

LARISSA PIOVEZANI

O USO DE TECNOLOGIA MICROSOFT. NET NO
DESENVOLVIMENTO DE APLICAÇÕES DE COMÉRCIO
ELETRÔNICO COM INTERFACE RICA.

Assis

2009

O USO DE TECNOLOGIA MICROSOFT. NET NO
DESENVOLVIMENTO DE APLICAÇÕES DE COMÉRCIO
ELETRÔNICO COM INTERFACE RICA.

LARISSA PIOVEZANI

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Almir Rogério Camolesi

Analisador (1): Douglas Sanches da Cunha

Área de Concentração: Aplicações Ricas com Silverlight

ASSIS
2009

DEDICATÓRIA

*Dedico este trabalho aos meus pais, as
minhas irmãs, a minha avó, e a todos que me
apoiaram.*

AGRADECIMENTOS

A Deus, pela minha vida e por todas as oportunidades nela concedidas.

Ao Orientador Prof. Dr. Almir Rogério Camolesi, pela paciência e apoio durante todo o trabalho.

A minha mãe, Marize Ferreira, pelo amor, dedicação e incentivo.

Ao meu pai, Silvano Piovezani Junior, por permitir que eu realizasse meus sonhos.

A minha avó, Yvonny Rodrigues Ferreira, pelos cuidados e apoio.

A minha prima Milene, pela ajuda no trabalho e pelo incentivo constante.

A todas as “xuxus” pelos momentos de descontração em meio a tanto nervosismo e ansiedade, pelos conselhos e pela amizade sincera.

Aos meus amigos - Leonardo, Rafael, Rodrigo, Simone, Vinícius e Wilson, por terem participado dos melhores anos da minha vida, compartilhando melhores e piores momentos e transformando a rotina noturna da faculdade em algo inesquecível.

A todos os professores que fizeram parte do meu crescimento profissional, me ensinando uma parte de tudo que aprenderam durante suas vidas.

RESUMO

Devido ao avanço da internet e seu constante e crescente uso mundial, o comércio viu a necessidade de levar os seus produtos para o meio eletrônico, para buscar novos clientes e atrair mais ainda os já conquistados. Atualmente muitas empresas implantam lojas virtuais para oferecer os seus produtos. Para facilitar este trabalho é proposta uma ferramenta para o gerenciamento de lojas de comércio eletrônico usando os conceitos de interfaces ricas e a tecnologia ASP .Net e Silverlight.

Palavras-chave: Lojas Virtuais, Aplicações Ricas, ASP .NET, Microsoft Silverlight.

ABSTRACT

Because of the internet advance and its constant and increasing world using, the business trade saw the need of taking its products out to the electronic means, thus, they may get new customers and attract more than the ones who they have already had. Nowadays, a lot of companies have web stores where their products are offered. To make this job easier, a tool is proposed for the management of e virtual stores using concepts of Rich Internet Applications and the ASP .Net and Silverlight technology.

Keywords: Web stores, Rich Internet Applications, ASP .NET, Microsoft Silverlight.

LISTA DE ILUSTRAÇÕES

Figura 1 – Proposta de desenvolvimento do Trabalho	14
Figura 2 – Aplicações Ricas para Internet	17
Figura 3 – Arquitetura .NET para Silverlight	26
Figura 4 – Arquitetura Silverlight	27
Figura 5 – Comunicação Silverlight	28
Figura 6 – Módulos Aplicação	31
Figura 7 – Diagrama de Caso de Uso	34
Figura 8 – Diagrama de Classe	36
Figura 9 – Diagrama de Sequência – Acesso Público	37
Figura 10 – Diagrama de Sequência – Cadastro Produto, curso normal	39
Figura 11 – Diagrama de Sequência – Adicionar itens ao carrinho de compras, curso normal	40
Figura 12 – Diagrama de Sequência – Finalizando Compra	41
Figura 13 – Página Inicial	43
Figura 14 – Descrição Menu	44
Figura 15 – Produtos	44
Figura 16 – Tela de Acesso	45
Figura 17 – Criar Loja – Tela visualizada após a autenticação	46
Figura 18 – Minha Loja - Tela visualizada após a autenticação	46
Figura 19 – Tela de Contato	47
Figura 20 – Código XAML da tela inicial – Camada de Apresentação	48
Figura 21 – Classe que contém um produto	49
Figura 22 – Classe que contém uma lista de produtos	50
Figura 23 – Camada de Negócio	50
Figura 24 – Camada de Dados	51

LISTA DE TABELAS

Tabela 1 – Lista de Eventos do Aplicativo.....	33
--	----

LISTA DE ABREVIATURAS E SIGLAS

ASP	Active Server Pages
B2C	Business-to-Consumer
CLR	Common Language Runtime
CSS	Cascading Style Sheets
CTS	Common Type System
GonLo	Gerenciador Online de Lojas Virtuais
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IIS	Internet Information Services
JIT	Just-in-time compilation
LINQ	Language Integration Query
MSIL	Microsoft Intermediate Language
RIA	Rich Internet Applications
UI	Users Interface
UX	User eXperience
WCF	Windows Communication Foundation
WPF	Windows Presentation Foundation
XAML	eXtensible Application Markup Language
XML	eXtensible Markup Language

SUMÁRIO

1. INTRODUÇÃO	12
2. CONCEITOS	16
2.1. RICH INTERNET APPLICATION - RIA.....	16
2.1.1. Histórico de Rich Internet Application	16
2.1.2. Conceito	17
2.1.3. Funcionamento.....	18
2.1.4. Vantagens da utilização de RIA	18
2.1.5. Desvantagens da utilização de RIA	19
2.1.6. Plataforma disponível para desenvolvimento – Microsoft Silverlight	19
2.1.7. Aplicações	20
3. PLATAFORMAS, TECNOLOGIAS E FERRAMENTAS MICROSOFT	21
3.1.ASP .NET	21
3.1.1. Histórico.....	21
3.1.2. Conceitos	22
3.2. VISUAL TEAM STUDIO 2008.	22
3.3. MICROSOFT SILVERLIGHT.....	24
3.3.1. Conceitos	24
3.3.2. Arquitetura do Silverlight	25
3.3.3. Ferramentas para desenvolvimento de Silverlight.....	28
3.4. WINDOWS PRESENTATION FOUNDATION (WPF)	29
4. GONLO – GERENCIADOR ONLINE DE LOJAS VIRTUAIS	30
4.1.COMÉRCIO ELETRÔNICO.....	30
4.2.A ARQUITETURA DA APLICAÇÃO	31
4.3.MODELAGEM	32
4.3.1. Diagrama de Caso de Uso	33
4.3.2. Diagrama de Classe	35
4.3.3. Diagrama de Sequência.....	36
4.3.4. Estrutura Tecnológica da Aplicação.....	42

4.4.IMPLEMENTAÇÃO DO TRABALHO.....	42
4.4.1. Interface do Aplicativo	42
4.4.2. Programação do Aplicativo	47
5. CONCLUSÃO	52
6. REFERÊNCIAS BIBLIOGRÁFICAS	53

1. INTRODUÇÃO

As lojas virtuais surgiram pouco tempo depois da criação da Web, em meados dos anos 90. Esse tipo de loja não necessita da presença física de compradores, vendedores e tampouco da própria mercadoria no ato da compra, sendo esta entregue alguns dias depois da transação efetuada. A loja virtual é um *web site* que vende produtos ou serviços online, ou seja, um site de e-commerce – transações comerciais feitas com o auxílio de um equipamento eletrônico, onde o cliente escolhe o produto, coloca no carrinho de compras e efetua o pagamento no caixa, numa transação totalmente online (FELIPINI, 1999).

O papel desempenhado por uma loja virtual é de extrema importância para a estratégia de qualquer negócio na internet. É através da loja virtual que o cliente será apresentado ao produto, poderá comparar os preços, se sentirá motivado ou não para comprá-lo. Considerando que os produtos oferecidos em uma loja virtual terão um preço justo, a decisão de compra vai depender da qualidade da loja virtual em transmitir as informações, da facilidade de acesso e navegação do usuário e da confiabilidade que a loja virtual terá em concretizar suas transações (FELIPINI, 1999).

Atualmente as lojas virtuais não apresentam todos os produtos de uma forma elegante. Para o usuário efetuar a compra, primeiro deve ser cadastrado, caso contrário deve navegar em outras páginas para efetuar o cadastramento e confirmações, desviando-se a atenção dos produtos que tinha se interessado. Se o usuário adiciona um produto no carrinho de compra e quer excluí-lo da lista deve acessar a página de cancelamentos e depois retornar para a página principal, tirando-o novamente da página do produto de interesse.

As lojas virtuais, em geral, estão preocupadas em mostrar os produtos para a venda, deixando de se preocupar com a usabilidade e a riqueza dos detalhes visuais desenvolvidos, detalhes estes que atraem tanto os usuários mais leigos quanto os com um grau de conhecimento maior.

Com o surgimento de aplicações ricas e da tecnologia da Microsoft, o Silverlight (DAYLEY, 2008) (KONO, 2008), as aplicações podem ter um acesso mais rápido e com um visual melhor, deixando o usuário entretido não apenas com o que ele quer comprar, mas também com o que ele vê. Mesclando um visual rico com a total usabilidade do usuário, as lojas virtuais poderiam atrair muito mais clientes, que buscariam a facilidade e a rapidez nas compras.

