os vidros do estoque para corte. Esse caso de uso trata da remoção das placas de vidro utilizadas em um certo corte de uma certa venda. <b>Tipo:</b> Secundário <b>Referência:</b> 3.1, 3.2.2, 3.2.3, 3.3.2, 3.3.3, 3.2.4, 3.4.1, 3.4.2	
Típica Sequência de Eventos	Resposta do Sistema
1. O caso de uso inicia quando o Operador do sistema solicita acesso para remoção de materiais no estoque da loja.	2. O sistema faz uma listagem ao usuário dos vidros do estoque, permitindo a busca por tipo e dimensões do vidro para facilitar.
3. O Operador informa os dados para buscar pelo vidro desejado e solicita a remoção dele.	4. O sistema verifica se existia mais de um material do mesmo tipo e com as mesmas dimensões no estoque. Caso afirmativo, a quantidade em estoque do material é atualizada, subtraindo-se da quantidade corrente a quantidade solicitada para remoção. Caso seja encontrada apenas uma ocorrência, o item é excluído do estoque. É esta então a operação encerrada.

### 6.1.7 Caso de Uso: Editar o Estoque

<p><b>Atores:</b> Operador(Iniciador)</p> <p><b>Propósito:</b> Atualizar o estoque de produtos.</p> <p><b>Descrição:</b> A vidraçaria efetua compras de novas placas de vidro e as recorta para venda aos seus clientes. Esse caso de uso trata da atualização ou remoção das informações de tipo e quantidade de vidros presente no estoque, tanto de material novo como de retalhos de placas de vidro reutilizáveis.</p> <p><b>Tipo:</b> Secundário</p> <p><b>Referência:</b> 3.1, 3.2.3, 3.3.3, 3.4.1, 3.4.2</p>	
Típica Sequência de Eventos	Resposta do Sistema
1. O caso de uso inicia quando o Operador do sistema solicita acesso para edição de algum item no estoque da loja.	2. O sistema responde a solicitação requisitando do Operador que informe qual dos estoques deseja visualizar, através de um menu de opções.
3. O Operador seleciona qual dos estoques deseja editar, dentre o estoque de material novo e o estoque de peças reutilizáveis.	4. O sistema retorna uma lista classificada dos materiais presentes no estoque selecionado, permitindo classificação por código, tipo, dimensão, quantidade ou data de inserção, além de prover um mecanismo de busca para o tipo de material. Logo após, o sistema solicita ao Operador que selecione um dos itens do estoque dentre aqueles apresentados, ou busque por um tipo específico através do campo de busca.
5. O Operador seleciona um dos itens do estoque na tela ou procede com uma busca por tipo de material.	6. Caso o Operador tenha optado por buscar por tipo, o Sistema varre o banco de dados do estoque e retorna uma listagem com os resultados, voltando para o passo anterior. Caso contrário, o Sistema procede, de posse do item selecionado pelo Operador, com uma solicitação para os dados a serem alterados.
7. O Operador modifica os campos para refletir a nova realidade do estoque físico e remete a informação.	8. O Sistema atualiza o item selecionado com as novas informações e encerra a operação.

### 6.1.8 Caso de Uso: Pesquisar no Histórico de Vendas

<b>Atores:</b> Operador (Iniciador) <b>Propósito:</b> Encontrar dados de vendas antigas <b>Descrição:</b> Existirá um banco de dados com registro de todas vendas já feitas pelo sistema. O operador decidirá se quer pesquisar por cliente para quem foi feita a venda, por data em que ela foi feita ou por tipo de vidro vendido. <b>Tipo:</b> Secundário <b>Referência:</b> 4.1.1, 4.1.2, 4.1.3, 4.4	
Típica Sequência de Eventos	Resposta do Sistema
1. O caso de uso se inicia quando o Operador solicita um relatório de vendas antigas ao Sistema.	2. O Sistema responde solicitando ao Operador que informe pelo menos 1 de 3 atributos a serem pesquisados: "nome de cliente", "data de venda" ou "tipo de vidro".
3. O Operador preenche ao menos 1 dos campos solicitados e envia a solicitação ao sistema.	4. O Sistema filtra o banco de dados de venda com base no que foi informado pelo Operador, ignorando os campos em branco, e retorna com uma listagem das vendas correspondentes.
5. O Operador analisa os resultados e quando for conveniente informa ao sistema a conclusão da tarefa.	6. O sistema então encerra a operação.

### 6.1.9 Caso de Uso: Editar Banco de Clientes

<b>Atores:</b> Operador (Iniciador) <b>Propósito:</b> Encontrar e editar dados de clientes cadastrados <b>Descrição:</b> Existirá um banco de dados para cadastros de clientes. O operador fará uma pesquisa por clientes e então poderá editar os dados cadastrados, assim como adicionar informações, como por exemplo no campo Observações, que é destinado para comentários sobre como foi vender para esse cliente, assim como indicações sobre o endereço do cliente. <b>Tipo:</b> Secundário <b>Referência:</b> 1.1.2, 1.4.1, 1.4.2, 1.4.3, 1.4.4	
Típica Sequência de Eventos	Resposta do Sistema
1. O caso de uso se inicia quando o Operador solicita alteração nos dados cadastrais no banco de clientes da loja.	2. O Sistema responde solicitando ao Operador que informe pelo menos 1 de 4 atributos a serem pesquisados: "nome de cliente", "RG de cliente", "cpf de cliente", no caso de pessoa física e "cnpj de cliente", no caso de pessoa jurídica.
3. O Operador preenche ao menos 1 dos campos solicitados e envia a solicitação ao sistema.	4. O Sistema filtra o banco de dados de clientes com base no que foi informado pelo Operador, ignorando os campos em branco, e retorna com uma listagem dos clientes correspondentes.
	5. O Sistema solicita ao Operador sobre qual desses clientes retornados será feita uma edição.
6. O Operador seleciona um dos clientes retornados pela busca.	7. O Sistema retorna com o cadastro do cliente e solicita ao Operador que faça as atualizações necessárias.
8. O Operador modifica os campos que lhe são convenientes e solicita ao Sistema a atualização dos dados.	9. O Sistema procede com a atualização dos dados no banco de clientes, encerrando a operação em seguida.



#### 6.1.10 Caso de Uso: Gerenciar Finanças

<b>Atores:</b> Vidraceiro (Iniciador) <b>Propósito:</b> Consultar caixa <b>Descrição:</b> O vidraceiro quer acompanhar o andamento das finanças. Para tanto ele irá consultar o sistema por informações de ganhos e despesas, assim como solicitar ao sistema que calcule o balanço. <b>Tipo:</b> Primário <b>Referência:</b> 4.1.2, 5.1, 5.2, 5.3.1, 5.3.2, 5.4	
Típica Sequência de Eventos	Resposta do Sistema
1. Este caso de uso se inicia quando o vidraceiro deseja consultar o sistema sobre o andamento das finanças da sua vidraçaria. Para tanto ele irá logar no sistema com sua conta de administrador.	2. O sistema checará se os dados de login conferem. Caso positivo, irá oferecer ao vidraceiro se ele gostaria de checar sobre os ganhos, custos ou o balanço final do sistema.
3. O vidraceiro dirá qual a opção que atende sua necessidade.	4. O sistema irá requisitar informações sobre o período de tempo que o cliente gostaria de consultar as informações.
5. O vidraceiro informa o solicitado.	6. O sistema irá fazer uma busca no banco de dados de Histórico de Vendas e irá informar ao vidraceiro.
	7. O sistema abrirá opções para fazer uma nova consulta, mudar o período de tempo da consulta, ou encerrar a operação.
8. Se o vidraceiro decidir fazer nova consulta ou modificar o período de tempo o sistema irá voltar ao estado correspondente. Caso contrário, a consulta será encerrada e o sistema irá ser deslogar.	

