

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

MATHEUS RIBEIRO GONZATTI

**APLICATIVO PARA GERENCIAMENTO DE OPERAÇÕES DE COMÉRCIO
EXTERIOR**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2015**

MATHEUS RIBEIRO GONZATTI

**APLICATIVO PARA GERENCIAMENTO DE OPERAÇÕES DE COMÉRCIO
EXTERIOR**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

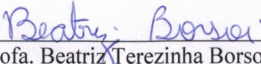
Orientadora: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2015**

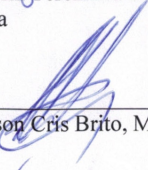
ATA Nº: 270

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO **MATHEUS RIBEIRO GONZATTI**.

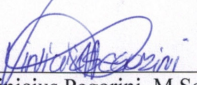
Às 13:30 hrs do dia 24 de novembro de 2015, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Robison Cris Brito (Convidado) e Vinicius Pegorini (Convidado), para avaliar o Trabalho de Diplomação do aluno Matheus Ribeiro Gonzatti, matrícula 833908, sob o título **Aplicativo para gerenciamento de operações de comércio exterior**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 14:10 hrs foi encerrada a sessão.



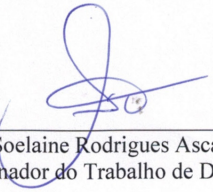
Prof. Beatriz Terezinha Borsoi, Dr.
Orientadora



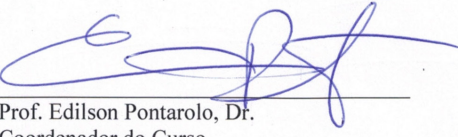
Prof. Robison Cris Brito, M.Sc.
Convidado



Prof. Vinicius Pegorini, M.Sc.
Convidado



Prof. Soelaine Rodrigues Ascari, M. Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

RESUMO

GONZATTI, Matheus Ribeiro. Aplicativo para gerenciamento de operações de comércio exterior. 2015. 50f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

A atividade de exportação de qualquer tipo de mercadoria possui um processo que é constituído de várias etapas até que o produto chegue ao seu destino. A principal dessas etapas é a realização do pedido. Nessa etapa, as empresas contam com vendedores que atuam em contato direto com os clientes, localizados em países diversos, realizando a negociação de quantidades, valores e prazos dos produtos a serem exportados. A partir desse momento, as informações precisam ser registradas para haver o controle da produção, do pagamento e do envio desses pedidos. Não existindo essa possibilidade de registros das informações, é difícil realizar o controle sem que dados sejam perdidos, levando a perda de qualidade no serviço e de tempo. Dessa forma, todo o processo é afetado, dificultando o crescimento do mercado de exportação. A partir disso, foi identificada a possibilidade de implementar um sistema web que permita que os processos envolvidos no comércio de exportação de gelatinas, um mercado específico, mas com características comuns a exportação de qualquer bem, sejam realizados e armazenados em um único local. O sistema foi desenvolvido para plataforma web, utilizando a tecnologia .NET e Microsoft SQL Serve para o banco de dados.

Palavras-chave: Aplicação web. .NET. Comércio exterior. Exportação de gelatinas.

ABSTRACT

GONZATTI, Matheus Ribeiro. Management of foreign trade operations application. 2015. 450f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

The export activity of any kind of merchandise has a process that consists of several steps until the product reaches its destination. The main step is the realization of the purchase order. At this step, the companies rely on sellers that work directly with the customer, located in several countries, negotiating quantity, prices, deadlines and others, about the product to be ordered. From this moment, those information need to be registered to production control, payment and shipment of this orders. If there is no possibility of registering those information, it's difficult to have control without losing data, leading to lack of quality in service and time. Thus, all process is affected, making it difficult for the export market growth. From that, it was identified the possibility of implement a web system that allow all the processes involved in the export commerce of gelatin, a specific market, but with common features to the exportation of any good, be made and stored in one location. The system was developed using the .NET technology and Microsoft SQL Server for database.

Palavras-chave: Application web. .NET. International trade. Export commerce of gelatin.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso.....	22
Figura 2 – Diagrama de entidades e relacionamentos do banco de dados	26
Figura 3 – Tela principal do aplicativo	31
Figura 4 – Formulário de cadastro de clientes.....	31
Figura 5 – Formulário de cadastro de clientes.....	32
Figura 6 – Formulário de cadastro de clientes.....	32
Figura 7 – Listagem de clientes.....	33
Figura 8 – Cadastro de pedidos	33
Figura 9 – Cadastro de pedidos	34
Figura 10 – Cadastro de pedidos	34
Figura 11 – Filtro do relatório de pedidos.....	35
Figura 12 – Relatório de pedidos.....	35

LISTA DE QUADROS

Quadro 1 – Tecnologias e ferramentas utilizadas na modelagem e na implementação	17
Quadro 2 – Requisitos funcionais.....	20
Quadro 3 – Requisitos não funcionais	21
Quadro 4 – Funcionalidade incluir dos casos de uso de cadastro	23
Quadro 5 – Funcionalidade alterar dos casos de uso de cadastro.....	23
Quadro 6 – Funcionalidade excluir dos casos de uso de cadastro.....	24
Quadro 7 – Funcionalidade consultar dos casos de uso de cadastro	25
Quadro 8 – Caso de uso de imprimir pedidos	25
Quadro 9 – Caso de uso de gerar relatórios	25
Quadro 10 – Campos da tabela Funcionarios.....	27
Quadro 11 – Campos da tabela Cargos	27
Quadro 12 – Campos da tabela Usuarios	27
Quadro 13 – Campos da tabela GruposUsuarios.....	27
Quadro 14 – Campos da tabela GrupoUsuariosPermissoes	27
Quadro 15 – Campos da tabela Permissoes	28
Quadro 16 – Campos da tabela Clientes	28
Quadro 17 – Campos da tabela ClienteEspecificacoes.....	28
Quadro 18 – Campos da tabela Especificacoes.....	28
Quadro 19 – Campos da tabela Pedidos.....	29
Quadro 20 – Campos da tabela Booking.....	29
Quadro 21 – Campos da tabela FormasPagamento	29
Quadro 22 – Campos da tabela PedidoComplemento	30
Quadro 23 – Campos da tabela Complemento	30
Quadro 24 – Campos da tabela Paises	30

LISTAGEM DE CÓDIGO

Listagem 1 – Site.MasterPage e ClienteCadastro.aspx.vb	36
Listagem 2 – ClienteCadastro.aspx.vb.....	37
Listagem 3 – btnSalvar_click() da classe cliente.....	38
Listagem 4 – Classe cliente.vb	39
Listagem 5 – Classe pedido.vb	40
Listagem 6 – btnSalvar_click() da classe pedido	42
Listagem 7 – Método CarregaPedido da classe pedido.....	43
Listagem 8 – grd_RowCommand() da classe pedido	44
Listagem 9 – XML do DataSet para o relatório de pedidos.....	46
Listagem 10 – Page_Load do relatório de pedidos.....	47

LISTA DE SIGLAS

CRUD	<i>Create Retrieve Update and Delete</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
IIS	<i>Internet Information Services</i>
RF	<i>Requisito Funcional</i>
RNF	<i>Requisito Não Funcional</i>
SQL	<i>Structured Query Language</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 CONSIDERAÇÕES INICIAIS	10
1.2 OBJETIVOS.....	11
1.2.1 Objetivo Geral.....	11
1.2.2 Objetivos Específicos.....	11
1.3 JUSTIFICATIVA	12
1.4 ESTRUTURA DO TRABALHO	12
2 REFERENCIAL TEÓRICO	14
2.1 CADEIAS DE SUPRIMENTOS	14
2.2 EXPORTAÇÃO.....	15
3 MATERIAIS E MÉTODO	17
3.1 MATERIAIS.....	17
3.2 MÉTODO	18
4 RESULTADO	19
4.1 ESCOPO DO SISTEMA.....	19
4.2 MODELAGEM DO SISTEMA.....	20
4.4 IMPLEMENTAÇÃO DO SISTEMA	36
5 CONCLUSÃO.....	48
REFERÊNCIAS.....	49

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais que definem o contexto no qual se insere o aplicativo desenvolvido como resultado principal da realização deste trabalho. Neste capítulo também são apresentados os objetivos e a justificativa do trabalho. E por fim está a organização do texto por meio da apresentação dos seus capítulos subsequentes.

1.1 CONSIDERAÇÕES INICIAIS

O comércio exterior, de maneira bastante simplificada, possibilita que países mantenham entre si relações comerciais de importação e exportação. Essa forma de comércio pode ocorrer, também, entre os integrantes de uma cadeia de valor no comércio de bens e serviços, como, por exemplo, o fornecimento de matéria-prima, a execução de determinadas etapas de processamento do produto e a realização de serviços específicos de distribuição do produto final. A globalização amplia e promove o comércio exterior e o uso de recursos tecnológicos de informação e comunicação, como a Internet, facilitando as trocas comerciais entre países distintos.

Como as exportações geralmente ocorrem por meio de grandes volumes de cargas e podem estar relacionadas a grandes distâncias é comum que estejam associadas às estruturas portuárias, embora o transporte possa ocorrer por outros meios, como ferroviário, aéreo e rodoviário. Contudo, nas exportações brasileiras os portos se destacam. Amato (2012) ressalta que os portos escoam 95% das exportações de produtos brasileiros.

O comércio exterior pelo grande volume de carga geralmente comercializado e transportado e pelos aspectos de legislação e por geralmente ocorrer entre países distintos, pode envolver tarefas e operações complexas. O uso de sistemas de informação, suportado por tecnologias de comunicação, auxilia nessas operações pelo gerenciamento de dados e facilidade de troca de informações entre os envolvidos nas operações comerciais. Os parceiros de negócio podem ser elos da cadeia de produção ou fornecedores e compradores (exportadores e importadores).

Considerando esse cenário verificou-se a possibilidade e mesmo a importância de desenvolver um sistema para informatizar as atividades envolvidas nas operações de comércio exterior para o mercado de gelatinas e auxiliar no controle do recebimento dos pedidos realizados pelos clientes e do encaminhamento (entrega) desses pedidos. Embora o sistema

tenha como foco um segmento específico de mercado, que é o de gelatinas, poderá, com os devidos ajustes nas especificidades do produto e do processo, ser utilizado para o gerenciamento de operações de comércio exterior de outros produtos. Dependendo do produto comercializado, ajustes para adequar-se as especificidades dos produtos e as regras comerciais e de legislação específicas podem ser necessários.

A modelagem do sistema, assim como a implementação das de operações básicas de cadastro visando definir os aspectos das tecnologias que utilizadas, foram realizados como trabalho de estágio supervisionado do autor deste trabalho. A modelagem é apresentada neste texto para facilitar o entendimento da solução proposta, embora a ênfase deste trabalho seja a implementação do sistema.

1.2 OBJETIVOS

A seguir estão o objetivo geral e os objetivos específicos definidos para este trabalho.

1.2.1 Objetivo Geral

Implementar um sistema para comércio de exportação de gelatinas, com controle de pedidos e gerenciamento financeiro.

1.2.2 Objetivos Específicos

- Implementar uma proposta de solução para um sistema que vise facilitar a realização de tarefas cotidianas no processo de exportação de gelatinas, como realização do pedido e agendamento de envios.
- Prover uma forma de facilitar o controle das operações de vendas de comércio exterior para uma empresa exportadora.
- Facilitar o gerenciamento das atividades realizadas no comércio de exportação de gelatinas.

1.3 JUSTIFICATIVA

A gelatina é uma proteína derivada da hidrólise parcial do colágeno animal, contido em ossos e peles, principalmente de suínos e bovinos (YANG et al., 2007). Almeida, Vanelle e Santana (2012, p. 9), ressaltam que a grande maioria das gelatinas comerciais é derivada de mamíferos, obtidas, principalmente, a partir de pele de suíno e couro bovino, mas por muitas razões socioculturais cresce a exigência de fontes alternativas. Entre essas razões estão as restrições religiosas como a do judaísmo e islamismo e doenças relacionadas aos bovinos (YAN *et al*, 2007; GIMÉNEZ *et al*, 2009) especialmente desde o surto da doença da vaca louca (CHO *et al*, 2005). Entre as fontes alternativas estão a gelatina de origem marinha obtida por meio da pele ossos e barbatanas de peixe (HAUG *et al*, 2004) e de pele e pés de frango (ALVES, PRUDÊNCIO-FERREIRA, 2002).

Em 2005, o Brasil possuía o segundo maior rebanho bovino do mundo e era o maior exportador da carne e, além disso, era responsável por cerca de 11,3% da gelatina produzida no mundo (ISAACK, 2005). Do total de gelatina produzido, mais de 74% era exportado. Em 2008, os dados de produção de gelatina cresceram, segundo a Scot Consultoria (2008) nesse ano o Brasil era responsável por 20% de toda gelatina produzida no planeta. As exportações eram de 80% da produção, tornando-o o maior exportador de gelatina do mundo. Em 2012 Brasil figurava como o maior exportador mundial de gelatina (15% das exportações mundiais), de acordo com dados das Nações Unidas – Comtrade de 2012, citados em Bain e Company e Gas Energy (2014).