Muitos usuários leigos deixam de comprar em lojas virtuais devido à dificuldade de navegação. O desenvolvimento de uma aplicação para gerenciamento de comércio eletrônico coloca em prática os estudos sobre aplicações ricas, usabilidade e Microsoft Silverlight, levando em consideração as necessidades práticas e visuais dos usuários em geral.

Perante a dificuldade de navegação dos usuários em lojas virtuais, surgiu à necessidade deste trabalho criar uma aplicação, baseado em aplicações ricas para a internet e considerando a maior usabilidade dos usuários. A aplicação será criada para criação e gerenciamento de lojas virtuais, visando facilitar a navegação do administrador, do usuário e do público em uma loja virtual, transformando o visual, não amigável, em algo agradável aos olhos dos usuários através da tecnologia Microsoft Silverlight e de aplicações ricas para internet.

Os objetivos deste trabalho são a criação de uma aplicação para a criação de *web sites* de comércio eletrônico, visando facilitar a utilização do mesmo pelos diversos tipos de usuários e contribuindo para a criação de lojas virtuais com interfaces ricas e total usabilidade dos clientes finais e o aumento do uso deste meio de comércio, reformulando o visual atualmente apresentado. Com isso será apresentado Aplicações Ricas para Internet e a tecnologia Microsoft, e criado documentos sobre o Silverlight em português, pois contém a maior parte da documentação em inglês. A figura 1 mostra graficamente o objetivo do trabalho.

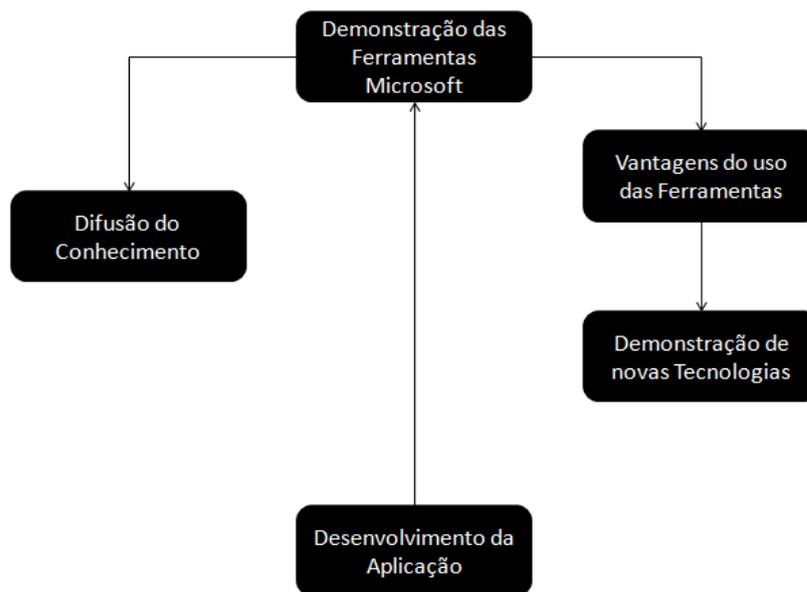


Figura 1. Proposta de Desenvolvimento do Trabalho (PONTES, 2008, p. 12)

O trabalho consiste em desenvolver uma aplicação onde todo o conceito estudado sobre as novas tecnologias são apresentados utilizando as novas ferramentas Microsoft e demonstrando suas vantagens. Isso fará a difusão de todo o conhecimento adquirido durante o trabalho.

Sendo assim, este trabalho é dividido da seguinte forma:

O capítulo 1 apresenta os objetivos, a justificativa e a motivação para o trabalho, esta introdução.

O capítulo 2 abordará conceitos, o histórico de Aplicações Ricas para Internet, suas vantagens e desvantagens e a tecnologia Microsoft Silverlight para o desenvolvimento dessas aplicações. Abordará também sobre Aplicações e suas características.

As tecnologias escolhidas, ASP .Net e Silverlight, para o desenvolvimento de uma aplicação para o gerenciamento de *web sites* de comércio eletrônico e a suíte de desenvolvimento serão abordadas no capítulo 3.

A arquitetura da ferramenta para criação de sites de comércio eletrônico utilizando conceito de aplicações ricas é tratada no quarto capítulo.

As conclusões e contribuições futuras serão apresentadas no capítulo 5.

Ficam disponíveis, por fim, as referências bibliográficas de todo o contexto do trabalho desenvolvido.

2. CONCEITOS

Este capítulo apresenta as tecnologias RIA - Aplicações Ricas para Internet e Aplicações. Descreve seu histórico, conceito, funcionamento, as vantagens e desvantagens de utilização e uma das plataformas de desenvolvimento disponíveis – Microsoft Silverlight.

2.1. RICH INTERNET APPLICATION - RIA

2.1.1. Histórico de Rich Internet Application

Em meados dos anos 90 foi criado um novo modelo de aplicações, Aplicações de Internet, que foca em um desenvolvimento de baixo custo e em uma aplicação mais rápida. Este modelo é baseado em páginas HTML (*HyperText Markup Language*) e nos browsers, servidores de aplicações cuja função é criar e enviar as páginas aos navegadores de Internet. A navegação fornecida ao cliente é de forma estática, devido a isso toda interação do cliente com a página é enviada ao servidor para processar a requisição, validar e consultar e devolver a página ao cliente com todas as informações devidamente atualizadas (ALLAIRE, 2002).

A Macromedia em 2002 introduziu o termo RIA - Aplicações Ricas para Internet (*Rich Internet Application*), este conceito apresentou outros nomes anteriormente pela Microsoft em 1998, como *Remote Scripting*, e pela *Forrester Research* em 2001, como *X Internet*. Este conceito baseia-se no modelo de Aplicações de Internet, buscando as melhores funcionalidades de uma Aplicação Desktop e o baixo custo de desenvolvimento de Aplicações Web, juntamente com conceitos de interatividade e tecnologias de comunicação (ALLAIRE, 2002).

As Aplicações Ricas para Internet superam as limitações do potencial que caracteriza a rede com a entrega de experiências de navegação mais efetivas e uma maior interatividade e riqueza nas interfaces. Significa o uso da tecnologia da

informação para unir o melhor da *web*, ao melhor do seu *desktop* e ao melhor das comunicações. É, em suma, uma experiência interativa muito mais rica para o usuário (ALLAIRE, 2002) (DUHL, 2003).

O conceito tradicional de internet impede o tipo de navegação *drag-and-drop*, desfazer e muito mais, pois, normalmente, é necessário clicar em um *link* para ver alguma opção em alguma página, depois clicar em outro, em outro, e assim por diante (DUHL, 2003).

2.1.2. Conceito

RIA ou *Rich Internet Application* utiliza o mesmo conceito arquitetônico de Aplicações de Internet e permite uma experiência de navegação semelhante ao ambiente *desktop*. As aplicações são executadas no lado do cliente, sendo enviadas ao servidor somente as consultas complexas, como o acesso ao banco de dados, tornando a navegação mais rápida devido à diminuição no tráfego de rede e aumentando a satisfação do usuário. Seus recursos interativos são comparados as interações Desktops (DUHL, 2003).

Aplicações Ricas para Internet são aplicações que contém características e funcionalidades de uma aplicação *desktop* tradicional (KONO, 2008).

A figura 2 ilustra as Aplicações Ricas para internet.

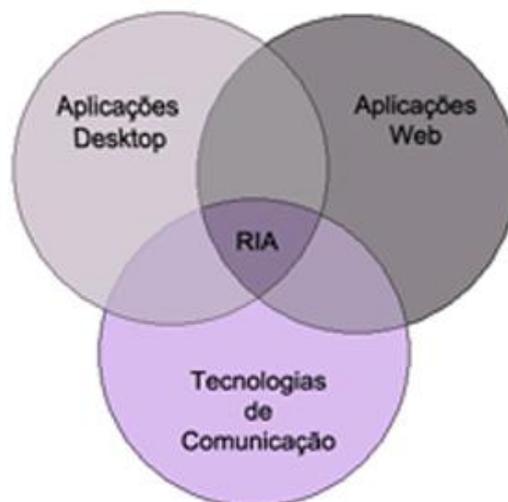


Figura 2. Aplicações Ricas para Internet (PONTES, 2008, p. 15)

2.1.3. Funcionamento

As aplicações Ricas para Internet são implementadas no servidor e utilizam a tecnologia *Rich Client* para fornecer um ambiente dinâmico, com a interatividade e sofisticação de aplicações *Desktop*. Esta tecnologia tem a capacidade de hospedagem de aplicativos compilados no lado do servidor e recebidos como arquivos através de HTTP (*Hypertext Transfer Protocol*). No lado do cliente os aplicativos conectam-se de volta aos *Back-Ends* de servidores de aplicativos existentes, por meio de uma arquitetura assíncrona de Cliente/Servidor que oferece segurança, estabilidade e que é bem adaptada ao novo modelo orientado a serviços que está sendo promovido pela adoção de serviços *web*. No modelo RIA as camadas de negócio e apresentação são completamente dissociadas (ALLAIRE, 2002) (DUHL, 2003).

As Aplicações ricas para internet podem ser executadas tanto em *browsers* como em dispositivos, com ela é possível a criação de aplicativos que podem ser disponibilizados em diversas plataformas de conexão à internet. Com a tecnologia *Rich Client* é possível utilizar elementos gráficos móveis, vídeos, áudio e formulários complexos e constitui um ambiente mais sólido para criação de interfaces dos usuários (DUHL, 2003).