### 6.1.11 Caso de Uso: Visualizar entregas pendentes

<p><b>Atores:</b> Operador (Iniciador)</p> <p><b>Propósito:</b> Ver listagem de entregas que não foram feitas ainda.</p> <p><b>Descrição:</b> Existe necessidade de acompanhar as entregas agendadas. O operador abrirá uma tela onde serão listadas as entregas do dia corrente, dos posteriores, assim como as atrasadas.</p> <p><b>Tipo:</b> Primário</p> <p><b>Referência:</b> 4.1.2, 4.5</p>	
Típica Sequência de Eventos	Resposta do Sistema
1. Tal caso de uso se inicia assim que o Operador seleciona a opção de abrir a listagem de entregas agendadas.	2. O sistema fará uma busca no histórico de vendas a fim de encontrar todas que não foram entregues ainda, ordenando-as pela data de entrega.
	3. O sistema abrirá uma tela onde listará esses resultados. As entregas atrasadas serão marcadas em vermelho, as do dia em amarelo e as futuras em verde. O sistema aguardará um comando para encerrar a tela, e então encerra-se a operação.

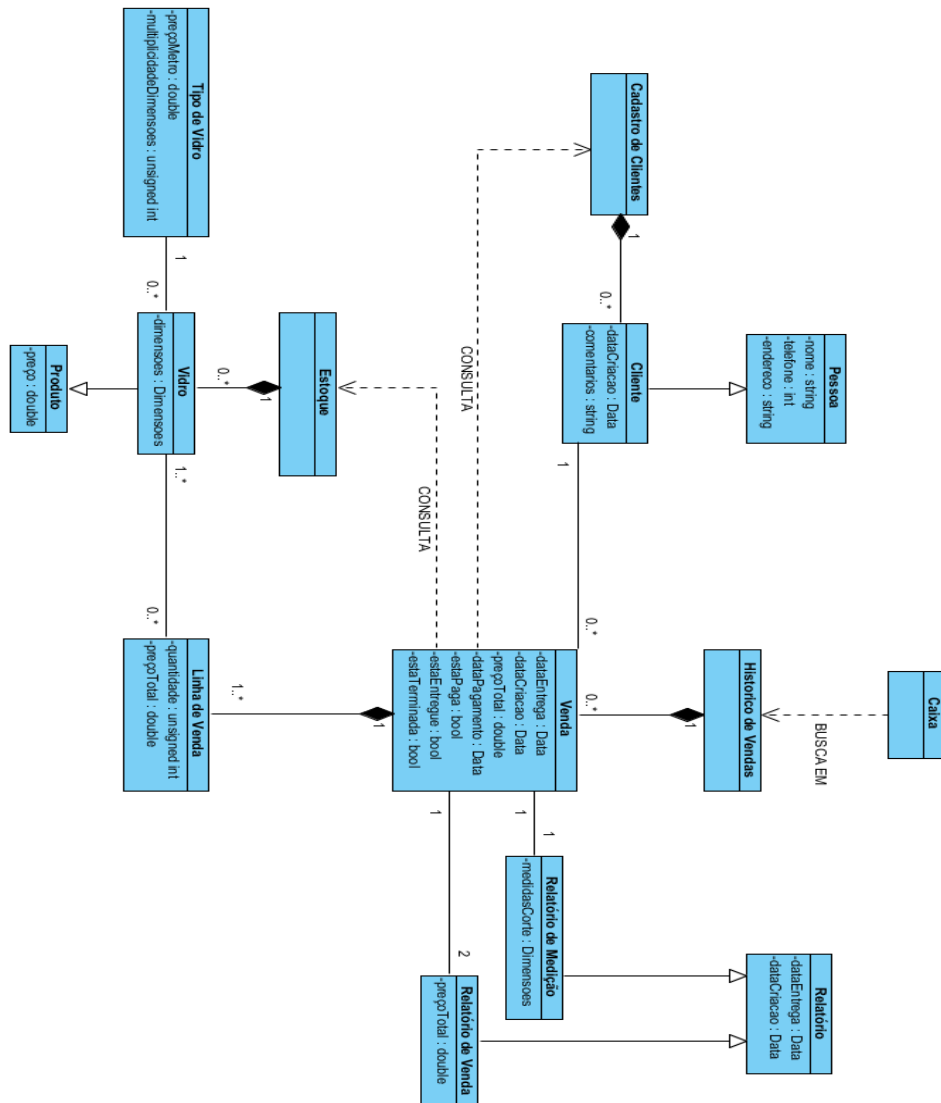
## 6.2 Diagrama de Casos de Uso

Abaixo o diagrama de casos de uso, que uma forma visual de identificar como cada ator se relaciona com as diferentes funções do sistema.



## 7 Modelo Conceitual

Após fazer a THF e os Casos de Uso, conseguimos distinguir quais são os conceitos que possivelmente tornar-se-ão as classes a serem usadas na implementação do sistema. Para tanto, criamos um Diagrama Conceitual, que indica tais conceitos bem como seus atributos e interrelações.