Esses dados indicam que o mercado de exportação de gelatinas é muito promissor ao Brasil e com o aumento da demanda nas vendas é necessário informatizar o processo, garantindo controle, precisão e o armazenamento das informações geradas e manipuladas. Sustenta-se, assim, a importância do auxílio de sistemas de informação para o gerenciamento das operações de exportação desse mercado. É nesse contexto que se insere a proposta do sistema desenvolvido como resultado da realização deste trabalho.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos. Este é o primeiro capítulo é apresenta as considerações iniciais com o contexto do sistema a ser desenvolvido, os seus objetivos e a justificativa. O Capítulo 2 apresenta o referencial teórico centrado em comércio internacional.

No Capítulo 3 estão as ferramentas e as tecnologias utilizada na modelagem do sistema e que serão utilizadas na implementação subsequente do sistema. No Capítulo 4 é apresentado o resultado da realização do trabalho, ou seja, o desenvolvimento do sistema. Por fim estão as considerações finais seguidas pelas referências

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial do trabalho que é sobre comércio exterior. O sistema modelado como resultado deste trabalho é para o gerenciamento de operações de comércio exterior.

2.1 CADEIAS DE SUPRIMENTOS

Para Oliveira e Longo (2008) uma cadeia de suprimentos (*supply chain*) é um conjunto de instalações dispersas geograficamente interagindo entre si. Esses autores citam como exemplos dessas instalações: fornecedores de matéria-prima, plantas de transformação de produtos como as indústrias, centros de distribuição, varejistas, estoque em trânsito, produtos intermediários e produtos acabados (YIN, 1991, *apud* SOUZA; CARVALHO; LIBOREIRO, 2006).

De maneira simplificada, uma cadeia de suprimentos é composta, basicamente, por fornecedores, produtores, distribuidores e vendedores. Nessa cadeia estão todos os envolvidos no ciclo de produção, processamento (industrialização) e comercialização do produto. O deslocamento do produto do produtor da matéria-prima para o ponto de venda para o consumidor, inclui o transporte desse produto, o que pode ocorrer em longas distâncias, especialmente quando envolve comércio internacional.

O gerenciamento da cadeia de suprimentos é composto por um conjunto de métodos que visam proporcionar a integração e a gestão melhoradas de todos os parâmetros da rede, incluindo: transportes, estoques, receitas, despesas, compras e vendas, entre outros (COELHO, 2010). Para esse autor, esses parâmetros estão presentes em todos os elos da cadeia, do fornecedor ao cliente, além da própria empresa.

É natural que em uma cadeia de suprimentos sejam necessárias negociações entre quem demanda produtos e/ou serviços e quem os fornece (oferta) (FENG, YANLI, 2009). Esse tipo de negociação em cadeias de suprimento é, geralmente, denominado de negociações comerciais internacionais (COLLINS, 2008, MOODIE; BOBROWSHKI, 2008, COHEN, STATHIS, 2007).

Essas negociações são necessárias porque nessas cadeias, as decisões tomadas por uma empresa podem ser conflitantes com as outras partes envolvidas. Isso porque cada empresa possui o seu modelo de negócio e seus interesses, além de que, todas querem minimizar

custos e maximizar lucros. Para resolver esses conflitos é que existem as negociações entre as organizações que são os parceiros de negócio que compõem uma determinada cadeia de suprimentos.

2.2 EXPORTAÇÃO

A exportação é, basicamente, a saída da mercadoria do território aduaneiro que é decorrente de contrato de compra e venda internacional (MINISTÉRIO..., 2015). Esse contrato pode ou não resultar na entrada de divisas para o país que está exportando.

As exportações desempenham um papel bastante relevante do desenvolvimento de países, assim, é comum que as teorias de crescimento regional centrem suas análises nas relações da região com o exterior no que se refere a sua capacidade de exportar (MUNDURUCA, 2010). Para Silva (2004) a teoria de base da exportação, que se fundamenta na capacidade de exportar de uma região e da importância que isso tem para o desenvolvimento da região, define que o crescimento das exportações, como atividade básica, gera um efeito multiplicador e de aceleração de crescimento do mercado interno, criando demanda por serviços, como, por exemplo, transportes, comunicações e financiamentos.

Assim, o aumento de exportações é uma alternativa para a elevação da demanda e isso pode contribuir para minimizar crises e favorecer a absorção pelo mercado global da oferta de produtos locais (MUNDURUCA, 2010). Souza (2002) destaca que, uma região ou país que produza bens que possam ser mais amplamente utilizados e consumidos em termos globais e ao mesmo tempo suportar o aumento da complexidade interna de sua economia, ampliar e melhorar a sua capacidade industrial pode conquistar novos mercados. Essa ampliação de mercados gerará economia de escala que associada a eficiência produtiva, pode reduzir os custos médios de produção, estimulando a acúmulo de capital e a melhoria dos níveis de emprego e de qualificação.

Souza (2002) ressalta que os principais argumentos em relação às vantagens de um modelo voltado ao crescimento das exportações são:

a) Complemento ao mercado interno - as exportações podem fornecer uma alternativa para os estoques quando o mercado interno estiver com capacidade ociosa ocasionada pela saturação da demanda interna.

b) Possibilidade de geração de economias de escala - o aumento das exportações dilui custos fixos, reduz os custos médios e aumenta a margem de lucro;

c) Melhoria da eficiência produtiva interna - a concorrência conduz a uma maior especialização e aumento de padrões de eficiência e competitividade;

d) Melhor aproveitamento dos recursos disponíveis – decorrente da redução da ociosidade produtiva da economia e aumento o emprego dos recursos disponíveis;

e) Geração de processo de interdependência tecnológica e econômica – que ocorre pela tendência de demanda de serviços e compras de insumos e de bens de capitais produzidos internamente.

A teoria da base exportadora define que a expansão das exportações exerce efeito multiplicador nas atividades do mercado interno que não estão relacionadas à exportação (MUNDURUCA, 2010). E esse efeito gera impacto pelo aumento de demanda no setor de serviço local. O aumento de demanda da economia local incrementa os níveis de emprego e renda da população (AGUIAR, 2008). Munduruca (2012, p. 613), contudo, ressalta que nenhuma atividade exportadora pode se desenvolver sem o devido apoio de determinados serviços e sem uma infraestrutura básica, como portos e outros meios de transporte, e de comunicações eficientes. Para esse autor, a base exportadora por si só não explica integralmente o crescimento econômico de um país ou região.

3 MATERIAIS E MÉTODO

A seguir estão os materiais e o método utilizados para a modelagem e a implementação do sistema obtido resultado deste trabalho.

3.1 MATERIAIS

O Quadro 1 apresenta as tecnologias e as ferramentas utilizadas para o desenvolvimento do trabalho.

Ferramenta / Tecnologia	Versão	Referência	Finalidade
Astah Community	6.8.0	http://astah.change-vision.com/	Modelagem de Casos de Uso.
Case Studio 2	2.25.0	https://www.casestudio.com	Modelagem do banco de dados.
Visual Studio 2012 Express for web	12.0	https://www.visualstudio.com/	Ambiente de desenvolvimento.
SQL server management Studio	11.0.2100.60	https://www.microsoft.com/pt-br/server-cloud/products/sql-server/default.aspx	Administrador de banco de dados.
Microsoft SQL Server 2012	6.3.9600.16384	https://www.microsoft.com/pt-br/server-cloud/products/sql-server/default.aspx	Banco de dados.
IIS (Internet Information Services)	6.1	https://www.iis.net/	Servidor <i>web</i> para a aplicação.
VB.NET e ASP.NET	10.0.30319.1	http://www.microsoft.com/net	Linguagens de programação.
HTML	5.0.0	http://www.w3.org/TR/html5/	Desenvolvimento da interface da aplicação.
CSS	3.0.0	http://www.w3.org/Style/CSS/Overview.en.html	Estilização da aplicação.
Bootstrap	3.3.5	http://getbootstrap.com/	<i>Framework front-end.</i>
JavaScript		https://www.javascript.com/	Validações em campos de formulário.

Quadro 1 – Tecnologias e ferramentas utilizadas na modelagem e na implementação

3.2 MÉTODO

O método consiste nas atividades de levantamento de requisitos, modelagem desses requisitos e implementação de funcionalidades básicas de cadastros do sistema, visando definir a forma de uso das tecnologias. A seguir descrita sucintamente a realização dessas etapas.

Levantamento de requisitos

O levantamento de requisitos foi realizado com o auxílio de uma pessoa que atua em comércio exterior trabalhando em empresa portuária de exportação no comércio de gelatinas. Essa pessoa indicou as funcionalidades desejadas para o sistema. Essas funcionalidades foram listadas do ponto de vista de negócio que eram representadas em requisitos do sistema. O mercado de exportação de gelatinas foi escolhido como escopo do sistema pelo fato de a pessoa que forneceu os requisitos atuar nesse mercado e conhecer o processo de negócio. As funcionalidades do sistema são aplicáveis a outros segmentos de comércio exterior.

Análise e projeto

Com base nas funcionalidades de negócio obtidas no levantamento dos requisitos o diagrama de casos de uso e de entidades e relacionamentos foram elaborados. Esses diagramas foram verificados pela pessoa que forneceu os requisitos avaliar se as necessidades do negócio estavam sendo atendidas.

Implementação

As funcionalidades definidas para o sistema foram implementadas utilizando as tecnologias VB.NET a ASP.NET para desenvolvimento de um aplicativo *web*. *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript foram utilizados para desenvolver a interface do aplicativo.

4 RESULTADO

Este capítulo apresenta o resultado da realização deste trabalho. O resultado está centrado na implementação de um sistema para gerenciamento de comércio exterior do mercado de gelatinas. A modelagem do sistema foi realizada como atividade de estágio pelo autor deste trabalho. Toda a documentação do sistema está presente no trabalho para facilitar o entendimento das funcionalidades desenvolvidas.

4.1 ESCOPO DO SISTEMA

O sistema visa informatizar as atividades envolvidas nas operações de comércio exterior de gelatinas. Essas operações são definidas como exportação e que tem o seu processo descrito a seguir.

Os usuários do sistema devem ser cadastrados, com permissões de acesso diferentes de acordo a função que exercem. O administrador será o responsável pela definição das permissões de cada usuário ou grupo de usuários.

O sistema permitirá a realização de cadastro de clientes, pedidos, *booking* (agendamento), especificações de gelatinas, formas de pagamentos e funcionários.

A empresa responsável pela exportação recebe, inicialmente, o pedido de seus clientes, os quais são realizados com os vendedores de cada área de mercado. Após o recebimento dos pedidos, é realizada a programação desses pedidos por meio de uma agenda para consulta de disponibilidade de data para carregamento da gelatina e o envio para o cliente.

O funcionário do setor de logística lançará esse pedido no sistema, definindo a forma de pagamento, o vendedor responsável pela venda, a quantidade, o valor, a via de transporte (marítimo, aéreo ou terrestre), as observações, a data de saída da planta (fábrica), a data prevista para o embarque e a data prevista para chegada.

O espaço no navio, avião ou caminhão será reservado com no máximo dois meses de antecedência com o agente responsável pelo pedido. E no dia da saída do pedido da fábrica é feito o cadastro de *booking* (reserva, fechamento) para o envio ao cliente.

O sistema permitirá incluir informações adicionais nos pedidos e atualização das informações iniciais, para que em casos de atraso da fábrica ou do navio, tudo seja registrado para consulta.

4.2 MODELAGEM DO SISTEMA

Quadro 2 apresenta a listagem dos requisitos funcionais identificados para o sistema.

RF significa Requisito Funcional.

Identificação	Nome	Descrição
RF01	Cadastro de usuários	Cadastrar informações de nome, <i>login</i> e senha. E definir as permissões de acesso ao sistema.
RF02	Cadastro de grupo de usuários	Agrupar usuários que possuam as mesmas permissões de acesso ao sistema.
RF03	Cadastro de funcionários	Dados dos funcionários, como nome, cargo, telefone, CPF, RG.
RF04	Cadastro de Cargos	Cargo dos funcionários da empresa.
RF05	Cadastro de clientes	Informações do cliente, nome, endereço, país, código postal, telefone, e-mail, contato, conta contábil, especificações e limite de crédito.
RF06	Cadastro de especificações	Especificações das gelatinas que cada cliente importa.
RF07	Cadastro de pedidos	Lançar os pedidos realizados com as seguintes informações: número do pedido, cliente, forma de pagamento, vendedor, quantidade, valor, especificação, via de transporte (marítimo, aéreo ou terrestre), listagem de documentos, observações, data de saída da planta, data prevista de embarque, data prevista de chegada.
RF08	Complementos de pedidos	Lançamento de informações complementares dos pedidos: data de embarque, porto de destino e o motivo dos possíveis atrasos.
RF09	Cadastro de <i>booking</i>	Registro do embarque, número do <i>booking</i> , data de embarque, data de chegada (previsão), nome do navio, nome do armador.
RF10	Relatório de pedidos	Relatórios com as informações dos pedidos realizados, pelo cliente.
RF11	Impressão de pedidos	Ao finalizar o pedido, o documento com as informações do mesmo é impresso para documentação da exportação.