2.1.4. Vantagens da utilização de RIA

Entre as vantagens da utilização de Aplicações Ricas para Internet está a riqueza da interface oferecida ao usuário, uma interatividade próxima ao ambiente *Desktop*, o equilíbrio entre Cliente/Servidor, ou seja, o processamento das informações é realizado no cliente e posteriormente enviado ao servidor somente as requisições necessárias, isso possibilita que o servidor possa atender outras requisições de outros clientes ao mesmo tempo, o que torna a navegação do usuário mais rápida (DUHL, 2003) (MORITZ, 2008).

Outra vantagem encontrada é a comunicação assíncrona entre o cliente e o servidor, com isso quando o usuário executa uma ação na interface, como um clique

em um botão, por exemplo, não é necessário esperar por uma resposta do servidor, pois a partir de uma solicitação são carregados antecipadamente os dados para que a resposta posterior seja acelerada. Isto reduz o tráfego na rede e possibilita uma agilidade na navegação (DUHL, 2003).

2.1.5. Desvantagens da utilização de RIA

Uma das desvantagens visíveis ao cliente é o tempo de carregamento da aplicação. Embora elas não necessitem instalação devem ser baixadas do servidor para o cliente e dependendo do tamanho ou tipo de solicitação o carregamento do script pode ser longo. Se o usuário desativar a execução de scripts em seu navegador as Aplicações Ricas para Internet podem não funcionar corretamente (DUHL, 2003).

Infelizmente os sites de buscas atualmente são incapazes de indexar os textos de uma Aplicação Rica (DUHL, 2003).

Para uma aplicação de comércio eletrônico a desvantagem significativa seria a não indexação dos textos de uma Aplicação Rica, para sanar este problema o aplicativo fará distinção de imagens e textos, anexando comentários a todas as imagens.

2.1.6. Plataforma disponível para desenvolvimento – Microsoft Silverlight

Para executar aplicações Silverlight no lado do cliente nos sistemas operacionais Windows é necessário a instalação de complemento cujo tamanho é em torno de 2 Mb, o Silverlight Runtime (<http://www.microsoft.com/silverlight/get-started/install/default.aspx>). No sistema operacional Linux é necessário a instalação do Moonlight (<http://www.mono-project.com/Moonlight>), que já possui sua versão inicial e está em desenvolvimento pelo projeto Mono, que é financiado pela Novell e pela própria Microsoft. Se houver a necessidade, é possível utilizar JavaScript para controlar alguns objetos no lado cliente (MSDN) (DURÃES, 2008).

De acordo com o MSDN,

Microsoft Silverlight é uma implementação do framework .NET independente de plataforma para construir e exibir a nova geração de

experiência em mídia e RIA para a Web. Silverlight unifica as capacidades do servidor, da Web, do Desktop, do código gerenciado e linguagens dinâmicas, a declaração e programação tradicional e a força do WPF (*Windows Presentation Foundation*).

Pode ser considerado um subproduto do WPF (*Windows Presentation Foundation*), pois utiliza o XAML, *eXtensible Application Markup Language*, linguagem usada para criação de interfaces e usuário de forma simples e rápida. (DURÃES, 2008).

“O .NET framework é um modelo de programação de código gerenciado da Microsoft para criar aplicativos em clientes, servidores e dispositivos móveis ou incorporados do Windows” (MSDN).

O Silverlight trabalha com um novo conceito de recursos visuais, como som e vídeo e animações e conta com a praticidade, agilidade, usabilidade e intuitividade de navegação, ou seja, a Experiência do Usuário (*UX – User eXperience*). Com essa tecnologia o desenvolvedor utiliza o conhecimento em .NET (.NET 2.0, 3.0 e 3.5) e de várias linguagens para o desenvolvimento de aplicativos, como C# e Visual Basic, além de não perder a capacidade de indexação de informações contidas na aplicação, este o ponto forte na Web 3.0 (*SEO – Search Engine Optimization*) (KONO, 2008).

Este assunto será abordado com mais detalhes no terceiro capítulo, onde são apresentadas as ferramentas escolhidas para o desenvolvimento da aplicação.

2.1.7. Aplicações

Software aplicativo, ou aplicação é um programa de computador que desempenha tarefas relacionadas ao processamento de dados, tendo como foco principal o usuário. As aplicações surgiram para facilitar o uso dos computadores pelo público.

Existem diversos tipos de aplicações, em versões para *web* e para *desktop*. Como dito anteriormente, a proposta do trabalho é a criação de uma aplicação baseada em aplicações ricas para internet (DURÃES, 2008).

3. PLATAFORMAS, TECNOLOGIAS E FERRAMENTAS MICROSOFT

Este capítulo apresenta as tecnologias Microsoft escolhidas para a criação da ferramenta de desenvolvimento de aplicações para comércio eletrônico.

3.1. ASP .NET

3.1.1. Histórico

A tecnologia ASP (*Active Server Pages*), da Microsoft, é focada no desenvolvimento de páginas dinâmicas e surgiu com o crescimento da web após a construção de páginas *web* tornarem-se popular com o padrão HTML (*HyperText Markup Language*), onde as páginas eram estáticas e não permitiam interação do usuário com a aplicação. O padrão da tecnologia ASP era um modelo baseado em *script*, na maioria das vezes implementado em VBScript, e hospedado em servidores web IIS (*Internet Information Services*) (DURÃES, 2008).

Surgiu então a plataforma Microsoft .NET Framework, uma nova estratégia no desenvolvimento de softwares usando plataforma Microsoft. Para Durães “.NET é um software para conectar informações, pessoas e dispositivos”, três pontos chaves são sua base: a integração, fazer mais com menos e funcionar com simplicidade (DURÃES, 2008) (MSDN).

O .NET possui um modelo baseado no conceito de máquina virtual, o CLR (*Common Language Runtime*). Com essa plataforma passou a se desenvolver aplicações para .NET Framework e não mais para o Windows diretamente. Com esse modelo qualquer plataforma que possua uma máquina virtual do .NET Framework roda aplicações baseadas em .NET Framework (DURÃES, 2008).

O .NET Framework tem como base um modelo orientado à objetos, possui em sua estrutura cerca de sete mil classes com diferentes funcionalidades, como acesso a dados e comunicação e manipulação de arquivos. Possui um conceito de CTS

(*Common Type System*), o que permite a utilização de várias linguagens de programação. É utilizado o padrão MSIL (*Microsoft Intermediate Language*), compila a aplicação no momento da primeira utilização, usando o processo JIT (*Just-in-time compilation*). Com isso surgiu uma nova versão do ASP, o ASP .NET (MSDN).

3.1.2. Conceitos

O ASP .NET é um novo modelo de desenvolvimento de aplicações web baseado nos princípios de orientação à objetos com suporte do .NET Framework. É utilizado para criação de aplicações Web (DURÃES, 2008).

Para o desenvolvimento em ASP .NET o desenvolvedor conta com o auxílio de um ambiente integrado de desenvolvimento, IDE (*Integrated Development Enviroment*), o Visual Studio 2008 (DURÃES, 2008) (MSDN).

As aplicações baseadas no ASP .NET utilizam todos os recursos do .NET Framework, como as linguagens de programação VB .NET e C# (CSharp), compilação e depuração. Com um ambiente integrado de desenvolvimento é possível desenvolver para web de forma similar a uma aplicação Windows (DURÃES, 2008).

O modelo de codificação do ASP .NET é o “*Code Behind*”, onde o código do aplicativo não é misturado com o código HTML. Aplicativos web ASP .NET são aplicativos compilados e não *scripts* interpretados a cada requisição, isso gera benefícios no desempenho da aplicação. Para a segurança da aplicação no ASP .NET não é mais necessário a inserção de códigos em todas as páginas, como era feito no ASP. A segurança é totalmente integrada e possui componentes prontos para essa funcionalidade, o desenvolvedor só precisa se preocupar com as regras de negócios (DURÃES, 2008) (MSDN).

3.2. VISUAL TEAM STUDIO 2008

Esta seção tem como fonte o site da Microsoft, MSDN.

O Visual Studio 2008 é uma ferramenta de desenvolvimento para construção de aplicações para plataforma .NET Framework, permite a criação de diversos projetos como:

- *Windows Applications (Windows Forms)*
- *Web Applications (ASP .NET)*
- *Windows Service (ASP .NET)*
- *Web Services*
- *SmartPhone / PocketPC*
- *Office 2003 / 2007*
- *SQL 2005 / SQL 2008*
- *Silverlight*

Lançado em fevereiro de 2007 traz nessa nova versão recursos como:

Multi Target: permite o desenvolvimento para .NET 2.0, .NET 3.0 e .NET 3.5 e garante ao desenvolvedor uma facilidade na migração do Visual Studio 2005 para o 2008 sem a necessidade da migração imediata da aplicação, que pode ser feita a qualquer momento com o mínimo de conflitos.

Microsoft AJAX: instalado dentro do Visual Studio 2008 elimina a necessidade de configurações adicionais. Possui novos *templates* de projeto como ASP.NET AJAX *Server Control Extender* e ASP.NET AJAX *Server Control*.