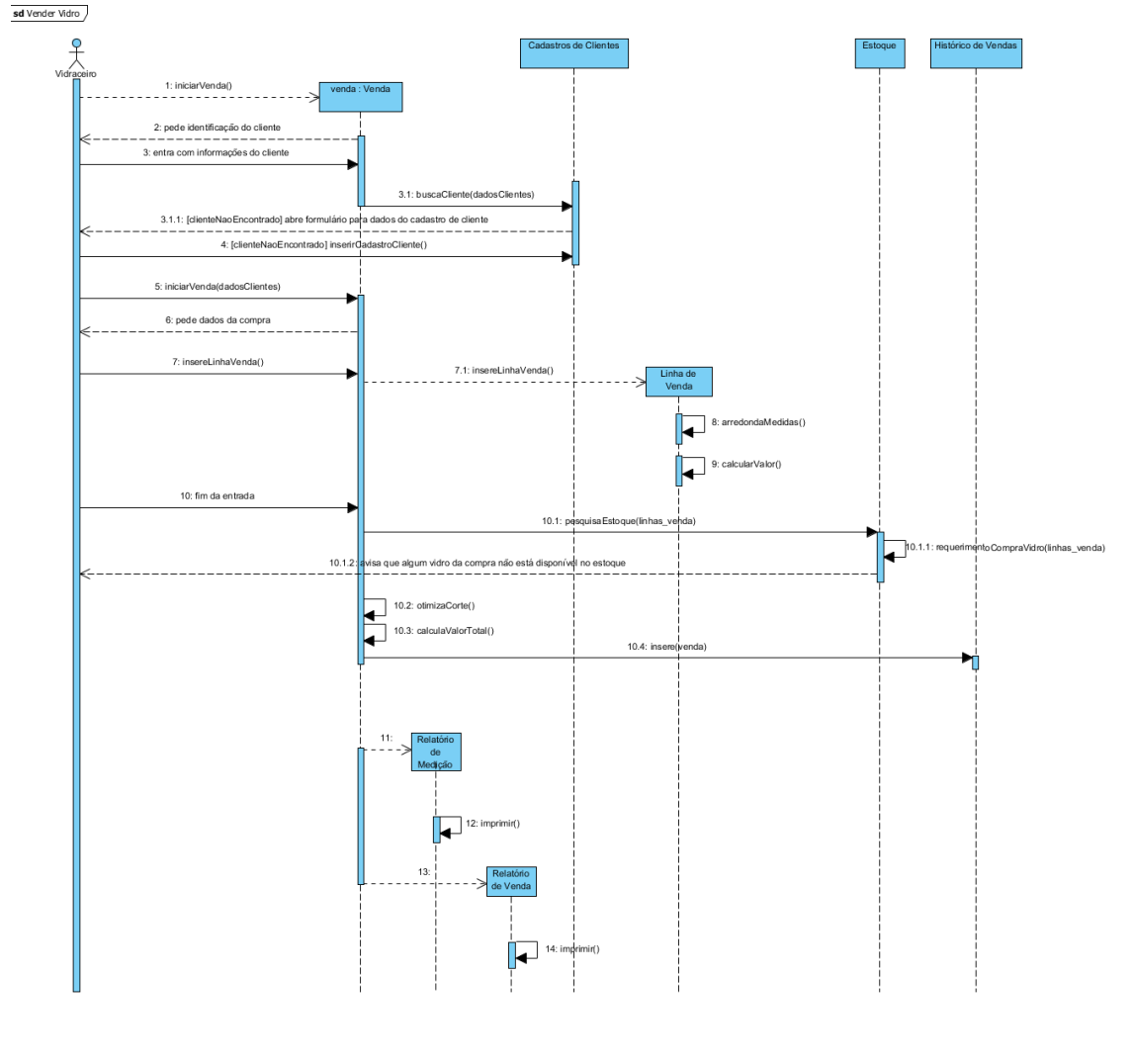


## 8 Diagramas de Interação

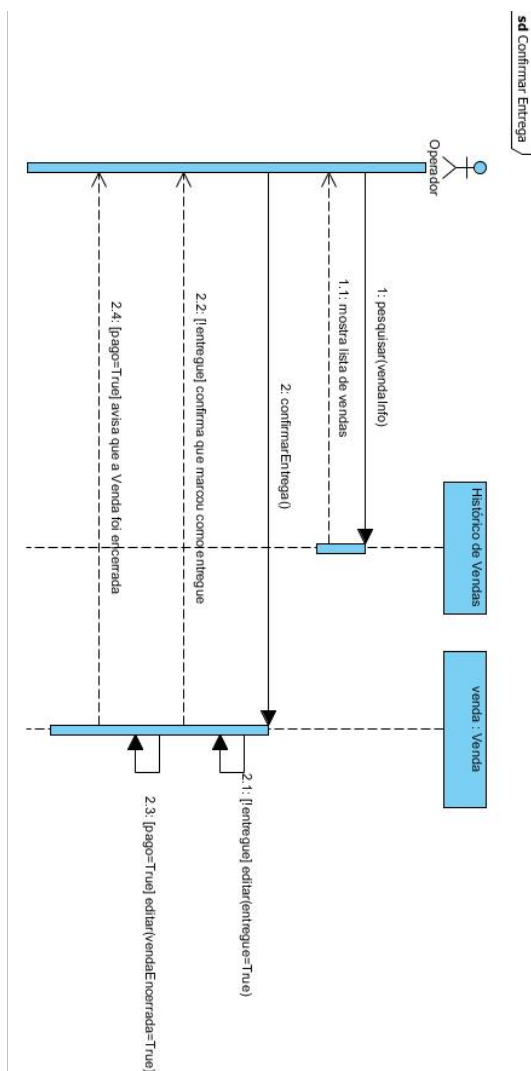
Como Diagrama de Interação, o grupo decidiu por utilizar os Diagramas de Sequência de Casos de Uso. Foi feita essa opção com base na intuitividade da representação das entidades e as mensagens trocadas entre elas.

Para cada caso de uso são explicitadas as funções realizadas pelos diferentes tipos de usuários e a consequência dessas pelo sistema, criando uma linguagem visual que facilmente descreve a complexidade e o nível de interação e propagação das mensagens pelas diferentes entidades e classes do sistema.