Quadro 2 – Requisitos funcionais

O Quadro 3 apresenta os requisitos não-funcionais identificados para o sistema, também denominados de requisitos suplementares. Os requisitos não funcionais explicitam

regras de negócio, restrições ao sistema de acesso, requisitos de qualidade, desempenho e segurança, dentre outros. RNF significa Requisito Não Funcional.

Identificação	Nome	Descrição
RNF01	Acesso ao sistema	O acesso ao sistema será realizado por meio de <i>login</i> e senha.
RNF02	Pedidos	Cada usuário poderá visualizar todos os pedidos realizados no sistema, mas somente irá editar os que estão em seu nome. Esse controle será feito por meio do vínculo do usuário ao cadastro de funcionário.
RNF03	Permissões	O usuário administrador irá, dentro do cadastro de usuário, definir para quais opções do sistema o funcionário terá acesso e quais as operações que poderão ser realizadas (Inclusão, edição e exclusão).
RNF04	Desempenho	O sistema será <i>web</i> , não tendo a necessidade de requisitos de alto desempenho em termos de hardware para a sua execução em decorrência do tipo de operações realizadas e do volume de dados manipulados.

Quadro 3 – Requisitos não funcionais

O diagrama de casos de uso apresentado na Figura 1 contém as funcionalidades essenciais do sistema. Essas funcionalidades são realizadas por dois atores que são: Administrador e Assistente de logística e de vendas. O Administrador é responsável pelos cadastros de acesso ao sistema. O Assistente de logística e vendas é quem realiza as atividades de cadastrar os pedidos, manter nos cadastros das informações necessárias para realizar os pedidos e informar o plano de pagamento do pedido realizado.

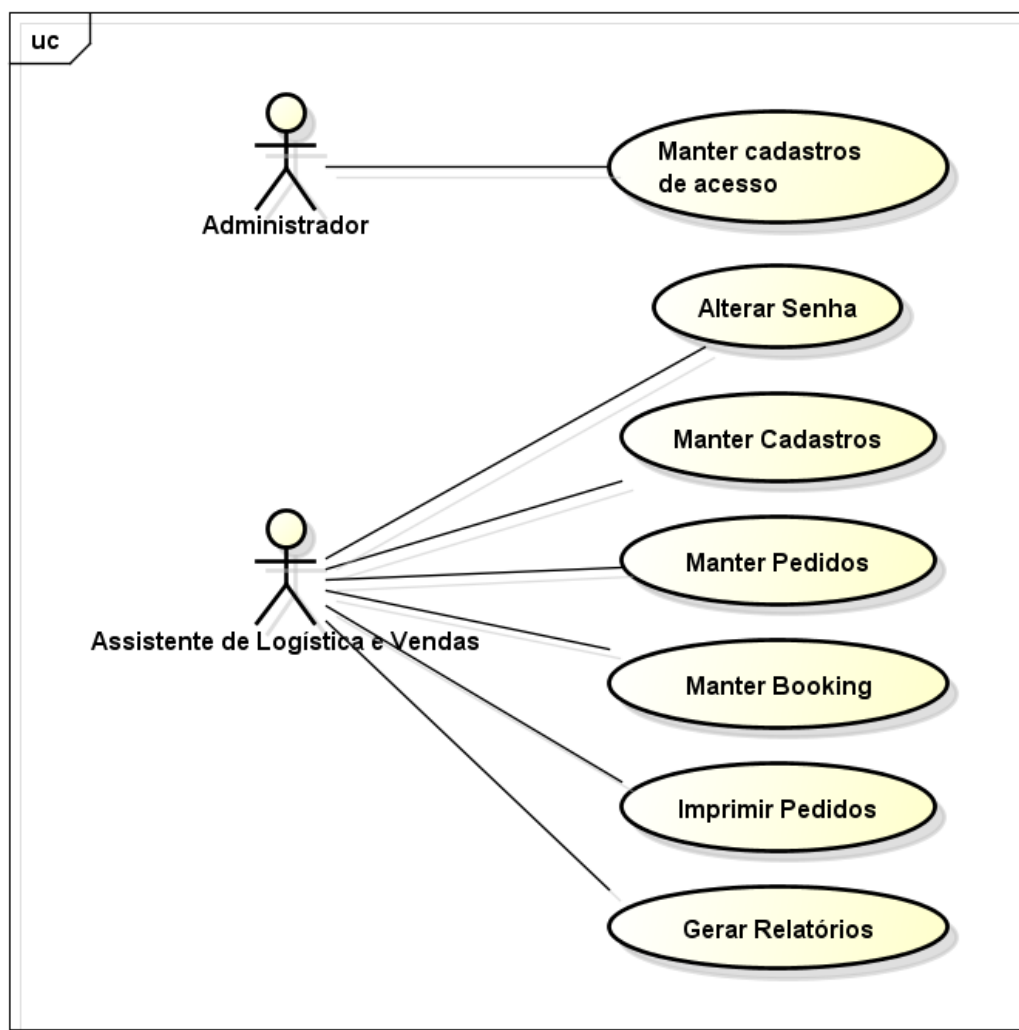


Figura 1 – Diagrama de casos de uso

No Quadro 4 está a descrição da funcionalidade incluir dos casos de uso de cadastro. Essa funcionalidade se aplica para todos os casos de uso que possuem a operação de inclusão de registros no banco de dados.

Caso de uso:

Incluir registros.

Descrição:

Esse caso de uso se refere às operações de inclusão dos registros necessários para a utilização do sistema.

Evento iniciador:

Necessidade de incluir um registro.

Atores:

De acordo com as permissões definidas pelo administrador.

Pré-condição:

Dados necessários disponíveis e permissão para acessar o cadastro desejado.

Sequência de eventos:

1. Ator seleciona o formulário no qual deseja realizar a operação.
2. Sistema apresenta o formulário.

<p>3. Ator realiza inserção dos dados solicitados no formulário.</p> <p>4. Sistema verifica se os dados são válidos e realiza a operação de inclusão.</p> <p>Pós-condição: Registro incluído com sucesso.</p>	
Validação de dados	Descrição
Linha 4: Dados não válidos.	<p>4.1 No momento de salvar o registro, o sistema faz a verificação e constata que há dados inválidos. É emitida mensagem e retornado para o formulário em estado de inclusão, indicando os campos nos quais há dados inválidos.</p> <p>Retorna ao passo 3 e o caso de uso prossegue com a sequência normal.</p>

Quadro 4 – Funcionalidade incluir dos casos de uso de cadastro

No Quadro 5 está a descrição da funcionalidade alterar dos casos de uso de cadastro.

<p>Caso de uso: Alterar registros</p> <p>Descrição: Esse caso de uso se refere às operações de alteração dos registros necessários para a utilização do sistema.</p> <p>Evento iniciador: Necessidade de alterar dados de um registro.</p> <p>Atores: Usuários que possuem permissão de acesso ao formulário.</p> <p>Pré-condição: Permissão para acessar o formulário desejado.</p> <p>Sequência de eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona o formulário no qual deseja realizar alteração de dados. 2. Sistema apresenta o formulário. 3. Ator realizar alteração dos dados. 4. Sistema verifica se os dados são válidos e realiza a alteração dos dados no registro. <p>Pós-condição: Registro alterado com sucesso.</p>	
Validação de dados	Descrição
Linha 4: Dados não válidos.	<p>4.1 No momento de salvar, o sistema faz a verificação e constata que há dados inválidos. É emitida mensagem e retornado para o formulário em estado de edição.</p> <p>Retorna ao passo 3 e o caso de uso prossegue com a sequência normal.</p>

Quadro 5 – Funcionalidade alterar dos casos de uso de cadastro

No Quadro 6 está a descrição da funcionalidade excluir dos casos de uso de cadastro do sistema.

<p>Caso de uso: Excluir registros</p> <p>Descrição: Esse caso de uso se refere às operações de exclusão dos registros no sistema.</p> <p>Evento iniciador: Necessidade de excluir um registro.</p> <p>Atores: De acordo com as permissões definidas pelo administrador.</p> <p>Pré-condição: Permissão para acessar o formulário desejado.</p> <p>Sequência de eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona o módulo do registro que deseja excluir. 2. Sistema apresenta a tela com o <i>grid</i> dos registros armazenados na respectiva tabela do banco de dados. 3. Ator seleciona o registro a ser excluído. 4. Sistema verifica se não existem vínculos com outras tabelas e realiza a operação de exclusão do registro. <p>Pós-condição: Registro excluído com sucesso.</p>	
Registro não pode ser excluído	Descrição
Linha 4: Exclusão de registro com dados de tabelas vinculadas ativas.	4.1 Se solicitada a exclusão de registro que possui tabelas vinculadas com dados ativos, é informada mensagem que não é possível realizar a exclusão. Sistema informa o usuário e permanece em estado anterior à realização da operação na Linha 3.

Quadro 6 – Funcionalidade excluir dos casos de uso de cadastro

No Quadro 7 está a descrição da funcionalidade consultar dos casos de uso de cadastro.

<p>Caso de uso: Consultar registros</p> <p>Descrição: Esse caso de uso se refere à consulta dos registros necessários para a utilização do sistema.</p> <p>Evento iniciador: Necessidade de consultar dados de um registro.</p> <p>Atores: De acordo com as permissões definidas pelo administrador.</p> <p>Pré-condição: Permissão para acessar o formulário desejado.</p> <p>Sequência de eventos:</p> <ol style="list-style-type: none"> 1. Ator seleciona o formulário do registro que deseja consultar. 2. Sistema apresenta a tela com o <i>grid</i> dos registros salvos. 3. Ator seleciona o registro que deseja consultar. 4. Sistema exibe a tela com as informações do registro. <p>Pós-condição: Consulta realizada com sucesso.</p>	
Inexistência de registros	Descrição
Linha 2: Tabela sem registros cadastrados.	4.1. É emitida mensagem para o usuário,

<p>informando que não existem registros cadastrados. Retorna ao passo 1 e o caso de uso prossegue com a sequência normal.</p>

Quadro 7 – Funcionalidade consultar dos casos de uso de cadastro

No Quadro 8 está a descrição do caso de uso imprimir pedidos.

<p>Caso de uso: Imprimir pedidos.</p> <p>Descrição: Esse caso de uso se refere à impressão dos pedidos realizados.</p> <p>Evento iniciador: Finalização de um cadastro de pedido.</p> <p>Atores: De acordo com as permissões definidas pelo administrador.</p> <p>Pré-condição: Permissão para impressão de pedidos.</p> <p>Sequência de eventos:</p> <ol style="list-style-type: none"> 1. Ator realiza o pedido no sistema e solicita a impressão. 2. Sistema apresenta a impressão do documento com as informações do pedido. <p>Pós-condição: Pedido impresso com sucesso.</p>

Quadro 8 – Caso de uso de imprimir pedidos

No Quadro 9 está a descrição do caso de uso gerar relatórios.

<p>Caso de uso: Gerar relatórios.</p> <p>Descrição: Esse caso de uso se refere à geração de relatórios.</p> <p>Evento iniciador: Necessidade de emitir um relatório.</p> <p>Atores: De acordo com as permissões definidas pelo administrador.</p> <p>Pré-condição: Permissão para geração de relatórios.</p> <p>Sequência de eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela de relatórios. 2. Ator informa os filtros que deseja realizar. 3. Sistema exibe o relatório solicitado. <p>Pós-condição: Relatório impresso com sucesso.</p>

Quadro 9 – Caso de uso de gerar relatórios

Figura 2 apresenta o diagrama de entidades e relacionamentos que representam o banco de dados da aplicação.