A nova versão do Visual Studio possui um novo editor de HTML, onde existe a divisão do código HTML e a visualização do mesmo. Conta, também, com um amplo suporte a CSS (*Cascading Style Sheets*) e novos recursos como o CSS Properties, *Manage Styles* e *Apply Styles* além de *Intellisense* direta no CSS.

Com o Visual Studio 2008 é possível inserir pelo designer *master pages* em cascata, ou seja, uma *master page* dentro da outra.

O .NET Framework 2.0 juntamente com *Windows Communication Foundation*, *Windows Presentation Foundation* e *Workflow Foundation* compõem a versão 3.0 e foi complementado para dar suporte ao LINQ (Linguagem Integrada de Consulta).

Com o LINQ é possível implementar o mapeamento objeto relacional, abstraindo do desenvolvedor os conhecimentos sobre a base de dados SQL e deixando que o LINQ faça a comunicação.

3.3. MICROSOFT SILVERLIGHT

3.3.1. Conceitos

Silverlight é uma tecnologia desenvolvida pela Microsoft para concorrer com o Macromedia Flash, é uma tecnologia focada na transmissão multimídia e RIA para Web. (MSDN). Uma das vantagens do Silverlight é que é possível programar em qualquer linguagem disponível no Framework .NET. (MSDN)

Anteriormente era chamado de WPF/E, o Silverlight oferece suporte aos navegadores Firefox, Safari e Internet Explorer. Este componente proporciona um conteúdo interativo e sofisticado e não depende de outros produtos para reproduzir seu conteúdo. (MSND)

De acordo com Durães (2008),

Silverlight é uma nova plataforma destinada à construção de aplicações com interfaces ricas para a Web. É baseado em um modelo *cross-browser* e *cross-plataform* permitindo o uso nos mais vários browsers e sistemas operacionais. O Silverlight é um dos componentes do WPF aliado ao grande poder de programação do Microsoft .NET Framework 3.5. O Silverlight suporta JavaScript/HTML DOM e pode ser codificado usando VB.NET ou C# para programação e XAML para interface. Você poderá utilizar o LINQ (Language Integrated Query) além do WCF (Windows Communication Foundation) para otimizar mais ainda a construção das aplicações. Na primeira vez que se carregar uma aplicação baseada no Silverlight ele vai instalar o plug-in no navegador (+- 4.6 Megas). Do ponto de vista de programação, a infra estrutura do Silverlight funciona como um micro versão do Microsoft .NET Framework que é distribuída pela web e instalada nos navegadores clientes.

Essa nova tecnologia de apresentação visual foi desenhada para funcionar em diversas plataformas. É um *plug-in* para renderização do XAML, que são arquivos XML (*eXtensible Markup Language*) usados para especificar uma interface de aplicação no Silverlight ou WPF, nos browsers e executa eventos. Para a construção de aplicações ricas na web utiliza o .NET que permite o desenvolvimento visual de

interfaces com experiências interativas e alta produção que podem ser executadas em navegadores, dispositivos móveis e sistemas operacionais desktops (DURÃES, 2008).

A base conceitual do Silverlight é o conjunto do WPF, Microsoft .NET Framework 3.0 com o XAML. Os pontos interessantes desta tecnologia são:

- O Silverlight é independente de navegador e Sistema Operacional;
- Possui um layout consistente, ou seja, independente do browser não terá sua imagens contorcidas;
- Permite ações de zoom e *drag-and-drop* (clique e arrastar);
- Suporta gráficos 3D;
- Faz leitura e recebimento de dados sem a necessidade de atualizar a página;
- É executado no contexto do navegador e não tem acesso à máquina do cliente;
- Realiza *Stream* de áudio e vídeo;
- O download do *plug-in* é pequeno (em torno de 4MB) (DURÃES, 2008) (MSDN).

3.3.2. Arquitetura do Silverlight

A arquitetura do Silverlight é dividida em duas grandes partes e no componente de instalação. (MSDN)

- Core Presentation: É o Framework de apresentação. Contém os componentes e serviços orientados para a Interface do Usuário (UI).
- .NET para Silverlight: Contém os componentes e as bibliotecas. É um subconjunto do .NET Framework.
- Instalador e atualizador: A instalação, feita através de um *plug-in*, ocorre apenas uma vez e obtém suas atualizações automaticamente, sem gerar impacto.

A figura 3 ilustra a arquitetura do .NET para Silverlight, e mostra as melhorias efetuadas com a atualização da primeira versão para a versão 2.0 do *plug-in*.

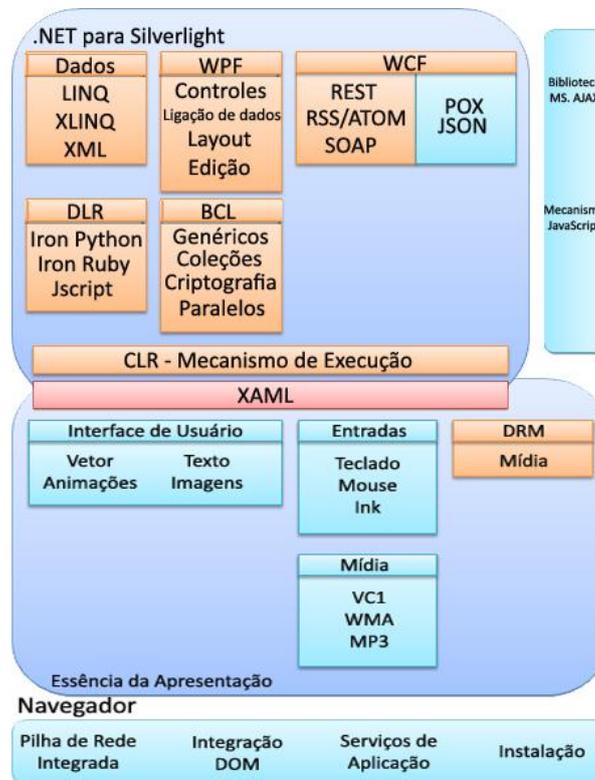


Figura 3. Arquitetura .NET para Silverlight. (MSDN)

O núcleo da funcionalidade do Silverlight é fornecido por um *plug-in* que retorna XAML e expõe os Modelos de Objeto do Documento – DOM para o browser via JavaScript. Desta forma é possível capturar os eventos do usuário, como o movimento e o clique do mouse e manipulá-los com a chamada de métodos JavaScript com elementos XAML (MSDN).

Desta forma um profissional *designer* pode reunir um documento XAML que contenha gráficos, animações e linhas do tempo e um profissional de desenvolvimento pode utilizá-lo junto ao código de uma página e implementar novas funcionalidades. O arquivo XAML é baseado em XML e pode ser emitido em tempo de execução por um aplicativo de servidor, o que proporciona uma experiência gráfica altamente dinâmica e personalizável (MSND).

A figura 4 mostra a arquitetura do Silverlight, onde o *plug-in* do Silverlight suporta os tipos de mídia para rodar em qualquer sistema operacional e funcionar em qualquer navegador.



Figura 4. Arquitetura Silverlight. (MSDN)

A principal interface de programação é a API do DOM *JavaScript*, com isso é possível enviar uma resposta aos eventos gerados no XAML do Silverlight, ou seja, quando um conteúdo acaba de ser carregado e quando conclui uma animação. Por trás disso existe o mecanismo de análise de XAML. O analisador cria o DOM XAML na memória para ser usado pelo núcleo da apresentação, este núcleo manipula o processamento da animações e dos gráficos definidos XAML. Os *codecs* necessários para reprodução do conteúdo multimídia WMV/WMA/MP3 estão contidos no tempo de execução, que contém, também, o núcleo da apresentação, que gerencia o processamento (MSDN).

Esse tempo de execução da apresentação é incorporado a um plug-in de navegador. O resultado final é um mecanismo de processamento de gráficos e mídia auto-suficiente, que pode ser conectado ao navegador e programado via *JavaScript* (MSDN).

A arquitetura de uma aplicação simples, usando um arquivo XAML que define a interface do usuário e o *JavaScript* que manipula os eventos, executada no navegador usando Silverlight é mostrada na Figura 5.

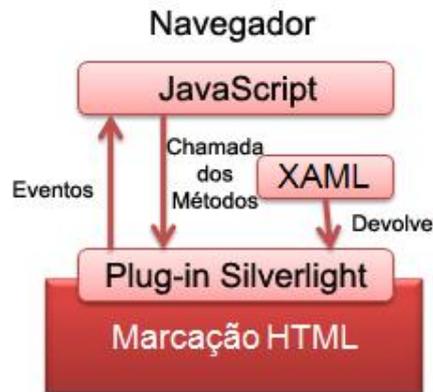


Figura 5. Comunicação Silverlight. (MSDN)

O navegador carrega o arquivo XAML ao instanciar o *plug-in*. Os eventos nesse arquivo são capturados pelo navegador e processados pelo *JavaScript*. Conforme o DOM do conteúdo do Silverlight é exposto, o código *JavaScript* pode também atualizar dinamicamente o conteúdo do Silverlight, alterando o estado do conteúdo processado (MSDN).

Como o aplicativo é executado dentro do navegador, ele é normalmente feito de marcação HTML. Esta marcação contém as chamadas para instanciar o *plug-in* Silverlight. Como os usuários interagem com o Silverlight, eles acionam eventos que podem ser capturados pelas funções escritas em *JavaScript*. Por sua vez, o programa escrito em código *JavaScript* pode fazer chamadas de métodos contra os elementos dentro do conteúdo Silverlight para manipulá-lo, acrescentar novos conteúdos, ou remover o conteúdo existente. Finalmente, o XAML pode ser lido pelo *plug-in*. O XAML pode existir tanto dentro da página como fora em um arquivo estático ou dinâmico que será devolvido por um servidor (MSDN).