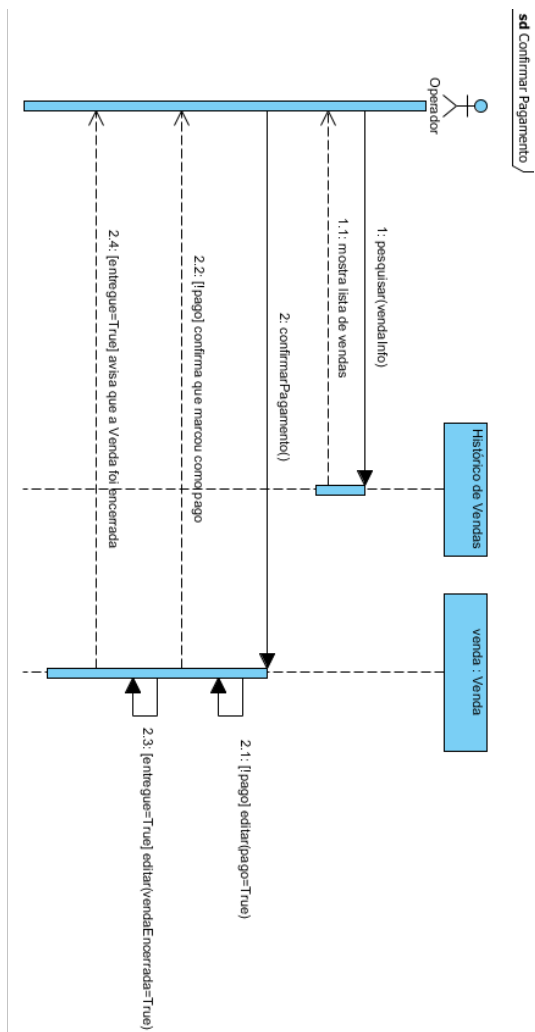
## 8.1 Vender Vidro



## 8.2 Confirmar Entrega

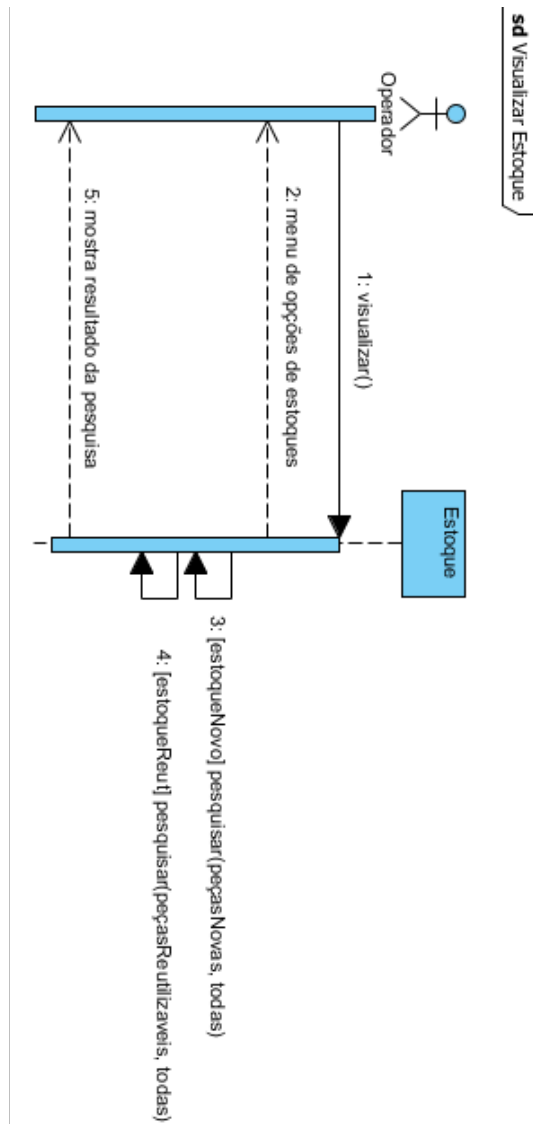


### 8.3 Confirmar Pagamento

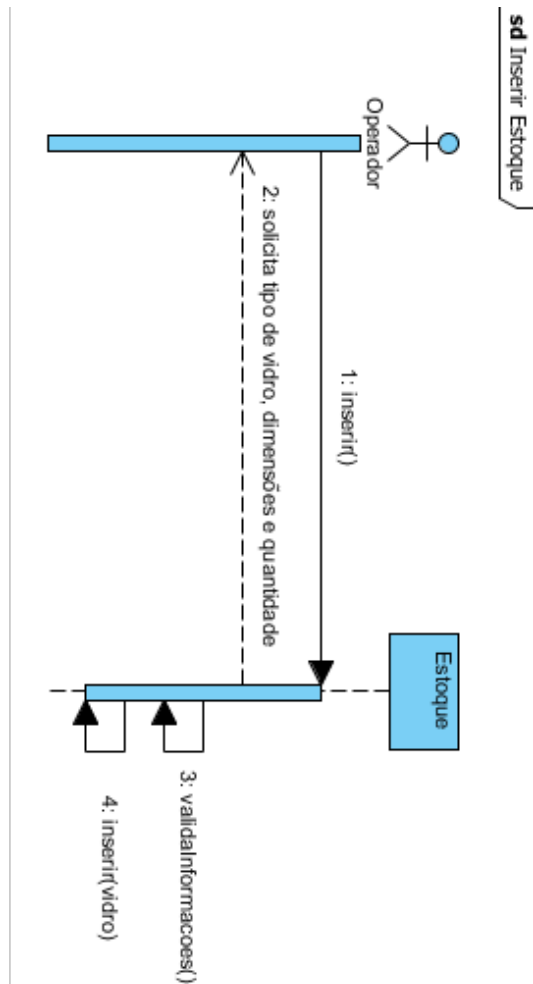




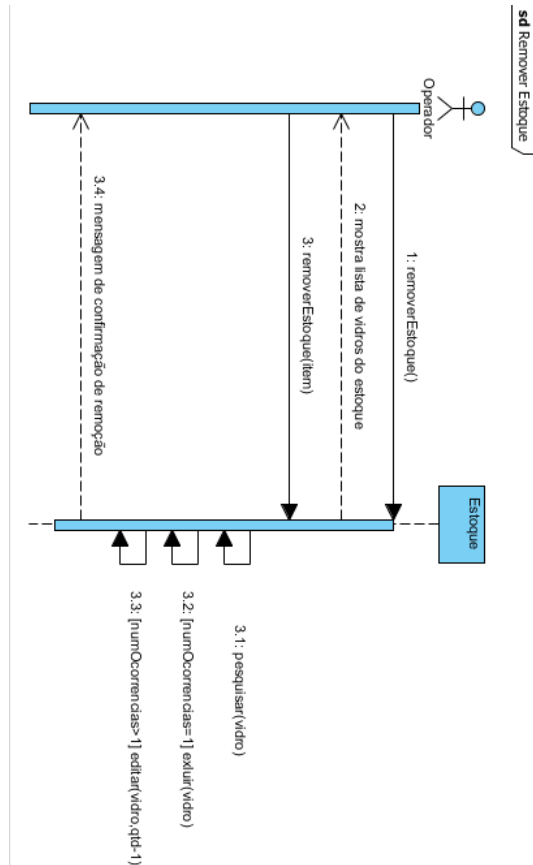
## 8.4 Visualizar Estoque



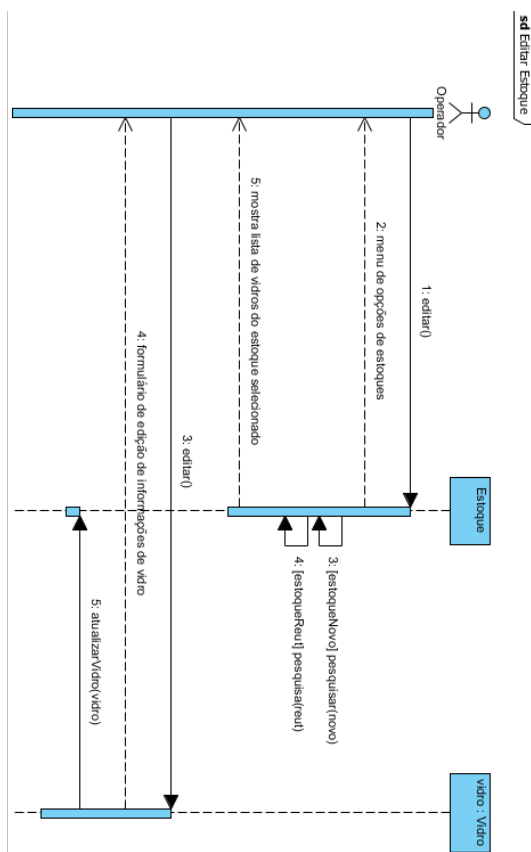
## 8.5 Inserir no Estoque



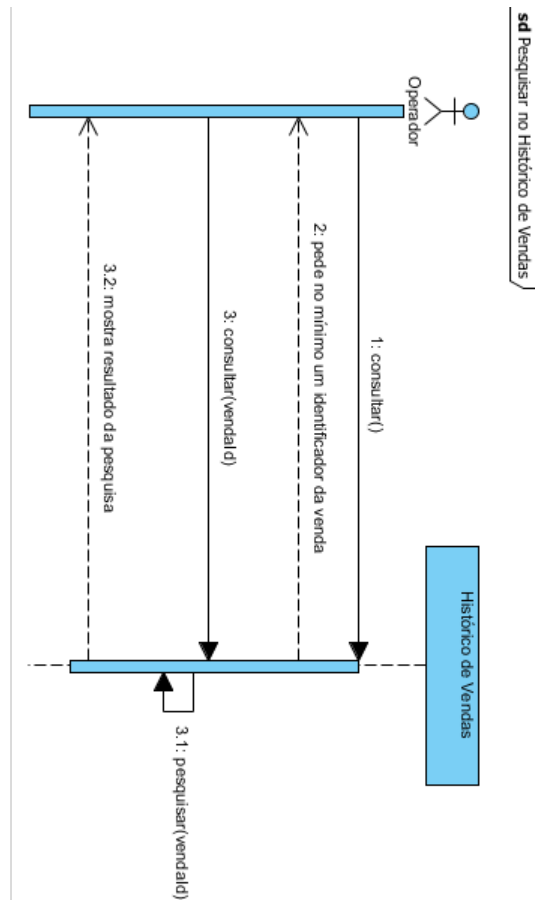
## 8.6 Remover do Estoque



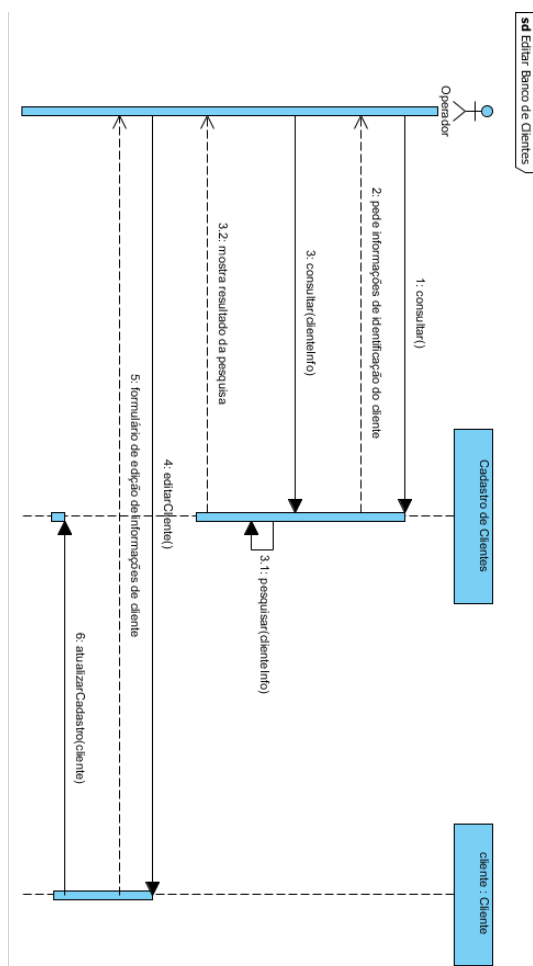
## 8.7 Editar o Estoque



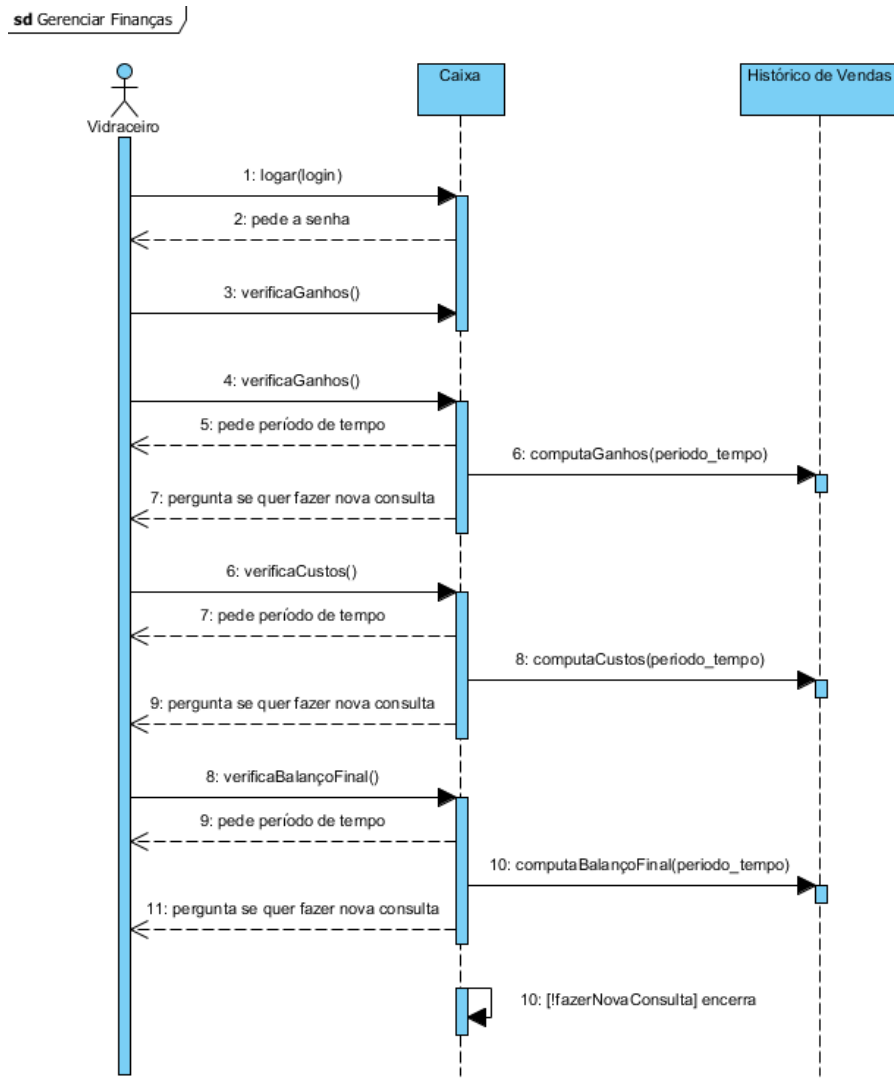
## 8.8 Pesquisar no Histórico de Vendas



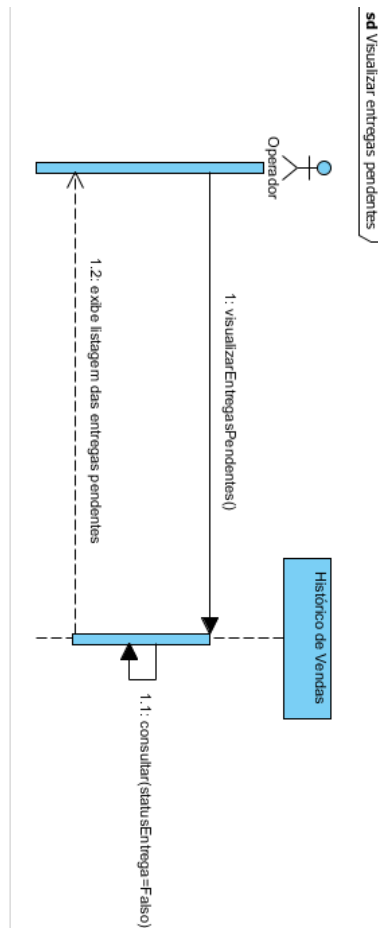
## 8.9 Editar Banco de Clientes



## 8.10 Gerenciar Finanças



## 8.11 Visualizar entregas pendentes

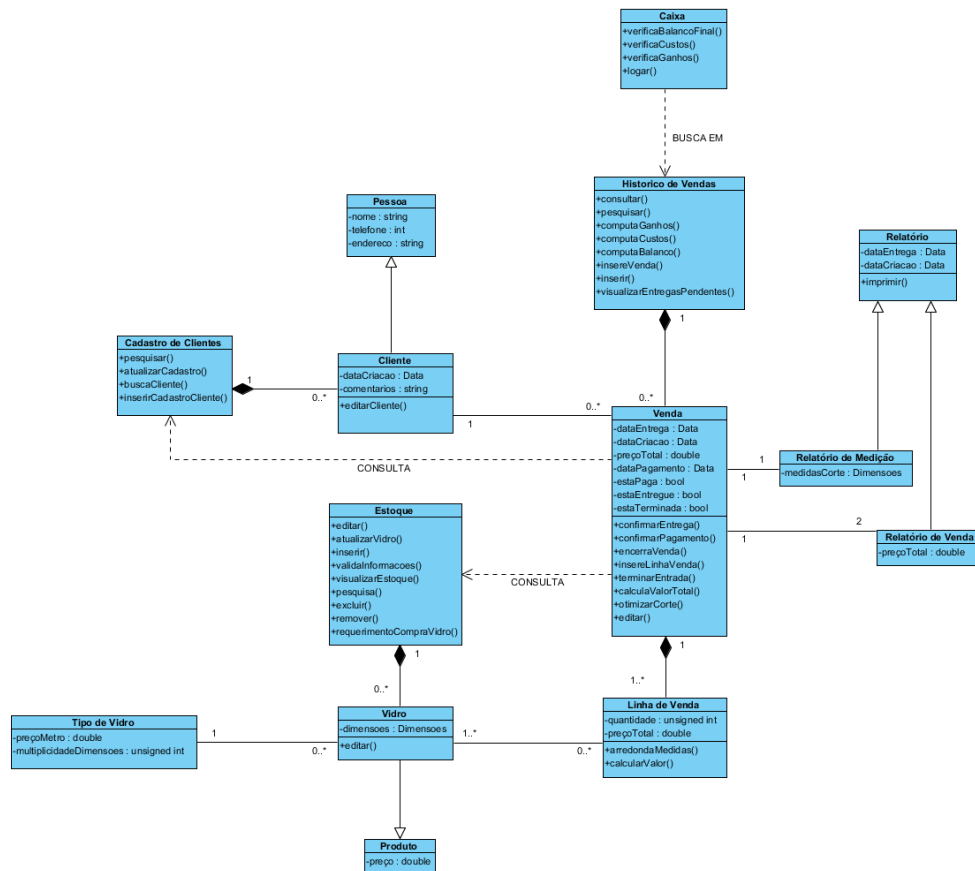




## 9 Diagrama de Classes

Em seguida, é apresentado o Diagrama de Classes do projeto final, que mostra todas as definições das classes do sistema. Os métodos de cada classe foram diretamente obtidos através da análise das mensagens dos Diagramas de Sequência de Casos de Uso.

Podemos notar a semelhança direta entre esse diagrama e o Diagrama Conceitual, mostrado anteriormente. Coincidentemente, o projeto corrente transpôs todos os conceitos para classes. Isso no