[1.1]

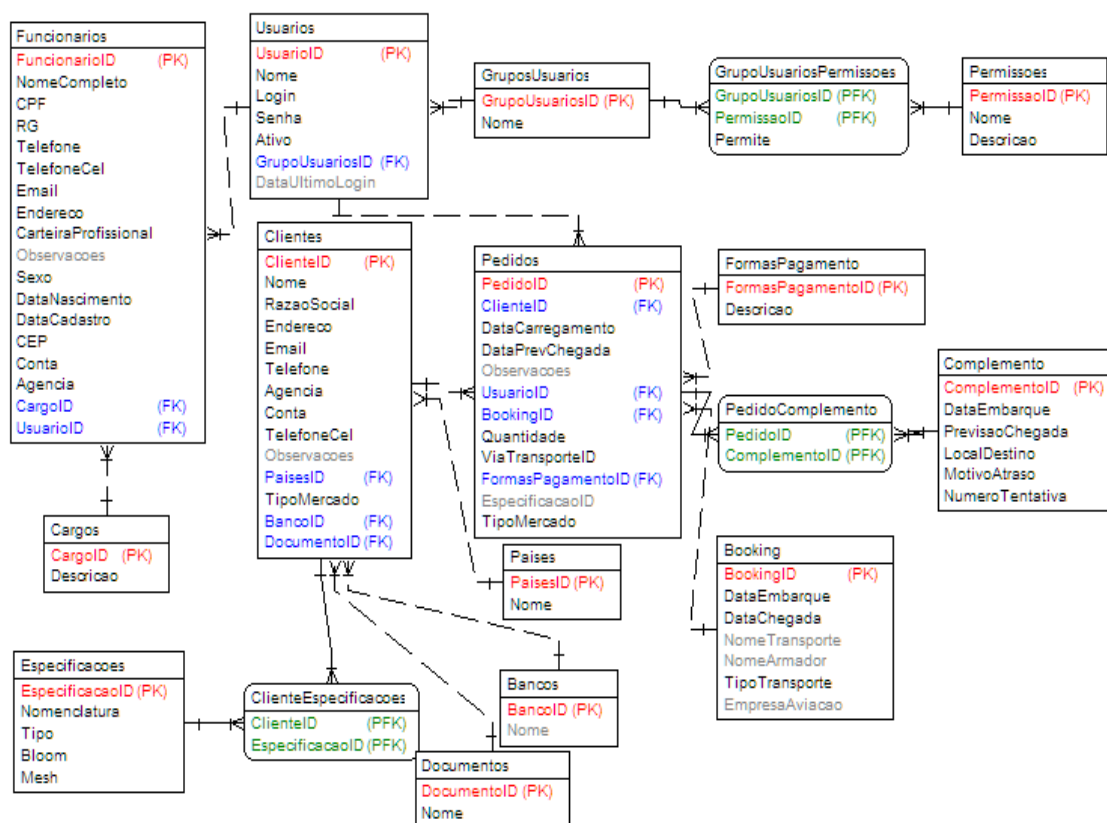


Figura 2 – Diagrama de entidades e relacionamentos do banco de dados

No Quadro 10 estão os campos da tabela de funcionários. Funcionários da empresa e que poderão ter acesso ao sistema.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
FuncionarioID	Numérico	Não	Sim	Não	
NomeCompleto	Texto	Não	Não	Não	
CPF	Texto	Não	Não	Não	
RG	Texto	Não	Não	Não	
Telefone	Texto	Não	Não	Não	
TelefoneCel	Texto	Não	Não	Não	
Email	Texto	Não	Não	Não	
Endereço	Texto	Não	Não	Não	
CarteriaProfissional	Texto	Não	Não	Não	
Observacoes	Texto	Sim	Não	Não	
Sexo	Caractere	Não	Não	Não	
DataNascimento	Data	Não	Não	Não	
DataCadastro	Data	Não	Não	Não	
CEP	Texto	Não	Não	Não	
Conta	Texto	Não	Não	Não	
Agencia	Texto	Não	Não	Não	

CargoID	Numérico	Não	Não	Sim	Da tabela Cargos
UsuarioID	Numérico	Sim	Não	Sim	Da tabela Usuarios

Quadro 10 – Campos da tabela Funcionarios

No Quadro 11 estão os campos da tabela de cargos. Os cargos são definidos para cada funcionário de acordo com as suas competências.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CargoID	Numérico	Não	Sim	Não	
Descricao	Texto	Não	Não	Não	

Quadro 11 – Campos da tabela Cargos

No Quadro 12 estão os campos da tabela de usuários. Para que seja possível acessar o sistema, um funcionário precisará estar cadastrado como usuário do sistema.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
UsuarioID	Numérico	Não	Sim	Não	
Nome	Texto	Não	Não	Não	
Login	Texto	Não	Não	Não	
Senha	Texto	Não	Não	Não	
Ativo	Numérico	Não	Não	Não	
Grupo UsuariosID	Numérico	Não	Não	Sim	Da tabela GruposUsuarios

Quadro 12 – Campos da tabela Usuarios

No Quadro 13 estão os campos da tabela de grupos de usuários. Será utilizado para agrupar os usuários que exercem a mesma atividade.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
GrupoUsuarioID	Numérico	Não	Sim	Não	
Nome	Texto	Não	Não	Não	

Quadro 13 – Campos da tabela GruposUsuarios

No Quadro 14 estão os campos da tabela de grupos usuários permissões. Serão definidas as permissões vinculadas aos grupos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
GrupoUsuarioID	Numérico	Não	Sim	Não	
PermissaoID	Numérico	Não	Não	Sim	Da tabela Permissoes
Permite	Numérico	Não	Não	Não	

Quadro 14 – Campos da tabela GrupoUsuariosPermissoes

No Quadro 15 estão os campos da tabela de permissões de acesso ao sistema.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
PermissaoID	Numérico	Não	Sim	Não	
Nome	Texto	Não	Não	Não	
Descricao	Texto	Não	Não	Não	

Quadro 15 – Campos da tabela Permissoes

No Quadro 16 estão os campos da tabela de clientes.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ClienteID	Numérico	Não	Sim	Não	
Nome	Texto	Não	Não	Não	
RazaoSocial	Texto	Não	Não	Não	
Endereco	Texto	Não	Não	Não	
Email	Texto	Não	Não	Não	
Telefone	Texto	Não	Não	Não	
Agencia	Texto	Não	Não	Não	
Conta	Texto	Não	Não	Não	
TelefoneCel	Texto	Não	Não	Não	
Observacoes	Texto	Sim	Não	Não	
PaísesID	Numérico	Não	Não	Sim	Da tabela Países

Quadro 16 – Campos da tabela Clientes

As especificações das gelatinas são definidas de acordo com a exigência do cliente.

No Quadro 17 estão os campos da tabela que vincula os clientes e especificações.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ClienteID	Numérico	Não	Não	Sim	Da tabela Clientes
EspecificacaoID	Numérico	Não	Não	Sim	Da tabela Especificacoes

Quadro 17 – Campos da tabela ClienteEspecificacoes

No Quadro 18 estão os campos da tabela de especificações.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
EspecificacaoID	Numérico	Não	Sim	Não	
Nomenclatura	Texto	Não	Não	Não	
Tipo	Numérico	Não	Não	Não	
Bloom	Numérico	Não	Não	Não	
Mesh	Numérico	Não	Não	Não	

Quadro 18 – Campos da tabela Especificacoes

No Quadro 19 estão os campos da tabela de pedidos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
PedidoID	Numérico	Não	Sim	Não	
ClienteID	Numérico	Não	Não	Sim	Da tabela Clientes
DataCarregamento	Data	Não	Não	Não	
DataPrevChegada	Data	Não	Não	Não	
Observacoes	Texto	Sim	Não	Não	
UsuarioID	Numérico	Não	Não	Sim	Da tabela Usuarios
BookingID	Numérico	Não	Não	Sim	Da tabela Booking
Quantidade	Numérico	Não	Não	Não	
ViaTransporteID	Numérico	Não	Não	Não	
FormasPagamentoID	Numérico	Não	Não	Sim	Da tabela FormasPagamento

Quadro 19 – Campos da tabela Pedidos

No Quadro 20 estão os campos da tabela de *booking*. Após a criação do pedido no sistema, será feita a consulta de datas disponíveis para o envio do produto e registrado o *booking* dentro do sistema.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
BookingID	Numérico	Não	Sim	Não	
DataEmbarque	Data	Não	Não	Não	
DataChegada	Data	Não	Não	Não	
NomeTransporte	Texto	Sim	Não	Não	
NomeArmador	Texto	Sim	Não	Não	
TipoTransporte	Numérico	Não	Não	Não	
EmpresaAviacao	Texto	Sim	Não	Não	

Quadro 20 – Campos da tabela Booking

No Quadro 21 estão os campos da tabela de formas de pagamento.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
FormasPagamentoID	Numérico	Não	Sim	Não	
Descricao	Texto	Não	Não	Não	

Quadro 21 – Campos da tabela FormasPagamento

Quando ocorre o atraso de algum envio de pedido, é cadastrado no sistema um complemento do pedido com as informações sobre o atraso. No Quadro 22 estão os campos da tabela que vincula o pedido ao complemento.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
PedidoID	Numérico	Não	Não	Sim	Da tabela Pedidos
ComplementoID	Numérico	Não	Não	Sim	Da tabela Complementos

Quadro 22 – Campos da tabela PedidoComplemento

No Quadro 23 estão os campos da tabela de complementos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ComplementoID	Numérico	Não	Sim	Não	
DataEmbarque	Data	Não	Não	Não	
PrevisaoChegada	Data	Não	Não	Não	
LocalDestino	Texto	Não	Não	Não	
MotivoAtraso	Texto	Não	Não	Não	
NumeroTentativa	Numérico	Não	Não	Não	

Quadro 23 – Campos da tabela Complemento

No Quadro 24 estão os campos da tabela de países do cadastro dos clientes.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
PaísesID	Numérico	Não	Sim	Não	
Nome	Texto	Não	Não	Não	

Quadro 24 – Campos da tabela Países

4.3 APRESENTAÇÃO DO SISTEMA

A tela principal do aplicativo é apresentada na Figura 3. Essa tela está organizada em três áreas ou regiões:

- a) na parte superior direita é apresentada a opção de acesso ao cadastro do funcionário que está vinculado ao usuário que está acessando o sistema;
- b) na área esquerda estão os menus, organizados nos grupos de funcionalidades, sendo: Central de informações, clientes, pedidos, relatórios e gerencial;
- c) na área central da página são apresentados os formulários e telas do sistema.

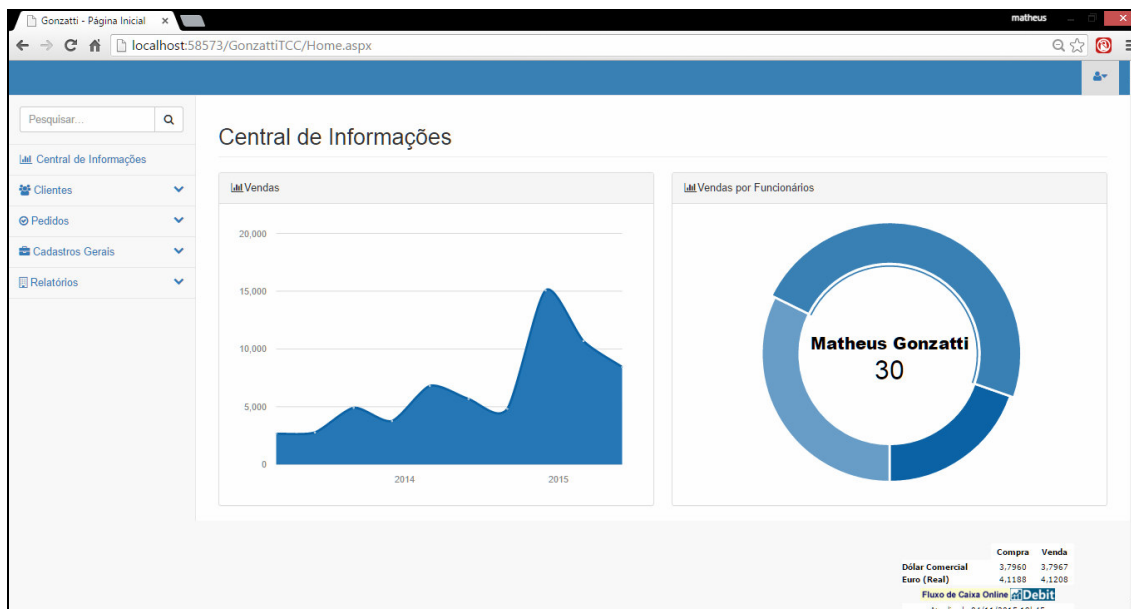


Figura 3 – Tela principal do aplicativo

O cadastro do cliente é apresentado nas Figuras 4, 5 e 6, que consiste em três etapas: os dados do cliente, como nome e razão social, informações de endereço e comunicação e por fim outras informações, como número de conta/agência e observações. As três telas estão em uma mesma página HTML, a qual foi dividida em abas para agrupar as informações que possuem o mesmo propósito, são elas: identificação do cliente, informações de comunicação, e informações complementares.

The screenshot shows the client registration form. The title is 'Clientes'. On the left, there is a search bar and a navigation menu with options: 'Central de Informações', 'Clientes', 'Pedidos', 'Cadastros Gerais', and 'Relatórios'. The main content area contains a form with three tabs: 'Dados do Cliente', 'Endereço / Comunicação', and 'Outras Informações'. The 'Dados do Cliente' tab is active, showing fields for 'Nome' and 'Razão Social'. A 'Salvar' button is visible at the bottom right. In the bottom right corner, there is a table with exchange rates:

	Compra	Venda
Dólar Comercial	3,7960	3,7967
Euro (Real)	4,1188	4,1208

Below the table, it says 'Fluxo de Caixa Online' and 'Atualizado 04/11/2015 19h45'.

Figura 4 – Formulário de cadastro de clientes

As informações relacionadas ao endereço são apresentadas na Figura 5.

Clientes

Dados do Cliente
 Endereço / Comunicação
 Outras Informações

Endereço:

Telefone:

E-mail:

Celular:

País:

	Compra	Venda
Dólar Comercial	3,7960	3,7967
Euro (Real)	4,1188	4,1208

Fluxo de Caixa Online [Debit](#)

Atualizado 04/11/2015 19h45

Figura 5 – Formulário de cadastro de clientes

Os dados bancários são apresentados na tela da Figura 6.