3.3.3. Ferramentas para desenvolvimento de Silverlight

A suíte de trabalho usada para o desenvolvimento de aplicações Silverlight é o Microsoft Expression Studio. Ela atende a todas as exigências do desenvolvimento Silverlight, são elas: um bom visual, todas as técnicas de usabilidade, interatividade e animação. Esta suíte de desenvolvimento possui seis ferramentas:

- Expression Web: ferramenta para a criação de sites Web modernos, dentro dos padrões W3C;
- Expression Blend: ferramenta para a criação de aplicativos ricos, tanto para a internet quanto para aplicações Windows. Auxilia, também, a criação do XAML;
- Expression Design: ferramenta para a criação de elementos visuais para aplicações Windows;
- Expression Media: ferramenta para a organização, manipulação e recuperação de elementos gráficos e multimídia;
- Deep Zoom Composer: ferramenta que prepara imagens para o uso do recurso deep zoom, incorporada ao Silverlight;
- Expression Encoder: ferramenta para codificar e publicar vídeos.

3.4. WINDOWS PRESENTATION FOUNDATION (WPF)

O WPF é baseado em um modelo de imagens vetoriais e tem sua implementação feita através do *eXtensible Application Markup Language* (XAML) que por sua vez pronuncia-se zamel. É um modelo de camada de apresentação da plataforma Microsoft .NET (DURÃES, 2008).

4. GONLO – GERENCIADOR ONLINE DE LOJAS VIRTUAIS.

Este capítulo apresenta detalhadamente o desenvolvimento do trabalho, ou seja, o desenvolvimento do aplicativo para criação de sites de comércio eletrônico utilizando conceito de aplicações ricas, a modelagem e explicações sobre os códigos fontes. Apresenta o conceito de comércio eletrônico para o melhor entendimento da aplicação criada.

4.1. COMÉRCIO ELETRÔNICO

O Comércio Eletrônico pode ser entendido como negociações de compra efetuadas através da internet, onde se aplica tecnologia de comunicação e de informação. Isto tem transformado as transações comerciais, a sociedade empresarial e até mesmo a própria internet (ALBERTIN, 2004).

De acordo com Kalakota e Whinston (1996) comércio eletrônico pode ser definido como “uma moderna tecnologia de negócios, que direciona as necessidades de organizações, mercados e consumidores diminuindo custos e aumentando a qualidade das mercadorias e serviços, acelerando o processo de entrega”. Esta combinação entre a tecnologia de informação, processos e estratégias de empresas que facilitam o intercâmbio de informações, produtos e serviços é dada como comércio eletrônico (CHEN, 2000). É um sistema comercial capaz de realizar compra/troca de bens ou serviços entre duas ou mais partes utilizando a internet, ou seja, uma interação das partes envolvidas.

A eliminação de barreiras entre empresa-cliente, ou seja, *business-to-consumer* (B2C) fornece ao consumidor informações que possibilitem uma maior opção de escolha, este tipo de compra é caracterizado como um evento discreto, onde o consumidor pode acessar vários sites e efetuar compras de artigos semelhantes (SHAW, 1999) (CLEMENTE, 1998) (MAYA E OTERO, 2002).

No entanto para o sucesso total de um site de comércio eletrônico é necessário a empresa conhecer o seu tipo de cliente e entender todas as mudanças que uma loja virtual acarreta ao comportamento do cliente, mesmo com toda a tecnologia disponível não é possível esquecer o principal mantedor das empresas, o cliente (YAMASHITA; GOUVÊA, 2004).

4.2. A ARQUITETURA DA APLICAÇÃO

Conforme os conceitos de comércio eletrônico, o Gerenciador Online de Lojas Virtuais – GonLo será dividido em três módulos, são eles: módulo administrativo, módulo usuário e módulo público. Como é exemplificado na figura 6.

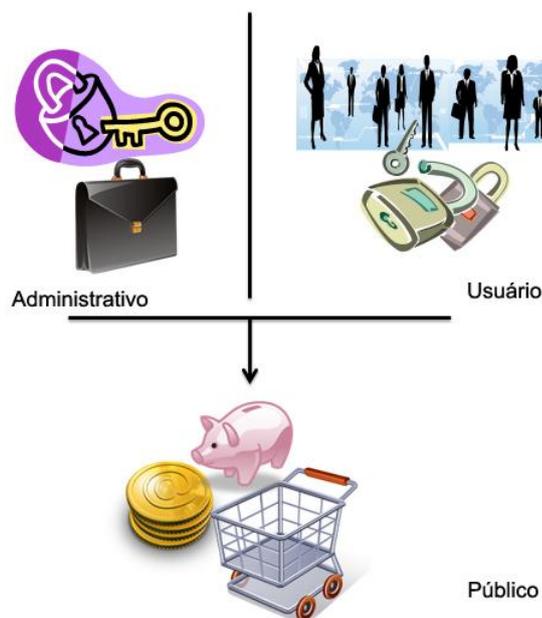


Figura 6. Módulos da Aplicação

O módulo Administrativo tem como função controlar as lojas virtuais. Nesse módulo é possível manter as lojas virtuais (cadastrar, modificar, apagar), gerar contratos e controlar as permissões do módulo de usuários, possui, também, todas as funções do módulo usuário. Apenas proprietários da aplicação possuem acesso ao módulo administrativo.

O módulo Usuário é o acesso de cada loja virtual. É a manutenção da loja, de seus produtos, categorias e fornecedores, ou seja, neste módulo os proprietários das lojas virtuais podem cadastrar, modificar, e apagar o itens citados acima. Neste módulo são selecionadas as ofertas e depois de todas as configurações finalizadas a loja será disponibilizada para o acesso público.

O módulo público consiste na loja virtual. Qualquer cliente com um computador provido de acesso à internet acessa de qualquer lugar do mundo a loja e pode fazer o cadastro para efetuar compras de produtos de seu interesse. O cliente pode controlar a sua compra, adicionando e excluindo produtos e modificando a quantidade deles enquanto a compra não for finalizada e o pagamento for confirmado. Neste módulo o cliente visualiza a loja, pode selecionar a categoria do produto que esteja procurando. Na primeira página, visualiza todas as ofertas da loja virtual, ao clicar em um produto terá acesso aos seus detalhes, preço e imagens.

4.3. MODELAGEM

Na concepção deste aplicativo foram levantados os requisitos necessários para que o modelo atenda todos os objetivos propostos.

O aplicativo tem o objetivo de demonstrar a utilização de RIA e suas funcionalidades utilizando Silverlight por meio do desenvolvimento de uma aplicação web com a função de criar e gerenciar lojas virtuais. Deve permitir o cadastro e controle de lojas virtuais, clientes, usuários e produtos.

As vendas realizadas podem ser canceladas, no entanto, não podem ser excluídas. Toda venda realizada é armazenada no banco de dados e é controlada a partir de um campo onde indica se está ativa ou cancelada.

Os proprietários das lojas virtuais não têm acesso e nem controle aos usuários e a criação de novas lojas virtuais, ficando esta característica restrita aos administradores do GonLo.

O ambiente público, onde os clientes se cadastram e efetuam compras, é gerenciado pelo público com as opções de inserir um novo cliente e uma nova compra bem como atualizar os dados de ambos. O público não tem a opção de

excluir seu cadastro e tão pouco excluir uma venda, as vendas podem ser canceladas, somente pelo cliente que realizou seu cadastro.

Os eventos do aplicativo são mostrados na tabela 1. Para cada evento existe um ou mais atores responsáveis pela sua execução.

Lista de Eventos			
Nº	Evento	Descrição	Ator
01	Efetuar Acesso	Efetuar Acesso no aplicativo	Administrador/Usuário/Público
02	Cadastrar Loja Virtual	Cadastrar nova Loja	Administrador
03	Cadastrar Cliente	Cadastrar novo cliente	Público
04	Cadastrar Produto	Cadastrar e gerenciar produtos	Usuário
05	Cadastrar Categorias	Cadastrar nova categoria	Administrador/ Usuário
06	Cadastrar Cidades	Cadastrar nova cidade	Administrador
07	Cadastrar Usuários	Cadastrar novo usuário	Administrador
08	Consultar Preços	Consultar preço de produtos	Público
09	Detalhes do Produto	Ver detalhes sobre um produto	Público
10	Adicionar ao Carrinho	Adicionar produto ao carrinho de compras	Público
11	Fechar Pedido	Fechar pedido e exibir valor total	Público
12	Gerenciar Pedidos	Consultar e administrar pedidos realizados	Usuário
13	Vendas por Produto	Gerar relatório de vendas por produto	Usuário
14	Vendas por Meses	Gerar relatório de vendas por mês	Usuário

Tabela 1. Lista de Eventos do Aplicativo

Os diagramas abaixo foram propostos seguindo a arquitetura da aplicação.

4.3.1. Diagrama de Caso de Uso

O Diagrama de Caso de Uso demonstra as funcionalidades de um sistema, após analisar os requisitos este diagrama possibilita a modelagem de funções ou serviços

que o sistema deve realizar. Para atender as necessidades propostas segue abaixo o Diagrama de Caso de Uso.