entanto não necessariamente sempre acontece, pois como dito antes os conceitos são apenas candidatos à classe. Ocorre que no desenvolvimento da especificação muitas das entidades e relações foram sendo descobertas, havendo atualizações paralelas dos dois Diagramas.



## 10 Estimativas de Prazos e Custos

Como mencionado logo no início da monografia, no capítulo sobre as metodologias empregadas no projeto, a determinação dos custos de desenvolvimento do software é extremamente importante. Seguem os cálculos e fórmulas utilizados para a obtenção do valor final do sistema.

### 10.1 Cálculo do UAW

	Peso	Nº de Atores	Resultado
Ator Simples	1	0	0
Ator Médio	2	0	0
Ator Complexo	3	3	9
		TOTAL	9

### 10.2 Cálculo do UUCW

	Peso	Nº de Casos	Resultado
Simples	5	4	20
Médio	10	5	50
Complexo	15	2	30
		TOTAL	120

### 10.3 Cálculo do Tfactor

Fator	Requisito	Peso	Influência	Resultado
T1	Sistema Distribuido	2	1	0
T2	Tempo de Resposta	2	3	3
T3	Eficiência	1	3	1
T4	Processamento complexo	1	3	0
T5	Código reusável	1	0	3
T6	Facilidade de instalação	0.5	0	5
T7	Facilidade de uso	0.5	5	5
T8	Portabilidade	2	0	0
T9	Facilidade de mudança	1	3	4
T10	Concorrência	1	0	0
T11	Recursos de segurança	1	0	4
T12	Acessível por terceiros	1	0	0