Clientes

Dados do Cliente
 Endereço / Comunicação
 Outras Informações

Agencia:

Conta:

Observações:

	Compra	Venda
Dólar Comercial	3,7960	3,7967
Euro (Real)	4,1188	4,1208

Fluxo de Caixa Online [Debit](#)

Atualizado 04/11/2015 19h45

Figura 6 – Formulário de cadastro de clientes

A listagem dos clientes cadastrados é apresentada na Figura 7. Esse é o padrão de listagem dos dados dos cadastros.

Código	Nome do Cliente	Produto	Quantidade	Data do Carregamento	Data de Previsão de Chegada		
1	Matheus Ribeiro Gonzatti	Teste - Bovina; Mesh: 1233 / Bloom: 123	12000 Kg	11/11/2015	11/12/2015	Editar	Excluir

Fluxo de Caixa Online
 Atualizado 13/11/2015 19h50

	Compra	Venda
Dólar Comercial	3.8324	3.8331
Euro (Real)	4.1108	4.1132

Figura 7 – Listagem de clientes

Nas Figuras 8, 9 e 10, é apresentada a tela de pedido. Na Figura 8, são as informações principais do pedido, como o cliente, o produto que será exportado, a quantidade, qual é a data do seu carregamento e a sua previsão de chegada.

Pedidos

☰ Dados do Pedido

Cliente: (Selecione) ▾

Especificação do Produto: ▾

Quantidade (Kg): 0,00

Forma de Pagamento: (Selecione) ▾

Observações:

Data do Carregamento:

Data Previsão de Chegada:

Via de Transporte:

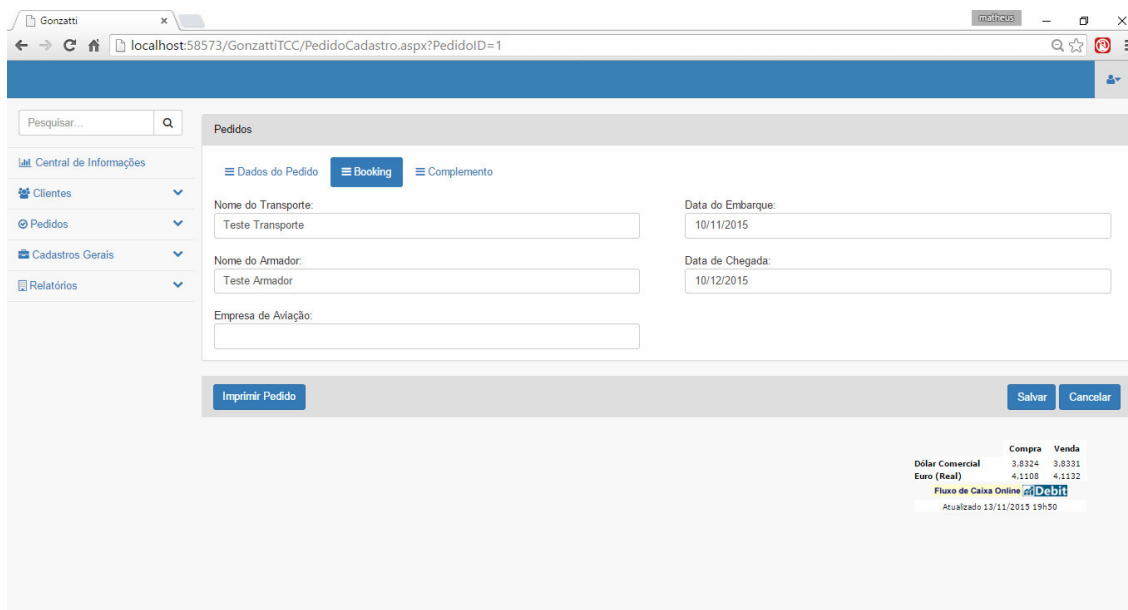
Salvar Cancelar

	Compra	Venda
Dólar Comercial	3.8324	3.8331

Figura 8 – Cadastro de pedidos

Se a data de carregamento estiver próxima é verificado o meio de transporte que será utilizado. Para isso é utilizado a tela apresentada na Figura 9. Nessa tela é informada a empresa responsável pelo envio, o nome identificador do meio de transporte, a data em que

será embarcado, o dia em que chegará ao seu destino e no caso de envio aéreo o nome da empresa que fará o transporte.



Pesquisar...

Central de Informações

Clientes

Pedidos

Cadastros Gerais

Relatórios

Pedidos

Dados do Pedido

Booking

Complemento

Nome do Transporte: Teste Transporte

Data do Embarque: 10/11/2015

Nome do Armador: Teste Armador

Data de Chegada: 10/12/2015

Empresa de Aviação:

Imprimir Pedido

Salvar

Cancelar

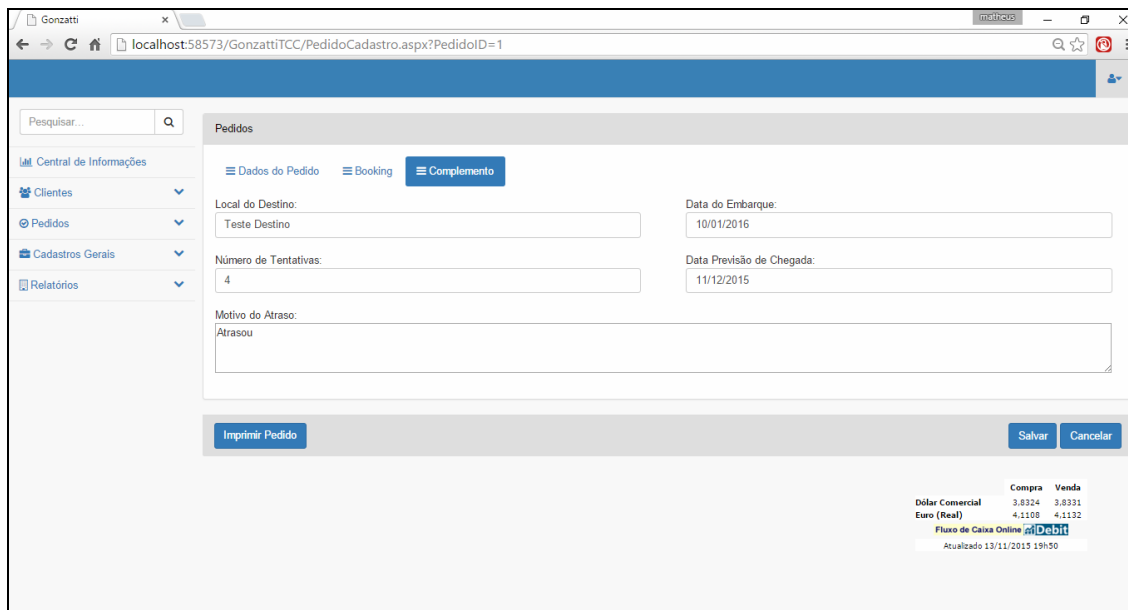
	Compra	Venda
Dólar Comercial	3,8324	3,8331
Euro (Real)	4,1108	4,1132

Fluxo de Caixa Online **Debit**

Atualizado 13/11/2015 19h50

Figura 9 – Cadastro de pedidos

Podem acontecer imprevistos, ocasionando atrasos. Esses atrasos podem ser registrados no cadastro apresentado na Figura 10, com as possíveis novas datas de embarque e chegada no destino, quantas vezes já foi tentado o envio e os motivos dos atrasos.



Pesquisar...

Central de Informações

Clientes

Pedidos

Cadastros Gerais

Relatórios

Pedidos

Dados do Pedido

Booking

Complemento

Local do Destino: Teste Destino

Data do Embarque: 10/01/2016

Número de Tentativas: 4

Data Previsão de Chegada: 11/12/2015

Motivo do Atraso: Atrasou

Imprimir Pedido

Salvar

Cancelar

	Compra	Venda
Dólar Comercial	3,8324	3,8331
Euro (Real)	4,1108	4,1132

Fluxo de Caixa Online **Debit**

Atualizado 13/11/2015 19h50

Figura 10 – Cadastro de pedidos

Na Figura 11 é apresentada a tela de filtro do relatório de pedidos. O filtro implementado foi o de clientes.

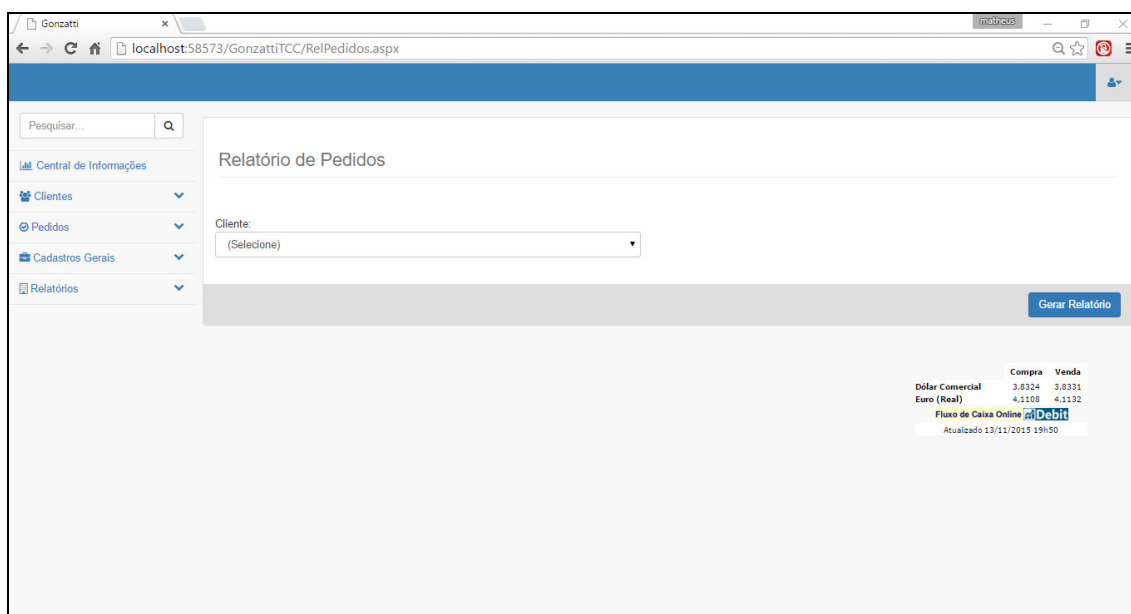


Figura 11 – Filtro do relatório de pedidos

Na Figura 12 é apresentado o relatório de pedidos por cliente.

Relatório de Pedidos				
Código do Cliente		Nome do Cliente		
1		Matheus Ribeiro Gonzatti		
Informações do Pedido:				
Produto	Quantidade	Carregamento	Previsão Chegada	Via Transporte
Teste - Bovina; Mesh: 1233 / Bloom: 123	12000 Kg	11/11/2015	11/12/2015	Marítimo
Observações do Pedido				
Informações do Envio:				
Nome Transporte	Nome Armador	Empresa Aviação	Data Embarque	Data Chegada
Teste Transporte	Teste Armador		10/11/2015	10/12/2015
Local Destino				
Teste Destino				
Data de Embarque (Complemento)			Numero de Tentativas	
10/01/2016			4	
Motivo do Atraso				
Atrasou				
Total de Páginas:				1

Figura 12 – Relatório de pedidos

4.4 IMPLEMENTAÇÃO DO SISTEMA

A seguir são apresentados trechos da codificação realizada com o objetivo de mostrar o uso dos recursos das tecnologias que foram empregadas no desenvolvimento do sistema.

A Listagem 1 apresenta trecho de código referente à página principal e de cadastro de clientes.

```
<%@ Master Language="VB" AutoEventWireup="true" CodeBehind="Site.master.vb"
Inherits="GonzattiTCCv1._0.MasterPage" %>

<%@ Page Title="Gonzatti - Cadastro de Clientes" Language="vb" AutoEventWireup="false"
MasterPageFile="~/Site.Master" CodeBehind="ClienteCadastro.aspx.vb"
Inherits="GonzattiTCC.ClienteCadastro" %>
<asp:Content ID="Content1" ContentPlaceHolderID="bodyContent" runat="server">
</asp:Content>

<div style="float:right; padding-right:5%; padding-top:2%;"><script
src="http://www.debit.com.br/resumogratico.php?info=novo_dolar"></script></div>
```

Listagem 1 – Site.MasterPage e ClienteCadastro.aspx.vb

Explicação do código da Listagem 1: no ASP.NET é feito o uso de diretivas que especificam as configurações que serão utilizadas pelos compiladores. Na diretiva @Page são definidos os atributos de página que serão analisados pelo ASP.NET. Foi criada uma página base chamada de Site.Master que conterá a estrutura básica das telas do sistema. Nela, ao invés de ser utilizada a diretiva @Page, foi utilizada a @Master. Para inserir conteúdos dentro dessa página é adicionado o controle <asp: ContentPlaceHolder > e, assim, as demais páginas do sistema podem herdar o conteúdo dessa *master page*. No exemplo é possível verificar que foi inserida a configuração MasterPageFile para referenciar qual é a *master* que será utilizada. Para associar o conteúdo de uma página a *master* é incluído o controle <asp:Content > e por meio propriedade ContentPlaceHolderID, o mesmo será vinculado com o ContentPlaceHolder das *master page*. A última linha apresentada, é um script utilizado para trazer as informações em tempo real das cotações de moedas.