A Figura 7 exhibe o Diagrama de Caso de Uso. De acordo com a arquitetura da aplicação os atores representam os módulos e as classes representam suas ações.

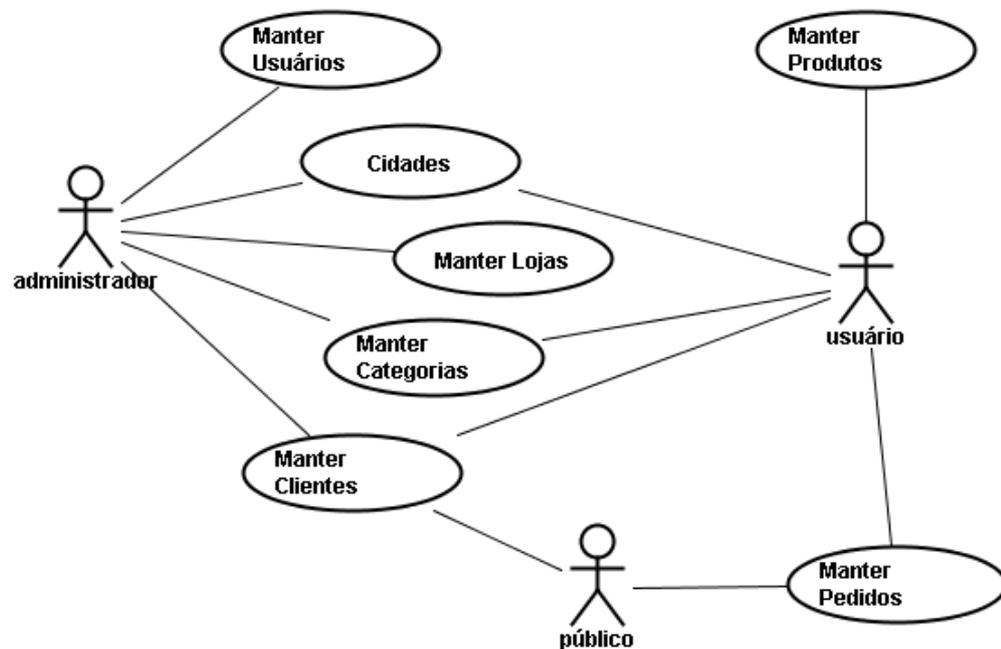


Figura 7. Diagrama de Caso de Uso

O ator administrativo recebe as funções de manutenção de usuários, cidades, lojas, categorias e clientes. O administrador não pode excluir clientes e nem alterá-los, tem a opção de visualização de inserção. Em relação às lojas e aos usuários, é possível realizar a inserção e alteração. As categorias e as cidades possuem, somente, a opção de inserção, pois são chaves estrangeiras em outras tabelas e não podem ter o seu valor alterado ou excluído.

O ator usuário tem acesso a cidades, categorias, clientes, produtos e vendas. Da mesma maneira que o administrador o usuário não tem a opção de alteração e exclusão de cidades e categorias e os clientes podem somente ser visualizados. Os

produtos podem ser inseridos, alterados e/ou excluídos. Em relação às vendas o usuário tem a opção de visualização e cancelamento.

O ator público tem as opções de inserir e alterar clientes e efetuar uma compra e, posteriormente, alterá-la ou cancelá-la.

4.3.2. Diagrama de Classe

Após a analisar os requisitos e modelar os casos de uso desta aplicação é possível descrever o formato das classes e objetos que se faz necessário para o desenvolvimento do aplicativo. Em uma aplicação orientada a objetos, tais itens são as principais primitivas para a construção deste tipo de aplicação, sendo que um sistema orientado à objetos é composto por classes e objetos que interagem entre si para a execução dos serviços necessários ao funcionamento do aplicativo (STADZISZ, 2002).

A figura 8 apresenta o diagrama de classe da arquitetura da aplicação. Este diagrama apresenta o relacionamento das tabelas, seus atributos e as funções que cada tabela pode executar. A tabela de cidades possui os atributos código, nome e estado e pode-se cadastrar, excluir ou atualizar uma cidade.

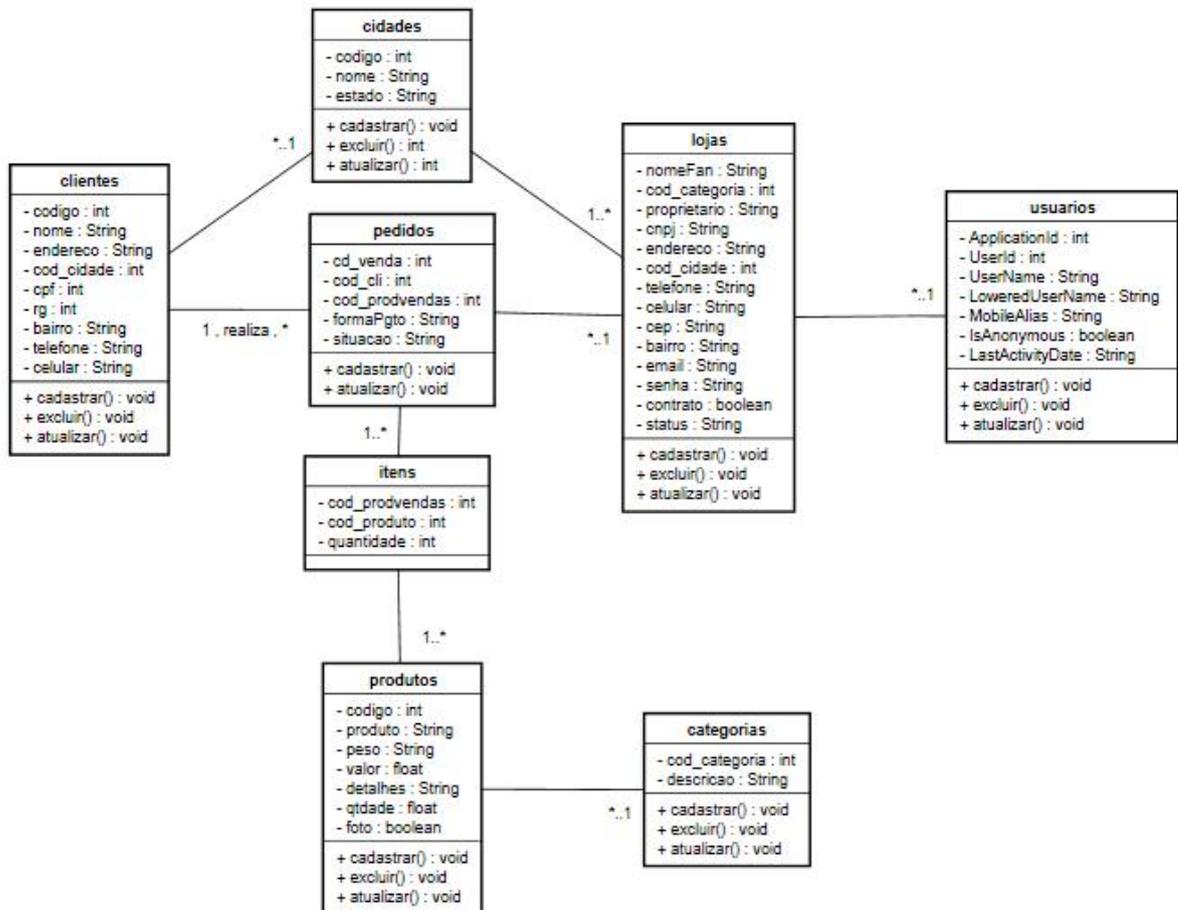


Figura 8. Diagrama de Classe

4.3.3. Diagrama de Seqüência

Um diagrama de seqüência é composto por um conjunto de objetos, neste diagrama são demonstradas as comunicações necessárias entre os objetos para a execução do sistema. Por meio de uma linha de tempo estes diagramas descrevem a seqüência de comunicação entre os objetos (STADZISZ, 2002).

A figura 9 mostra o diagrama de seqüência para o primeiro evento: Efetuar Acesso, utilizando o modo público.

O caso de uso abaixo possui o curso normal em que:

1. Público solicita opção da tela de acesso;
2. Aplicativo disponibiliza tela de acesso;

3. Público informa usuário e senha;
4. Aplicativo consulta em usuário se os dados estão corretos;
5. Aplicativo libera o acesso ao ator Público;
6. Aplicativo emite mensagem: usuário liberado.

E o curso alternativo, que mostra que o usuário e/ou a senha estão incorretos ou não existem.

- 4.1. Aplicativo consulta em usuário se os dados não estão corretos;

- 4.1.1. Aplicativo retorna ao item dois do curso normal.