T13	Requer treinamento especial	1	0	4
			Tfactor	29

### 10.4 Cálculo do Efactor

Fator	Descrição	Peso	Influência	Resultado
E1	Familiaridade com RUP ou outro processo formal	1.5	2	3
E2	Experiência com a aplicação em desenvolvimento	0.5	0	0
E3	Experiência em Orientação a Objetos	1	5	5
E4	Presença de analista experiente	0.5	3	1.5
E5	Motivação	1	3	3
E6	Requisitos estáveis	2	3	6
E7	Desenvolvedores em meio-expediente	-1	4	-4
E8	Linguagem de programação difícil	2	1	2
			Efactor	16.5

### 10.5 Cálculo de Pontos por Caso de Uso

A partir dos valores encontrados para UAW, UUCW, Tfactor e Efactor nos passos anteriores, podemos prosseguir com o restante dos cálculos que permitirão encontrar o custo do projeto especificado. Primeiramente, realizamos o cálculo do *UUCP* (Unadjusted Use Case Points).

$$UUCP = UAW + UUCW = 9 + 120 = 109$$

A seguir, calcularemos *TCF* (Technical Complexity Factor), a partir do *Tfactor*, e o *ECF* (Environmental Complexity Factor), a partir do *Efactor*.

$$TCF = 0.6 + (0.01 \times Tfactor) = 0.6 + (0.01 \times 29) = 0.89$$

$$ECF = 1.4 + (-0.03 \times Efactor) = 1.4 + (-0.03 \times 16,5) = 17.87$$

Finalmente, calculamos os pontos de casos de uso do projeto (UCP – Use Case Points).