Para criar os menus laterais, foi utilizada a tag e a tag para os itens dessa lista não ordenada. A Listagem 2 exemplifica essas *tags*. Foram utilizadas classes do Bootstrap para a estilização e a propriedade data-toggle="collapse" para o efeito de seleção das opções nos menus.

```

<li>
  <a data-toggle="collapse" href="#ulClientes">
    <i class="fa fa-group"></i>
    Clientes <span class="fa fa-chevron-down" style="float: right;"></span></a>
    <ul id="ulClientes" class="nav nav-second-level collapse" aria-expanded="false">
      <li>
        <a href="Clientes.aspx">Consultar Clientes</a>
      </li>
      <li>
        <a href="ClienteCadastro.aspx">Novo Cliente</a>
      </li>
    </ul>
  </li>

```

Listagem 2 – ClienteCadastro.aspx.vb

Dentro do *code-behind* (Listagem 1) da página de cadastro dos clientes é instanciado um objeto da classe Clientes e é carregado o objeto com as informações dos controles do ASP.NET. Dentro do comando de *try catch* (Listagem 3) é verificada a operação que está sendo realizada. Caso a *request* da página tenha o conteúdo ClienteID é chamado o método Alterar da classe Clientes e passado por parâmetro o objeto e o id do cliente a ser alterado. Caso não exista informação na *request*, é chamado o método incluir e passado o objeto por parâmetro. Após a operação ser concluída com sucesso, o usuário é redirecionado a página de listagem dos clientes.

```

Imports GonzattiTCC.Clientes
Imports GonzattiTCC.Validacoes
Imports System.Data.SqlClient

Public Class ClienteCadastro
    Inherits System.Web.UI.Page

    Protected Sub btnSalvar_Click(ByVal sender As Object, ByVal e As System.EventArgs)
        Handles btnSalvar.Click
            Dim oCliente As New Cliente

            With oCliente
                .Nome = txtNome.Text
                .RazaoSocial = txtRazaoSocial.Text
                .Endereco = txtEndereco.Text
                .Email = txtEmailPrincipal.Text
                .Agencia = txtAgencia.Text
                .Conta = txtConta.Text
                .Telefone = txtFoneResidencial.Text
                .TelefoneCel = txtCelular.Text
                .Observacoes = txtObservacao.Text
                .Paises = cmbPaises.SelectedValue
            End With

            Try
                If Not Request("ClienteID") Is Nothing Then
                    Alterar(oCliente, Val(Request("ClienteID")), Transaction)
                Else
                    Incluir(oCliente, Transaction)
                End If
            Catch ex As Exception

```

```

        MsgBox("O registro não foi alterado!", MsgBoxStyle.Information, "Atenção")
    Finally
        Response.BufferOutput = True
        Response.Redirect("Cliente.aspx")
    End Try
End Sub
End Class

```

Listagem 3 – btnSalvar_click() da classe cliente

Dentro da classe de Clientes (Listagem 4) são declarados a os seus membros de acordo com o seu tipo, utilizando o modificador *private* para restringir o acesso por outras classes. Também são inseridos os *gets* e *sets* das propriedades.

```

Imports System.Data.SqlClient
Imports GonzattiTCCv1._0.Conexao

Public Class Cliente

    Private nClienteID As Integer
    Private sNome As String
    Private sRazaoSocial As String
    Private sEndereco As String
    Private sEmail As String
    Private sTelefone As String
    Private sAgencia As String
    Private sConta As String
    Private sTelefoneCel As String
    Private sObservacoes As String
    Private nPaisID As Integer
    Private nTipoMercado As Integer
    Private nBancoID As Integer

    Public Property Nome() As String
        Get
            Return sNome
        End Get
        Set(ByVal Value As String)
            sNome = Value
        End Set
    End Property
    Public Shared Sub Incluir(ByVal oCliente As Cliente, Optional ByRef Transaction As
    SqlTransaction = Nothing)

        Dim oConexao As SqlConnection = New SqlConnection(RetConnectionString())
        Dim cmdClientes As New SqlCommand("INSERT INTO Clientes (ClienteID, Nome,
    RazaoSocial, Endereco, Email,Telefone, Agencia, Conta, TelefoneCel, Observacoes,
    PaisID, TipoMercado, BancoID, DocumentoID) VALUES(@ClienteID,@Nome,@RazaoSocial,
    @Endereco, @Email, @Telefone, " "@Agencia, @Conta, @TelefoneCel, @Observacoes,
    @PaisID, @TipoMercado, @BancoID, @DocumentoID)", oConexao)

        Try
            oConexao.Open()

            cmdClientes.Parameters.Clear()
            Transaction = oConexao.BeginTransaction
            cmdClientes.Transaction = Transaction

            Dim nClienteID As Integer

```

```

nClienteID = Conexao.RetProximoCodigo("Clientes", "ClienteID", Transaction)

cmdClientes.Parameters.Add(New SqlParameter("@ClienteID", nClienteID))
cmdClientes.Parameters.Add(New SqlParameter("@Nome", oCliente.Nome))
cmdClientes.Parameters.Add(New SqlParameter("@RazaoSocial", oCliente.RazaoSocial))
cmdClientes.Parameters.Add(New SqlParameter("@Endereco", oCliente.Endereco))
cmdClientes.Parameters.Add(New SqlParameter("@Email", oCliente.Email))
cmdClientes.Parameters.Add(New SqlParameter("@Telefone", oCliente.Telefone))
cmdClientes.Parameters.Add(New SqlParameter("@Agencia", oCliente.Agencia))
cmdClientes.Parameters.Add(New SqlParameter("@Conta", oCliente.Conta))
cmdClientes.Parameters.Add(New SqlParameter("@TelefoneCel", oCliente.TelefoneCel))
cmdClientes.Parameters.Add(New SqlParameter("@Observacoes", oCliente.Observacoes))
cmdClientes.Parameters.Add(New SqlParameter("@PaisID", oCliente.Pais))
cmdClientes.Parameters.Add(New SqlParameter("@TipoMercado", oCliente.TipoMercado))
cmdClientes.Parameters.Add(New SqlParameter("@BancoID", oCliente.BancoID))
cmdClientes.Parameters.Add(New SqlParameter("@DocumentoID", oCliente.DocumentoID))

Catch ex As Exception
    Transaction.Rollback()
    oConexao.Close()
Finally
    cmdClientes.ExecuteNonQuery()
    Transaction.Commit()
    oConexao.Close()
End Try
End Sub

```

Listagem 4 – Classe cliente.vb

Ainda dentro da classe Clientes (Listagem 4) está o método incluir, recebendo o objeto com as informações a serem inseridas no banco e a transação a ser utilizada. É criada uma conexão com o banco e nesse caso é chamada a função RetConnectionString() para que seja obtida o endereço de conexão com o banco (SQL Server). Tendo a conexão criada, é possível abri-la com a propriedade open() e então é declarado o comando a ser executado como um novo sqlCommand, adicionamos os parâmetros dentro do comando e é executado o comando *Structured Query Language* (SQL) com o ExecuteNonQuery() e é fechada a conexão. O método Alterar() segue o mesmo fluxo do método Incluir(), que receberá por parâmetro, além do objeto e a transação, o id do cliente a ser alterado dentro do banco pelo comando SQL UPDATE.

Dentro da classe de Pedidos (Listagem 5) são declarados os seus membros de acordo com o seu tipo, utilizando o modificador *private* para restringir o acesso por outras classes. Também são inseridos os *getters* e *setters* das propriedades.

```

Public Class Pedido
    Private nPedidoID As Integer
    Private nClienteID As Integer
    Private dDataCarregamento As Date
    Private dDataPrevChegada As Date
    Private sObservacoes As String
    Private nUsuarioID As Integer
    Private nBookingID As Integer
    Private nQuantidade As Integer

```



```

Private nViaTransporteID As Integer
Private nFormasPagamentoID As Integer
Private nEspecificacaoID As Integer

Public Property PedidoID() As Integer
    Get
        Return nPedidoID
    End Get
    Set(ByVal Value As Integer)
        nPedidoID = Value
    End Set
End Property

Public Shared Sub Incluir(ByVal oPedido As Pedido, Optional ByRef Transaction As
SqlTransaction = Nothing)

    Dim oConexao As SqlConnection = New SqlConnection(RetConnectionString())

    Dim cmdPedidos As New SqlCommand("INSERT INTO Pedidos(PedidoID, ClienteID,
DataCarregamento, DataPrevChegada, Observacoes, UsuarioID,
BookingID, Quantidade, ViaTransporteID, FormasPagamentoID, EspecificacaoID)
VALUES (@PedidoID, @ClienteID, @DataCarregamento, @DataPrevChegada, @Observacoes,
@UsuarioID, BookingID, @Quantidade, @ViaTransporteID, @FormasPagamentoID,
@EspecificacaoID)", oConexao, Transaction)

    Try
        oConexao.Open()

        cmdPedidos.Parameters.Clear()
        Transaction = oConexao.BeginTransaction
        cmdPedidos.Transaction = Transaction

        Dim nPedidoID As Integer

        nPedidoID = Conexao.RetProximoCodigo("Pedidos", "PedidoID", Transaction)

        cmdPedidos.Parameters.Add(New SqlParameter("@PedidoID", nPedidoID))
        cmdPedidos.Parameters.Add(New SqlParameter("@ClienteID", oPedido.ClienteID))
        cmdPedidos.Parameters.Add(New SqlParameter("@DataCarregamento",
oPedido.DataCarregamento))
        cmdPedidos.Parameters.Add(New SqlParameter("@DataPrevChegada",
oPedido.DataPrevChegada))
        cmdPedidos.Parameters.Add(New SqlParameter("@Observacoes", oPedido.Observacoes))
        cmdPedidos.Parameters.Add(New SqlParameter("@UsuarioID", oPedido.UsuarioID))
        cmdPedidos.Parameters.Add(New SqlParameter("@BookingID", 0))
        cmdPedidos.Parameters.Add(New SqlParameter("@Quantidade", oPedido.Quantidade))
        cmdPedidos.Parameters.Add(New SqlParameter("@ViaTransporteID",
oPedido.ViaTransporteID))
        cmdPedidos.Parameters.Add(New SqlParameter("@FormasPagamentoID",
oPedido.FormasPagamentoID))
        cmdPedidos.Parameters.Add(New SqlParameter("@EspecificacaoID",
oPedido.EspecificacaoID))
        Catch ex As Exception
            Transaction.Rollback()
            oConexao.Close()
        Finally
            cmdPedidos.ExecuteNonQuery()
            Transaction.Commit()
            oConexao.Close()
        End Try
    End Sub

```

Listagem 5 – Classe pedido.vb

A classe de pedidos possui o mesmo método de inclusão da classe de clientes, assim como todas as outras classes que possuem informações a serem inseridas no banco de dados.