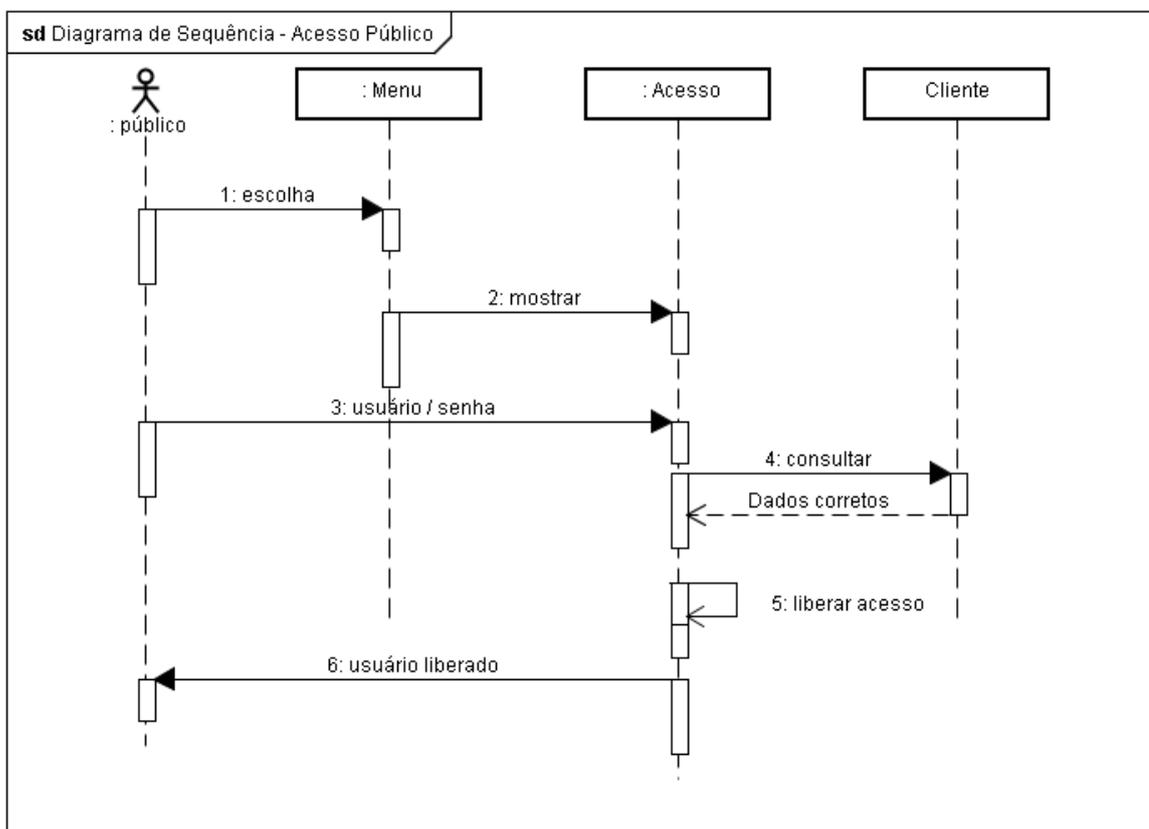


Figura 9. Diagrama de Seqüência – Acesso Público

A figura 10 mostra o curso normal do cadastro de produtos. Neste percurso ocorre:

1. Ator Usuário solicita opção da tela de cadastrar produto;
2. Aplicativo disponibiliza tela de cadastrar produto;

3. Aplicativo consultar produtos cadastrados;
4. Aplicativo exibe formulário de cadastro e lista de produtos;
5. Usuário informa dados do produto;
6. Aplicativo grava em produto;
7. Aplicativo retorna ao item três do curso normal.

Este evento possui dois percursos alternativos, a primeira alternativa mostra a alteração de um produto. Os passos seguidos para alteração seguem abaixo:

- 5.1. Usuário solicita atualizar dados do produto;
 - 5.1.1. Aplicativo consulta dados atuais do produto;
 - 5.1.2. Aplicativo disponibiliza dados atuais para edição;
 - 5.1.3. Usuário informa dados atualizados;
 - 5.1.4. Aplicativo atualiza os novos dados em Produto;
 - 5.1.5. Retorna ao item três do curso normal.

A segunda alternativa realiza a exclusão de um produto.

- 5.2. Administrador solicitar apagar produto;
 - 5.2.1. Aplicativo atualiza situação de produto para indisponível;
 - 5.2.2. Retorna ao item três do curso normal.

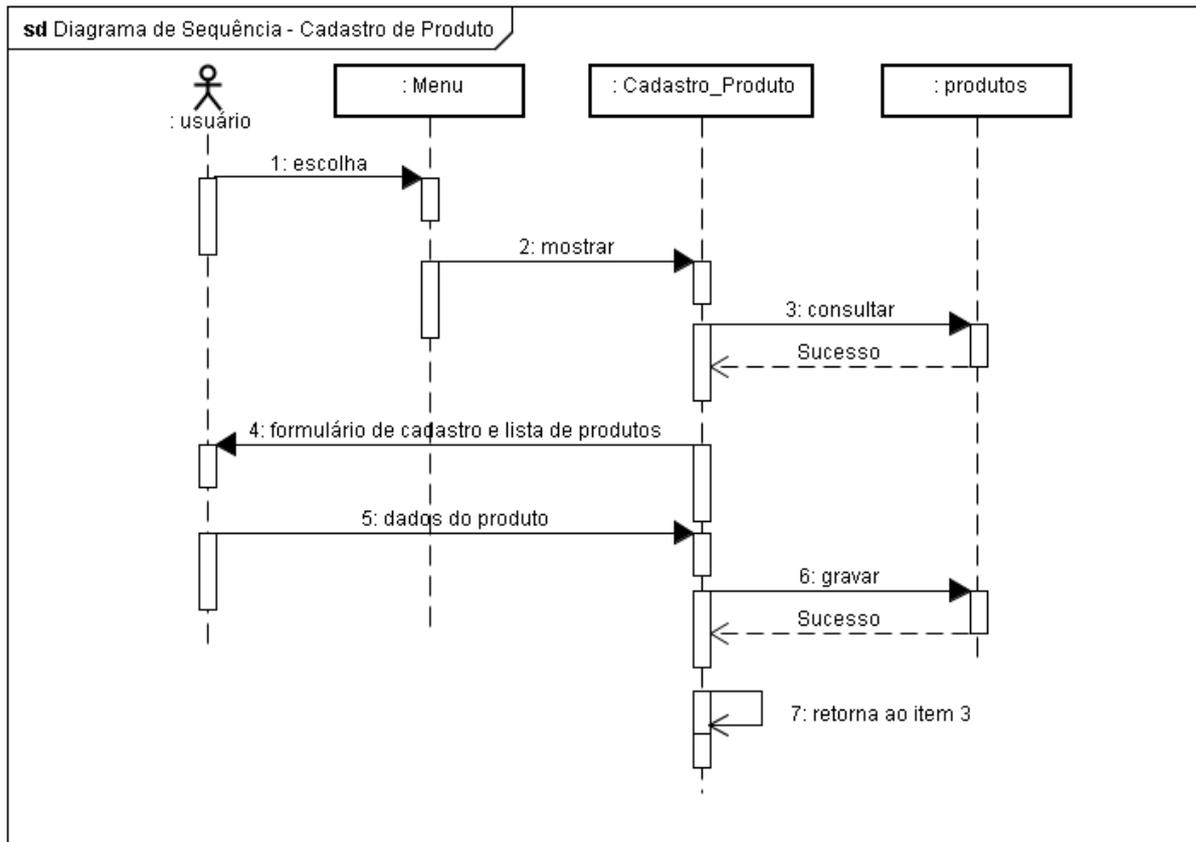


Figura 10. Diagrama de Seqüência – Cadastro Produto, curso normal

A figura 11 mostra o diagrama de seqüência de uma venda em seu curso normal. Este percurso possui o curso normal e três cursos alternativos. No curso normal:

1. Público solicita adicionar item ao carrinho;
2. Aplicativo armazena item selecionado;
3. Aplicativo consulta a descrição em produto;
4. Aplicativo exibe relação de itens do carrinho.

O primeiro curso alternativo mostra a compra de um produto com mais de um item.

- 4.1.1 Público informa nova quantidade do item;
- 4.1.2 Aplicativo atualiza quantidade do item;
- 4.1.3 Aplicativo retorna ao item três.

O curso segundo alternativo apresenta a exclusão de um item do carrinho de compras.

4.2.1 Público solicita retirar item do carrinho;

4.2.2 Aplicativo retira o item do carrinho;

4.2.2 Aplicativo retorna ao item três do curso normal.

Por fim, o último curso alternativo mostra o cancelamento da venda, ou seja, a exclusão de todos os produtos do carrinho de compras.