$$UCP = UUCP \times TCF \times ECF = 109 \times 0.89 \times 17.87 = 177.76$$

## 10.6 Cálculo do Tempo de Trabalho

Para calcular o tempo de trabalho estimado usaremos como padrão a média de 20 horas por Ponto de Casos de Uso. Então, e sabendo que  $T_e$  = Tempo Estimado, temos que:

$$T_e = 177,76 * 20 = 3555,2 \text{ horas de trabalho}$$

## 10.7 Cálculo do Custo de Desenvolvimento

A estimativa de custo de desenvolvimento é obtida a partir da multiplicação do número de casos de uso estimados, pelo valor médio da hora de desenvolvimento. Para chegar ao valor do preço por hora de desenvolvimento, prevemos um programador trabalhando 8 horas diárias num ambiente com poucos desafios computacionais, o que nos permitira a contratação de uma equipe com baixo/médio nível técnico. O valor que estipulamos para o salário mensal de um integrante dessa equipe é de R\$1120,00, o que equivale a R\$7,50 a hora de trabalho.

Sabendo que  $H_t$  = horas de trabalho,  $M_{ht}$  = valor médio da hora de desenvolvimento, temos:

$$H_t \times M_{ht} = 3555,2 * 7,5 = R\$26.664,00$$

## 11 Ferramenta CASE

### 11.1 Visual Paradigm for UML 7.2 Community Edition

O Visual Paradigm é um software bastante rico. Ele é extremamente completo em questão de recursos e funcionalidades, permitindo a construção de todos diagramas desta monografia e muitos outros mais, os quais não chegamos a experimentar. Ocorre que, apesar de num primeiro momento a interface parecer poluída, ela é bastante bem construída levando em conta toda a gama de opções proporcionada, o que contribui para que o usuário acesse rapidamente apenas aquilo de que precisa, sem passar por telas intermediárias e outras burocracias. No princípio, aparecem várias caixas de diálogo questionando o usuário a respeito do comportamento esperado do software na ocorrência de certas ações. Por exemplo: ao deletar uma classe, ele apresentará opções a respeito da deleção das relações desta com outras existentes; mas logo podes marcar opções que determinam um padrão, para que na próxima vez o programa saiba o que fazer.

Sua interface é dinâmica, baseada na disponibilização de opções relativas à posição do mouse, ou seja, exatamente no que o usuário está trabalhando naquele momento. No caso de um Diagrama de Classes, ao repousares o mouse sobre um classe, pequenas caixas aparecerão ao redor daquela, mostrando diversas ações possíveis que podes fazer com ela; movendo o cursor na parte de mensagens, mais caixas aparecerão ao seu lado, com os tipos de mensagens que podem ser enviadas. Mas nem por isso ela é misteriosa ou confusa para quem ainda não sabe onde encontrar essas opções escondidas: tudo aparece num simples clique direito do mouse, onde terás todas as funcionalidades relativas à entidade em que clicaste, classificadas pelas categorias de ações que podes tomar naquele contexto.

Pra não dizer que tudo correu perfeitamente, há de se ressaltar que o software em seu maior esplendor é pago. A versão que utilizamos é a Community, o que significa que, adquirindo gratuitamente uma licença pelo site da Visual Paradigm, terás acesso a um subconjunto das funcionalidades do programa completo, além de que seu uso não é permitido para fins comerciais. Na realidade, do que verdadeiramente precisamos dele para realização do projeto desta monografia, a única opção que não estava inteiramente disponível era a de exportação dos diagramas em imagens (ele exportava, mas incluindo uma notável marca d'água). Um detalhe é a demora para iniciar o programa: levando em torno de 1 minuto para desaparecer a "splash screen" e a interface estar carregada. Mas depois disso o software é relativamente leve, não ocupando muito espaço na memória e sendo bastante rápido na responsividade.

Utilizamos o Visual Paradigm por indicação de colegas que já trabalharam com este software, e terminamos o projeto tentando imaginar o porquê de não ser o escolhido por todos estudantes da cadeira. Na realidade, o Visual Paradigm for UML 7.2 tem sua própria curva de aprendizado, não sendo no primeiro uso

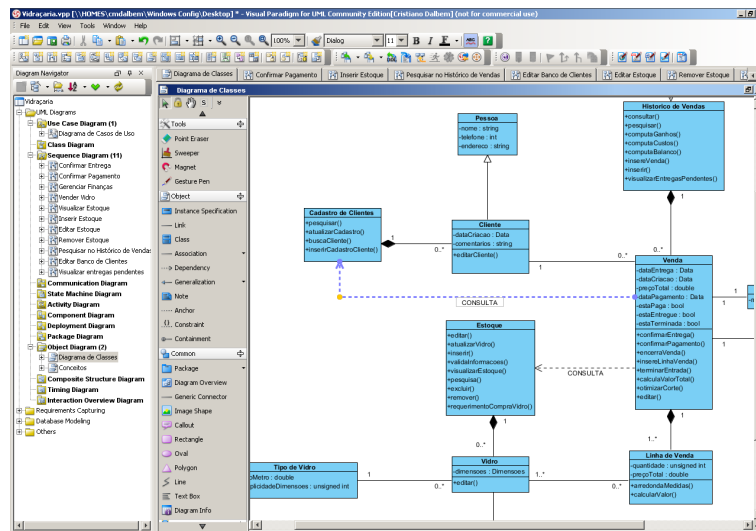


Figura 1: Interface da VP para UML - Visão Geral

que saíras fazendo tudo que queres com agilidade. Por outro lado é um programa que não necessita de tutoriais ou manuais para que tenhas a capacidade de operá-lo, considerando que um pouco de prática permite um trabalho eficiente e até, por que não dizer, prazeroso.

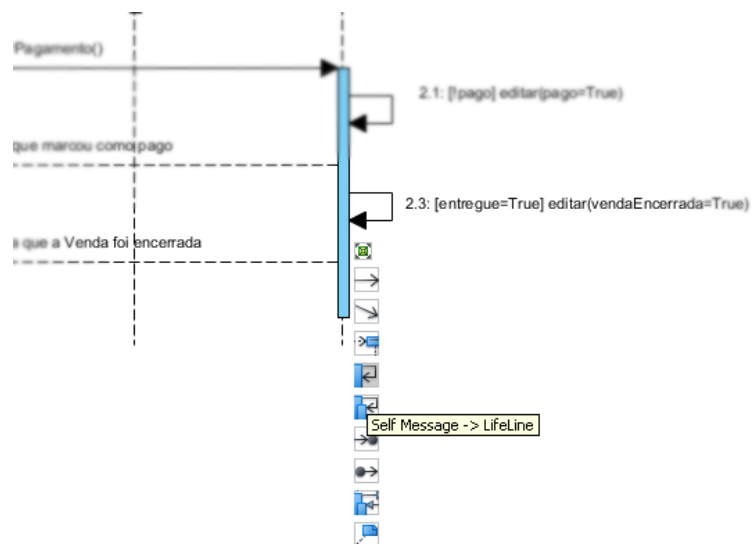


Figura 2: Interface da VP para UML - Exemplo de utilização de interface dinâmica

## 12 Conclusão

O trabalho desenvolvido na cadeira foi a primeira experiência do grupo com especificação formal de sistemas. Até então no curso só haviam sido trabalhados aspectos práticos da programação, em cadeiras como 'Algoritmos e Programação' e 'Estruturas de Dados', aspectos teóricos de computação nas demais cadeiras. Na cadeira de Técnicas de Construção de Programas vimos alguns aspectos de Engenharia de Software e boas práticas de programação, porém, como o nome sugere, mais voltados ao desenvolvimento em si (implementação).