O *code-behind* (Listagem 6), da tela de cadastro do pedido, possui a mesma estruturação da tela de cadastro dos clientes, porém com algumas validações para os cadastros de *booking* e complemento do pedido, que serão possíveis apenas após o pedido já estar cadastrado no sistema.

```

Private Sub btnSalvar_Click(sender As Object, e As EventArgs) Handles btnSalvar.Click
    Dim oPedido As New Pedido

    With oPedido
        .ClienteID = cmbCliente.SelectedValue
        .DataCarregamento = CDate(dtCarregamento.Text)
        .DataPrevChegada = CDate(dtPrevChegada.Text)
        .EspecificacaoID = cmbEspecificacoes.SelectedValue
        .FormasPagamentoID = cmbFormaPagamento.SelectedValue
        .Quantidade = txtQuantidade.Text
        .Observacoes = txtObservacao.Text
        .ViaTransporteID = cmbViaTransporte.SelectedValue
    End With

    If Not Request("PedidoID") Is Nothing AndAlso txtArmador.Text <> "" Then
        Dim oBooking As New Booking

        With oBooking
            .NomeArmador = txtArmador.Text
            .NomeTransporte = txtNomeTransporte.Text
            .EmpresaAviacao = txtEmpresaAviacao.Text
            .TipoTransporte = cmbViaTransporte.SelectedValue
            .DataEmbarque = CDate(dtEmbarque.Text)
            .DataChegada = CDate(dtChegada.Text)
        End With

        Dim oConexao As SqlConnection = New SqlConnection(RetConnectionString())

        oConexao.Open()

        Dim dtBooking As DataTable = RetDataTable(String.Format("SELECT * FROM Pedidos
INNER JOIN Booking ON Pedidos.BookingID = Booking.BookingID WHERE PedidoID = {0}",
Val(Request("PedidoID"))), oConexao, Transaction)

        oConexao.Close()

        If dtBooking.Rows.Count > 0 Then
            Booking.Alterar(oBooking, Transaction)
        Else
            Booking.Incluir(oBooking, Val(Request("PedidoID")), Transaction)
        End If

        If txtDestino.Text <> "" Then
            Dim oComplemento As New Complemento

            oConexao.Open()

            With oComplemento
                .LocalDestino = txtDestino.Text
                .DataEmbarque = CDate(dtEmbarqueComplemento.Text)
                .PrevisaoChegada = CDate(dtPrevChegadaComplemento.Text)
                .MotivoAtraso = txtMotivoAtraso.Text
            End With
        End If
    End If
End Sub

```

```

        .NumeroTentativa = txtNumTentativas.Text
    End With

    Dim dtComplemento As DataTable = RetDataTable(String.Format("SELECT
ComplementoID FROM PedidoComplemento WHERE PedidoID = {0}", Val(Request("PedidoID"))),
oConexao, Transaction)

    If dtComplemento.Rows.Count > 0 Then
        Complemento.Alterar(oComplemento, Val(Request("PedidoID")), oConexao,
Transaction)
    Else
        Complemento.Incluir(oComplemento, Val(Request("PedidoID")), oConexao,
Transaction)
    End If

    oConexao.Close()
End If

End If

Try
    If Not Request("PedidoID") Is Nothing Then
        Alterar(oPedido, Val(Request("PedidoID")), Transaction)
    Else
        Incluir(oPedido, Transaction)
    End If
Catch ex As Exception
    MsgBox("O registro não foi alterado!", MsgBoxStyle.Information, "Atenção")
Finally
    Response.BufferOutput = True
    Response.Redirect("Pedidos.aspx")
End Try
End Sub

```

Listagem 6 – btnSalvar_click() da classe pedido

A Listagem 7 mostra o método criado para o carregamento do pedido na sua edição. Nesse método é criada a conexão com o banco e que é aberta pela propriedade Open(), criado o comando SQL para buscar as informações, carregado um DataSet com as informações trazidas pela consulta e preenchidos os campos, fazendo uma verificação de já existe um *booking* e um complemento cadastrado.

```

Private Sub CarregaPedido(ByVal nID As Integer)
    Dim oConexao As SqlConnection = New SqlConnection(Conexao.RetConnectionString())

    oConexao.Open()

    Dim oComando As SqlCommand = New SqlCommand(String.Format("SELECT C.ClienteID,
B.BookingID, CP.ComplementoID, E.EspecificacaoID, FormasPagamentoID, Quantidade,
convert(nvarchar(10),DataCarregamento,103) as DataCarregamento,
convert(nvarchar(10),DataPrevChegada,103) as DataPrevChegada, ViaTransporteID,
P.Observacoes, NomeTransporte, NomeArmador, EmpresaAviacao,
convert(nvarchar(10),B.DataEmbarque,103) as DataEmbarque,
convert(nvarchar(10),B.DataChegada,103) as DataChegada,
LocalDestino, convert(nvarchar(10),CP.DataEmbarque,103) as DataEmbarqueComp,
NumeroTentativa, MotivoAtraso FROM Pedidos P
INNER JOIN Clientes C ON C.ClienteID = P.ClienteID
INNER JOIN ClienteEspecificacoes CE ON CE.ClienteID = C.ClienteID AND
CE.EspecificacaoID = P.EspecificacaoID

```

```

"INNER JOIN Especificacoes E ON E.EspecificacaoID = CE.EspecificacaoID
LEFT JOIN Booking B ON B.BookingID = P.BookingID
LEFT JOIN PedidoComplemento PC ON PC.PedidoID = P.PedidoID
LEFT JOIN Complemento CP ON CP.ComplementoID = PC.ComplementoID
WHERE P.PedidoID = {0}", nID), oConexao)

Dim oAdaptador As SqlDataAdapter = New SqlDataAdapter(oComando)

Dim ds As New DataSet
oAdaptador.Fill(ds)

oConexao.Close()

'Carrega Pedido
cmbCliente.SelectedValue = ds.Tables(0).Rows(0)("ClienteID")
cmbEspecificacoes.SelectedValue = ds.Tables(0).Rows(0)("EspecificacaoID")
txtQuantidade.Text = ds.Tables(0).Rows(0)("Quantidade")
cmbFormaPagamento.SelectedValue = ds.Tables(0).Rows(0)("FormasPagamentoID")
txtObservacao.Text = ds.Tables(0).Rows(0)("Observacoes")
dtCarregamento.Text = ds.Tables(0).Rows(0)("DataCarregamento")
dtPrevChegada.Text = ds.Tables(0).Rows(0)("DataPrevChegada")
cmbViaTransporte.SelectedValue = ds.Tables(0).Rows(0)("ViaTransporteID")

'Carrega Booking
If Not IsDBNull(ds.Tables(0).Rows(0)("BookingID")) Then
    txtNomeTransporte.Text = ds.Tables(0).Rows(0)("NomeTransporte")
    txtArmador.Text = ds.Tables(0).Rows(0)("NomeArmador")
    txtEmpresaAviacao.Text = ds.Tables(0).Rows(0)("EmpresaAviacao")
    dtEmbarque.Text = ds.Tables(0).Rows(0)("DataEmbarque")
    dtChegada.Text = ds.Tables(0).Rows(0)("DataChegada")
End If

'Carrega Complemento
If Not IsDBNull(ds.Tables(0).Rows(0)("ComplementoID")) Then
    txtDestino.Text = ds.Tables(0).Rows(0)("LocalDestino")
    txtNumTentativas.Text = ds.Tables(0).Rows(0)("NumeroTentativa")
    dtEmbarqueComplemento.Text = ds.Tables(0).Rows(0)("DataEmbarqueComp")
    dtPrevChegadaComplemento.Text = ds.Tables(0).Rows(0)("DataPrevChegada")
    txtMotivoAtraso.Text = ds.Tables(0).Rows(0)("MotivoAtraso")
End If
End Sub

```

Listagem 7 – Método CarregaPedido da classe pedido

Dentro da tela de listagem dos pedidos (Listagem 8) está o evento de comando das linhas do *grid* utilizado para listagem dos registros, que verificará qual o comando está sendo solicitado, carrega o Id do registro para a variável *nID*, no caso da edição chamamos a tela de cadastro do pedido passando o Id pela *request* e no caso da exclusão executamos o comando *delete* no banco de dados.

```

Protected Sub grd_RowCommand(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewCommandEventArgs) Handles grdPedidos.RowCommand
    If e.CommandName.Equals("EditarRegistro") Then
        Dim nID As Integer = grdPedidos.Rows(e.CommandArgument).Cells(0).Text

        Response.BufferOutput = True
        Response.Redirect("PedidoCadastro.aspx?PedidoID=" & nID)
    ElseIf e.CommandName.Equals("ExcluirRegistro") Then

```

```

Dim oConexao As SqlConnection = New SqlConnection(Conexao.RetConnectionString())

Dim nID As Integer = grdPedidos.Rows(e.CommandArgument).Cells(0).Text

Try
    oConexao.Open()

    Dim oComando As SqlCommand = New SqlCommand("Delete from Pedidos where PedidoID =
" & nID, oConexao)

    oComando.CommandType = CommandType.Text
    oComando.Connection = oConexao

    oComando.ExecuteNonQuery()

Catch ex As Exception
    MsgBox("O registro não foi excluído!", MsgBoxStyle.Information, "Atenção")
Finally
    CarregaGrid()
    oConexao.Close()
End Try
End If
End Sub

```

Listagem 8 – `grd_RowCommand()` da classe `pedido`

Na Listagem 9, é mostrado o arquivo *Extensible Markup Language* (XML) criado para o DataSet utilizado para o carregamento das informações no arquivo com extensão `.rdlc` do relatório de clientes.

```

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="dsPedidos"
    targetNamespace="http://tempuri.org/dsPedidos.xsd"
    elementFormDefault="qualified"
    attributeFormDefault="qualified"
    xmlns="http://tempuri.org/dsPedidos.xsd"
    xmlns:mstns="http://tempuri.org/dsPedidos.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
    xmlns:msprop="urn:schemas-microsoft-com:xml-msprop">
  <xs:annotation>
    <xs:appinfo source="urn:schemas-microsoft-com:xml-msdatasource">
      <DataSource DefaultConnectionIndex="0"
        FunctionsComponentName="QueriesTableAdapter" Modifier="AutoLayout, AnsiClass, Class,
        Public" SchemaSerializationMode="IncludeSchema" xmlns="urn:schemas-microsoft-com:xml-
        msdatasource">
        <Connections />
        <Tables />
        <Sources />
      </DataSource>
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="dsPedidos" msdata:IsDataSet="true"
    msdata:UseCurrentLocale="true" msprop:Generator_UserDSName="dsPedidos"
    msprop:Generator_DataSetName="dsPedidos">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Pedidos" msprop:Generator_UserTableName="Pedidos"
          msprop:Generator_RowDeletedName="PedidosRowDeleted"
          msprop:Generator_RowChangedName="PedidosRowChanged"
          msprop:Generator_RowClassName="PedidosRow"

```

```

msprop:Generator_RowChangingName="PedidosRowChanging"
msprop:Generator_RowEventArgsName="PedidosRowChangeEvent"
msprop:Generator_RowEventHandlerName="PedidosRowChangeEventHandler"
msprop:Generator_TableClassName="PedidosDataTable"
msprop:Generator_TableVarName="tablePedidos"
msprop:Generator_RowDeletingName="PedidosRowDeleting"
msprop:Generator_TablePropName="Pedidos">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PedidoID"
msprop:Generator_UserColumnName="PedidoID"
msprop:Generator_ColumnVarNameInTable="columnPedidoID"
msprop:Generator_ColumnPropNameInRow="PedidoID"
msprop:Generator_ColumnPropNameInTable="PedidoIDColumn" type="xs:int" default="0" />
      <xs:element name="ClienteID"
msprop:Generator_UserColumnName="ClienteID"
msprop:Generator_ColumnVarNameInTable="columnClienteID"
msprop:Generator_ColumnPropNameInRow="ClienteID"
msprop:Generator_ColumnPropNameInTable="ClienteIDColumn" type="xs:int" default="0" />
      <xs:element name="NomeCliente"
msprop:Generator_UserColumnName="NomeCliente"
msprop:Generator_ColumnVarNameInTable="columnNomeCliente"
msprop:Generator_ColumnPropNameInRow="NomeCliente"
msprop:Generator_ColumnPropNameInTable="NomeClienteColumn" type="xs:string"
minOccurs="0" />
      <xs:element name="Produto"
msprop:Generator_UserColumnName="Produto"
msprop:Generator_ColumnVarNameInTable="columnProduto"
msprop:Generator_ColumnPropNameInRow="Produto"
msprop:Generator_ColumnPropNameInTable="ProdutoColumn" type="xs:string" default="0" />
      <xs:element name="Quantidade"
msprop:Generator_UserColumnName="Quantidade"
msprop:Generator_ColumnVarNameInTable="columnQuantidade"
msprop:Generator_ColumnPropNameInRow="Quantidade"
msprop:Generator_ColumnPropNameInTable="QuantidadeColumn" type="xs:int" default="0" />
      <xs:element name="DataCarregamento"
msprop:Generator_UserColumnName="DataCarregamento"
msprop:Generator_ColumnVarNameInTable="columnDataCarregamento"
msprop:Generator_ColumnPropNameInRow="DataCarregamento"
msprop:Generator_ColumnPropNameInTable="DataCarregamentoColumn" type="xs:string"
default="0" />
      <xs:element name="DataPrevChegada"
msprop:Generator_UserColumnName="DataPrevChegada"
msprop:Generator_ColumnVarNameInTable="columnDataPrevChegada"
msprop:Generator_ColumnPropNameInRow="DataPrevChegada"
msprop:Generator_ColumnPropNameInTable="DataPrevChegadaColumn" type="xs:string"
default="0" />
      <xs:element name="ViaTransporte"
msprop:Generator_UserColumnName="ViaTransporte"
msprop:Generator_ColumnVarNameInTable="columnViaTransporte"
msprop:Generator_ColumnPropNameInRow="ViaTransporte"
msprop:Generator_ColumnPropNameInTable="ViaTransporteColumn" type="xs:string"
default="0" />
      <xs:element name="Observacoes"
msprop:Generator_UserColumnName="Observacoes"
msprop:Generator_ColumnVarNameInTable="columnObservacoes"
msprop:Generator_ColumnPropNameInRow="Observacoes"
msprop:Generator_ColumnPropNameInTable="ObservacoesColumn" type="xs:string"
default="0" />
      <xs:element name="NomeTransporte"
msprop:Generator_UserColumnName="NomeTransporte"
msprop:Generator_ColumnVarNameInTable="columnNomeTransporte"