O grupo reconhece a importância de, para sistemas muito complexos e potencialmente críticos, uma especificação formal detalhada e precisa, que não só facilita o desenvolvimento futuramente como também auxilia na detecção e conserto de erros.

O trabalho permitiu ao grupo também vivenciar da sofrível relação analista-cliente, tão mistificada no meio da Engenharia de Software, mas ao mesmo tempo bastante real.

Em suma, o grupo considerou o trabalho uma ótima ferramenta para o ensino da Engenharia de Software, tendo em vista que apenas aulas teóricas (depois do almoço ainda) simplesmente mostrariam os conceitos sem nenhuma conexão direta com a realidade, apesar da experiência do professor com trabalhos relacionados ao assunto, inclusive um projeto atual, mas que mesmo assim não são suficientes para mostrar o que realmente é a Engenharia de Software.

## 13 Referências

- UML, business process and database design tool for software development projects - Visual Paradigm. [Online] <http://www.visual-paradigm.com/product/vpuml>
- Eng. Software N. [Online] Prof. Dr. Sérgio Felipe Zirbes <http://inf.ufrgs.br/zirbesEngSoftwareN.htm>
- UML na Wikipedia [Online] [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)

## 14 Apêndice - Atas de Reunião

### 14.1 Primeira Reunião

**Participantes :**

- Cristiano
- Fábio



- Juliano
- Rudinei

**Local:** Casa do Cristiano, Porto Alegre, RS

**Objetivo da Reunião:** Eliciação de Requisitos

**Pauta :**

- Apresentação dos envolvidos.
- Apresentação da vidraçaria e breve contextualização no mercado.
- Descrição do funcionamento da vidraçaria.

**Observações :**

- O Sr. Rudinei foi apresentado ao Fábio e ao Juliano, e vice-versa.
- O cliente explicou, brevemente, como funciona o mercado de venda de vidros, assim como o funcionamento do trabalho na sua vidraçaria.
- Os analistas explicaram rapidamente como se daria o trabalho, indicando brevemente sobre os diagramas que seriam feitos, e qual seria a participação do cliente na validação dos mesmos.
- Após ambos os lados entendidos, os analistas decidiram que iriam fazer uma especificação geral do sistema e em próxima reunião seriam discutidos detalhes mais aprofundados.

**Plano de Ação :**

- Como combinado em reunião, os analistas iriam preparar uma visão geral do sistema, na forma de uma Tabela Hierárquica de Funções, enquanto o cliente iria pensar um pouco mais sobre o que ele precisaria.
- Combinou-se que na próxima reunião seria validado ou não a THF criada, e seriam feitas melhorias sobre a mesma.

## 14.2 Segunda Reunião

**Participantes :**

- Cristiano
- Fábio
- Juliano
- Rudinei

**Local:** Casa do Cristiano, Porto Alegre, RS

**Objetivo da Reunião:** Validação da THF

**Pauta :**

- Apresentação da THF.
- Correção da THF.
- Análise mais aprofundada do negócio (venda de vidros).
- Comparação com softwares ditos profissionais.

**Observações :**

- Explicamos o que era e apresentamos a THF ao cliente, que logo encontrou várias funções que seriam futuramente cortadas da tabela.
- O cliente explicou mais à fundo como funcionava a vidraçaria, e quais seriam seus principais interesses em informatizá-la.
- O cliente comentou sobre um software profissional (e comercial) para gerência de vidraçarias, mas que era voltado somente à grandes empresas.
- Durante a explicação mais aprofundada dos desejos do cliente, ficou mais claro o que realmente tinha de ser feito, e o que deveria ser alterado na THF.

**Plano de Ação :**

- Os analistas ficaram encarregados de atualizar a THF, assim como começar a descrever os Casos de Uso para o sistema, levando em conta as novas exigências e descartando o que estava incorreto.

### 14.3 Terceira Reunião

**Participantes :**

- Cristiano
- Fábio
- Juliano
- Rudinei

**Local:** Casa do Cristiano, Porto Alegre, RS

**Objetivo da Reunião:** Análise dos Casos de Uso

**Pauta :**

- Apresentação das descrições de alguns casos de uso.
- Apresentação da nova versão da THF.

**Observações :**

- O cliente ficou satisfeito com a nova THF, a qual refletia as funcionalidades que o sistema teria. Ele queria algo simples, e que resolvesse, inicialmente, apenas alguns problemas que ele apontou em reuniões anteriores, nós demos isso à ele.
- Os analistas explicaram rapidamente o objetivo dos Casos de Uso, e apresentaram os principais já feitos pelo grupo.
- Durante a análise dos Casos de Uso, foi discutido quem operaria o sistema, e eventualmente chegou-se ao velho problema da interface usuário-sistema. Como o cliente era bastante leigo no quesito computador, a discussão ficou complicada, e chegou-se ao consenso que era preciso algum funcionário minimamente familiarizado com informática participando da reunião, para dar opiniões.
- Foi agendada outra reunião, da qual deveriam participar mais pessoas.

**Plano de Ação :**

- Ficou acordado que participaria da próxima reunião a esposa do cliente, Sra. Graciê Hoff, que tinha algum contato com informática e poderia nos ajudar na formulação de como se daria a interação entre o operador e o sistema.
- Além disso o grupo iria terminar de descrever os casos de uso, para posterior análise e aprovação.

## **14.4 Quarta Reunião**

**Participantes :**

- Cristiano
- Fábio
- Juliano
- Rudinei
- Graciê

**Local:** Casa do Cristiano, Porto Alegre, RS

**Objetivo da Reunião:** Reanálise dos Casos de Uso e análise dos Diagramas de Sequência

**Pauta :**

- Validação dos casos de uso.
- Apresentação dos diagramas de sequência.

**Observações :**

- Agora com a Sra. Graciê, esposa do cliente e sócia da vidraçaria, presente, discutimos novamente os casos de uso, de um ponto de vista mais da interação entre o operador e o sistema, e chegamos a alguns consensos quanto ao assunto, praticamente encerrando o assunto casos de uso.
- Como o grupo precisava entregar os diagramas de sequência, eles foram feitos com antecedência, baseados no que se tinha até o momento, e discutidos muito rapidamente. Esses diagramas teriam de ser revisados juntamente com os casos de uso.

**Plano de Ação :**

- Finalizar os casos de uso, e concertar os diagramas de sequência para refletir tais alterações.
- Fazer o Diagrama de Classes.