```

```

msprop:Generator_ColumnPropNameInRow="NomeTransporte"
msprop:Generator_ColumnPropNameInTable="NomeTransporteColumn" type="xs:string"
default="0" />
    <xs:element name="NomeArmador"
msprop:Generator_UserColumnName="NomeArmador"
msprop:Generator_ColumnVarNameInTable="columnNomeArmador"
msprop:Generator_ColumnPropNameInRow="NomeArmador"
msprop:Generator_ColumnPropNameInTable="NomeArmadorColumn" type="xs:string"
default="0" />
    <xs:element name="EmpresaAviacao"
msprop:Generator_UserColumnName="EmpresaAviacao"
msprop:Generator_ColumnVarNameInTable="columnEmpresaAviacao"
msprop:Generator_ColumnPropNameInRow="EmpresaAviacao"
msprop:Generator_ColumnPropNameInTable="EmpresaAviacaoColumn" type="xs:string"
default="0" />
    <xs:element name="DataEmbarque"
msprop:Generator_UserColumnName="DataEmbarque"
msprop:Generator_ColumnVarNameInTable="columnDataEmbarque"
msprop:Generator_ColumnPropNameInRow="DataEmbarque"
msprop:Generator_ColumnPropNameInTable="DataEmbarqueColumn" type="xs:string"
default="0" />
    <xs:element name="DataChegada"
msprop:Generator_UserColumnName="DataChegada"
msprop:Generator_ColumnVarNameInTable="columnDataChegada"
msprop:Generator_ColumnPropNameInRow="DataChegada"
msprop:Generator_ColumnPropNameInTable="DataChegadaColumn" type="xs:string"
default="0" />
    <xs:element name="LocalDestino"
msprop:Generator_UserColumnName="LocalDestino"
msprop:Generator_ColumnVarNameInTable="columnLocalDestino"
msprop:Generator_ColumnPropNameInRow="LocalDestino"
msprop:Generator_ColumnPropNameInTable="LocalDestinoColumn" type="xs:string"
default="0" />
    <xs:element name="DataEmbarqueComp"
msprop:Generator_UserColumnName="DataEmbarqueComp"
msprop:Generator_ColumnVarNameInTable="columnDataEmbarqueComp"
msprop:Generator_ColumnPropNameInRow="DataEmbarqueComp"
msprop:Generator_ColumnPropNameInTable="DataEmbarqueCompColumn" type="xs:string"
default="0" />
    <xs:element name="NumeroTentativa"
msprop:Generator_UserColumnName="NumeroTentativa"
msprop:Generator_ColumnVarNameInTable="columnNumeroTentativa"
msprop:Generator_ColumnPropNameInRow="NumeroTentativa"
msprop:Generator_ColumnPropNameInTable="NumeroTentativaColumn" type="xs:int"
default="0" />
    <xs:element name="MotivoAtraso"
msprop:Generator_UserColumnName="MotivoAtraso"
msprop:Generator_ColumnVarNameInTable="columnMotivoAtraso"
msprop:Generator_ColumnPropNameInRow="MotivoAtraso"
msprop:Generator_ColumnPropNameInTable="MotivoAtrasoColumn" type="xs:string"
default="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>

```

Listagem 9 – XML do DataSet para o relatório de pedidos

Na Listagem 10, é mostrada a consulta realizada no relatório de pedidos, verificando ao final da consulta se foi filtrado por algum cliente e qual o cliente.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

    SqlDataSource1.SelectCommand = "SELECT P.PedidoID, C.ClienteID, C.Nome as
NomeCliente, cast(Nomenclatura as NVARCHAR(MAX)) + ' - ' + CASE Tipo WHEN 1 THEN
'Suína; ' ELSE 'Bovina; ' END + 'Mesh: ' + cast(Mesh AS NVARCHAR(MAX)) + ' / Bloom: '
+ cast(Bloom as NVARCHAR(MAX)) as Produto, cast(Quantidade as NVARCHAR(MAX)) + ' Kg'
as Quantidade, convert(nvarchar(10),DataCarregamento,103) as DataCarregamento,
convert(nvarchar(10),DataPrevChegada,103) as DataPrevChegada, CASE ViaTransporteID
WHEN 1 THEN 'Aéreo' WHEN 2 THEN 'Marítimo' ELSE 'Terrestre' END AS ViaTransporte,
p.Observacoes, NomeTransporte, NomeArmador, EmpresaAviacao,
convert(nvarchar(10),B.DataEmbarque,103) as DataEmbarque,
convert(nvarchar(10),B.DataChegada,103) as DataChegada, LocalDestino,
convert(nvarchar(10),CP.DataEmbarque,103) as DataEmbarqueComp, NumeroTentativa,
MotivoAtraso FROM Pedidos P INNER JOIN Clientes C ON C.ClienteID = P.ClienteID INNER
JOIN ClienteEspecificacoes CE ON CE.ClienteID = C.ClienteID AND CE.EspecificacaoID =
P.EspecificacaoID INNER JOIN Especificacoes E ON E.EspecificacaoID =
CE.EspecificacaoID LEFT JOIN Booking B ON B.BookingID = P.BookingID LEFT JOIN
PedidoComplemento PC ON PC.PedidoID = P.PedidoID
LEFT JOIN Complemento CP ON CP.ComplementoID = PC.ComplementoID" &
If(Not Request("ClienteID") Is Nothing, " WHERE C.ClienteID = " &
Val(Request("ClienteID")), "")

End Sub
```

Listagem 10 – Page_Load do relatório de pedidos

5 CONCLUSÃO

Foram apresentadas a modelagem do sistema e a implementação de suas funcionalidades. Todas as funcionalidades apresentadas na modelagem foram implementadas. Ajuste serão necessários a partir da utilização do sistema pelos usuários.

Para gerenciar a execução da aplicação *web* foi utilizado o IIS (*Internet Information Services*), um servidor *web* criado pela Microsoft. Para a implementação das funcionalidades foi utilizado VB .NET e JavaScript. E para a implementação da interface, foi utilizado o *framework front-end* Bootstrap, juntamente com ASP.NET, HTML5 e CSS. Essas tecnologias apresentam recursos que facilitam o desenvolvimento da aplicação e permitem desenvolver interface com funcionalidades que visam auxiliar na interação do usuário com o sistema.

Neste trabalho foi possível perceber que as tecnologias voltadas ao desenvolvimento de aplicações *web* desde frameworks voltados à criação de sua interface até a linguagem de programação utilizada em suas funcionalidades, apesar de apresentar um nível de dificuldade maior que em aplicações *desktop*, possuem resultados mais eficazes e mais rápidos.

A linguagem VB .NET tem como principal vantagem, muitas funções e funcionalidades prontas para serem utilizadas.

O *framework* Bootstrap tem como principal vantagem, a possibilidade de criar interfaces sem a necessidade de utilização de CSS, apenas trabalhando com suas mais diversas classes é possível desenvolver uma interface com o usuário que atenda a requisitos e padrões de usabilidade.

Como sequência do trabalho, a implementação será finalizada e serão aplicados os testes para verificação do atendimento das funcionalidades. Será realizada a identificação de erros de codificação e de regras de negócio que não tenham sido detectados nos testes unitários realizados durante a implementação.

REFERÊNCIAS

- AGUIAR, G. A. O. **Comportamento das exportações e o crescimento econômico do Estado do Ceará de 1996-2006**. Fortaleza: Universidade de Fortaleza, 2008. 17 f. Disponível em: <<http://www.corecon-ce.org.br/GilsonAndre.pdf>>. Acesso em: 21 jun. 2009
- ALMEIDA, Poliana F.; VANALLE, Rosângela M.; SANTANA, José Carlos C. Produção de gelatina: uma perspectiva competitiva para a cadeia produtiva de frango de corte. **Produto & Produção**, v. 13 n. 2, p. 22-39, jun. 2012.
- ALVES, Sandra Geres T.; PRUDÊNCIO-FERREIRA, Sandra Helena Propriedades funcionais de material colagenoso de pés de frango. **ALAN**, Venezuela, v. 52, n. 3, p. 289-293, 2002.
- AMATO, Fábio. **Portos escoam 95% das exportações do país, mas seguem com gargalos**, 2012. Disponível em: <<http://g1.globo.com/economia/noticia/2012/12/portos-escoam-95-das-exportacoes-do-pais-mas-seguem-com-gargalos.html>>. Acesso em: 03 mar. 2015.
- BAIN E COMPANY; GAS ENERGY. **Potencial de diversificação da indústria química Brasileira. Relatório 4 – Aditivos alimentícios**. 2014. Disponível em: <<http://www.bndes.gov.br>>. Acesso em: 5 nov. 2015.
- CHO, Suengmok et al. Extracting optimization and physical properties of yellowfin tuna (*Thunnus albacares*) skin gelatin compared to mammalian gelatins. **Food Hydrocolloids**, v. 19, p. 221-229, 2005.
- COELHO, Leandro Callegari. Gestão da Cadeia de Suprimentos – conceitos, tendências e ideias para melhoria. **Revista Today Logistics**, n. 51, agosto de 2010. Disponível em: <<http://www.logisticadescomplicada.com/gestao-da-cadeia-de-suprimentos-%E2%80%93-conceitos-tendencias-e-ideias-para-melhoria/>>. Acesso em: 28 mar. 2015.
- COHEN, Marcel; STATHIS, Kostas. Strategic change stemming from ecommerce: implications of multi-agent systems on the supply chain, **Strategic Change**, v. 10, p. 139-149, 2007.
- COLLINS, John; TSVETOVAT, Maksim; BILOT, Corey; SUNDARESWARA, Rashmi; LEE, Tim; GINI, Maria; MOBASHER, Bamshad. **A framework for mixed initiative agent-based contracting**, In: First IAC Workshop on Internet Based Negotiation Technologies, 2008
- FENG, Weibin; YANLI, Xiao. **The study on international trade negotiation based on supply chain management**. 2009 International Conference on Test and Measurement (ICTM 2009), p. 204-207.
- GIMÉNEZ, Begoña; et al. Use of lactic acid for extraction of fish skin gelatin. **Food Hydrocolloids**, v. 19, p. 941–950, 2005.

HAUG, Ingvild J.; DRAGET, Kurt I.; SMIDSRÖD, Olav. Physical and rheological properties of fish gelatin compared to mammalian gelatin. **Food Hydrocolloids**, v. 8, p. 203–213, 2004.

ISAAC, Fabio L. **Brasil produz 11 % da gelatina do mundo**. Disponível em: <<https://www.scotconsultoria.com.br/noticias/todas-noticias/433/brasil-produz-11-da-gelatina-do-mundo.htm>>. Acesso em: 29 mar. 2015.

MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA E COMÉRCIO. **Exportação**. Disponível em: <<http://www.desenvolvimento.gov.br/sitio/interna/interna.php?area=5&menu=245>>. Acesso em: 18 de ago. 2015.

MOODIE, Douglas R.; BOBROWSHKI, Paul. Due date demand management: negotiating the trade-off between price and delivery. **International Journal of Production Research**, v. 37, n. 5, p. 997-1021, 2008.

MUNDURUCA, DANILO F. V. **Comércio exterior como estratégia de crescimento econômico: uma proposta de priorização de produtos exportáveis para a economia sergipana**. Dissertação (mestrado) Universidade Federal de Sergipe. Mestrado profissional em desenvolvimento regional e gestão de empreendimentos locais, 2010. Disponível em: <http://bdtd.ufs.br/tde_busca/arquivo.php?codArquivo=359>. Acesso em: 28 mar. 2015.

OLIVEIRA, Marcos Berberick; LONGO, Orlando Celso. **Gestão da cadeia de suprimentos**. IV Congresso Nacional de Excelência em Gestão, p. 1-17, 2008.

SCOT CONSULTORIA. **Paulo Reimann – Gelita South América**. Disponível em: <<https://www.scotconsultoria.com.br/imprimir/noticias/16842>>. Acesso em: 29 mar. 2015.

SILVA, Jorge Antonio Santos. O modelo da base econômica e o grau de endogeneização do desenvolvimento turístico. **Revista de Desenvolvimento Econômico**, Salvador, ano 6, n.10, jul. de 2004. Disponível em: <<http://www.revistas.unifacs.br/index.php/rde/article/viewFile/105/109>>. Acesso em: 27 mar. 2015.

SOUZA, Gleim Dias de, CARVALHO, Maria do Socorro M. V. de & LIBOREIRO, Manuel Alejandro Martínez. Gestão da Cadeia de Suprimentos Integrada à Tecnologia da Informação. **Revista de Administração Pública**, v. 40, n 4. Rio de Janeiro. Julho e agosto de 2006.

SOUZA, Nali de J. Exportações e crescimento econômico do Rio Grande do Sul, 1951-2001. **Ensaio FEE**, Porto Alegre, v. 23, n. especial, p. 565-602, 2002. Disponível em: <http://www.fee.rs.gov.br/sitefee/download/eeg/1/mesa_1_%20souza.pdf>. Acesso em: 18 abr. 2015.

YANG, Hongshun; WANG, Yifen; JIANG, Mingkang; OH, Jun-Hyun; HERRING, Josh; ZHOU, Peng. 2-Step optimization of the extraction and subsequent physical properties of channel catfish (*Ictalurus punctatus*) skin gelatin. **Food Chemistry and Toxicology**, v. 72, n.4, p.188-195, 2007. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/17995759>>. Acesso em: 26 jun. 2015.