4.3.1 Público solicita esvaziar carrinho;

4.3.3 Aplicativo retira todos os itens do carrinho;

4.3.2 Aplicativo retorna ao item quatro do curso normal.

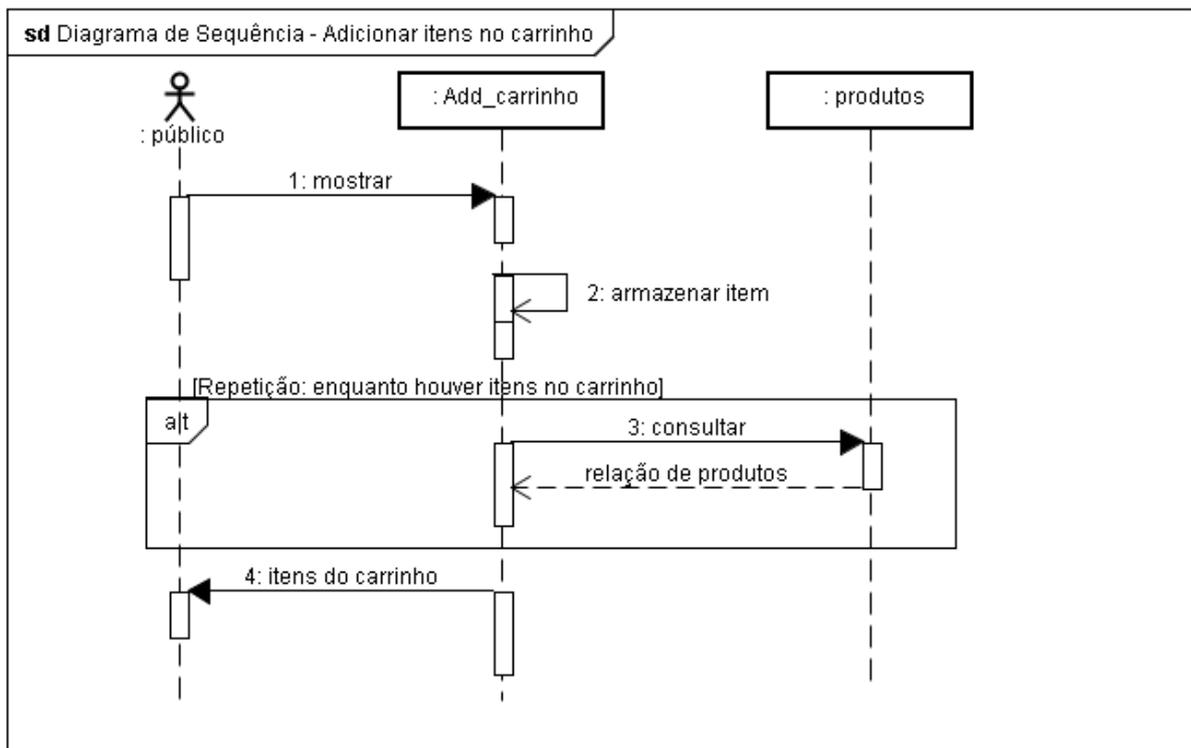


Figura 11. Diagrama de Seqüência – Adicionar itens ao carrinho de compras, curso normal

A figura 12 mostra o diagrama de seqüência para a finalização da compra, em seu curso normal apresenta os seguintes passos:

1. Público solicita fechar pedido;
2. Aplicativo verifica se o usuário público está conectado;
3. Aplicativo grava em pedido;
4. Aplicativo grava em itens;
5. Aplicativo esvazia o carrinho;
6. Aplicativo exibe dados do pedido.

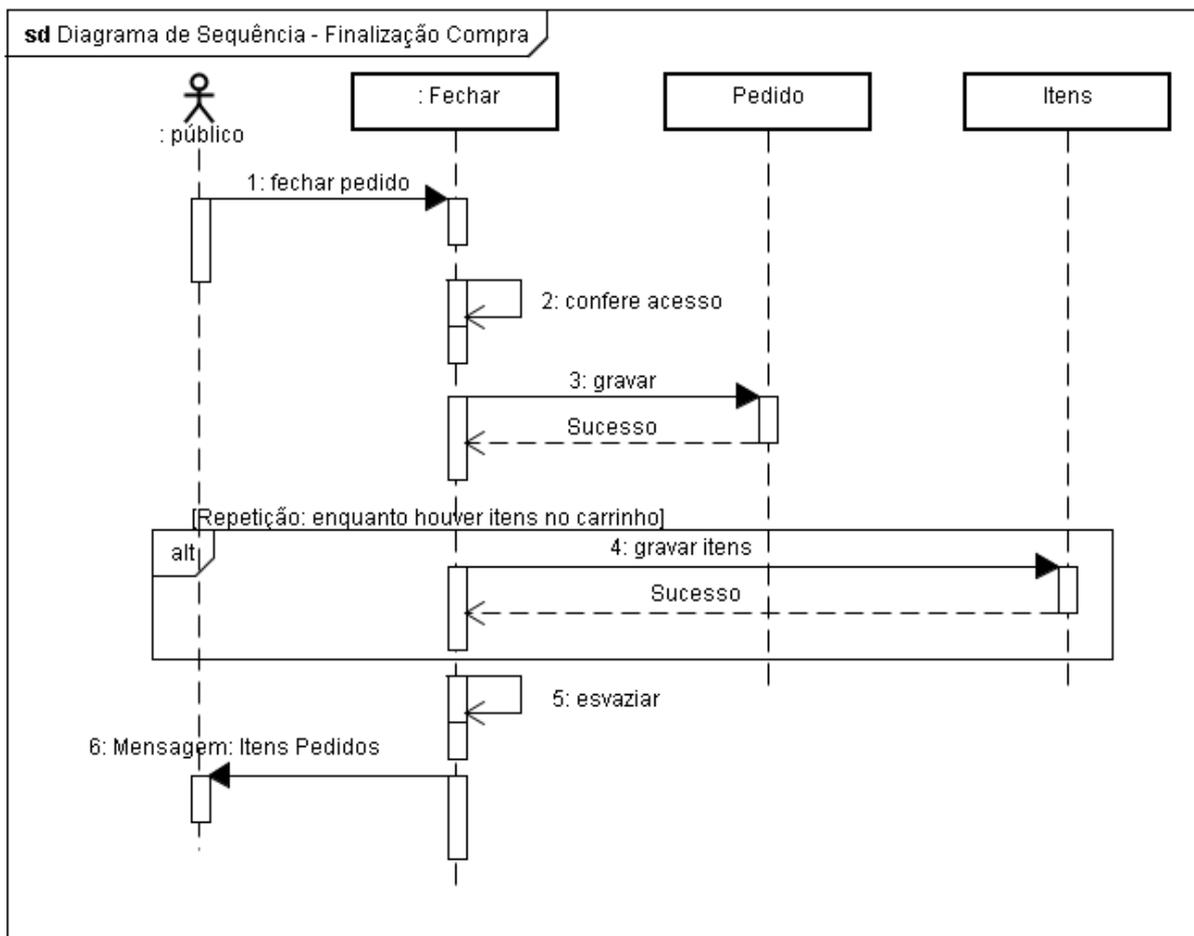


Figura 12. Diagrama de Seqüência – Finalização Compra

4.3.4. Estrutura Tecnológica da Aplicação

A Aplicação será desenvolvida utilizando as tecnologias abaixo:

- ❖ Linguagem C#;
- ❖ Ferramenta de Desenvolvimento Visual Studio 2008;
- ❖ Banco de dados SQL Server 2005;
- ❖ Tecnologia .NET Framework;
- ❖ Microsoft Silverlight;

4.4. IMPLEMENTAÇÃO DO TRABALHO

4.4.1. Interface do Aplicativo

O *GonLo, Gerenciador On-Line de Lojas Virtuais* teve sua interface desenvolvida conforme a análise acima.

O aplicativo possui a página inicial, que pode ser acessada por qualquer usuário (ambiente público) sem a necessidade de cadastro no site, onde são apresentadas todas as lojas virtuais cadastradas no GonLo. O menu do site possui as opções de Início, onde a partir de qualquer outra página é possível o retorno à página inicial, a opção de Produtos, onde os usuários públicos tem a opção de visualizar, ao invés das lojas, os produtos cadastrados em todas as lojas disponíveis no GonLo, que são visualizados separados por categorias, maior ou menor preço e promoções. Possui, também, a opção Minha Loja, onde o usuário da Loja Virtual pode acessar diretamente o cadastro de novos produtos, ao clicar em Minha Loja o usuário será levado a uma página de Acesso, onde deverá digitar seu nome de usuário e senha para ter acesso à sua loja virtual e cadastrar novos produtos. Apresenta a opção de Criar Loja, onde é necessário o acesso com o nome de usuário e senha administrativo, onde é possível realizar o cadastro de novas lojas virtuais ou manter as existentes e controlar os usuários e seus acessos. Possui, ainda, a opção de Contato, onde o público pode entrar em contato com o setor administrativo do

GonLo, para críticas e sugestões. No canto inferior direito da tela apresenta uma imagem de carrinho de compras, onde o público pode fazer o acesso e então começar sua compra. Se o ator público (cliente) adicionar produtos ao carrinho de compras sem ter feito o acesso ao finalizar a compra será solicitado seu nome de usuário e senha.

A figura abaixo apresenta a interface da página inicial do GonLo. Nesta página são mostradas as lojas virtuais cadastradas, o menu de opções e o campo de busca com *links* prontos de busca por categorias, ao clicar em alguma categoria será direcionado a tela de produtos apresentando somente os produtos da categoria selecionada.



Figura 13. Página Inicial

Ao passar o cursor do *mouse* sobre os ícones são mostradas suas descrições.

A figura 14 apresenta o menu de opções com as descrições dos ícones.



Figura 14. Descrição Menu

A figura abaixo (figura 15) apresenta a interface do GonLo quando clicado na segunda opção do menu, a opção Produtos. Quando selecionada, muda a visualização da página principal, onde antes apresentava imagens e detalhes das lojas virtuais agora apresenta detalhes e imagens dos produtos, separados por categorias. Esta tela é visualizada, também, quando selecionado algum *link* de categoria no canto superior direito de qualquer tela.



Figura 15. Produtos

A figura 16 apresenta a tela de acesso, esta tela apresenta a opção de efetuar o acesso ou criar um novo usuário. A opção de criar um novo usuário permite, apenas, a criação de usuários públicos, ou seja, clientes das lojas virtuais.

Ao efetuar o acesso no site, dependendo da permissão do usuário será encaminhado para outra tela com as opções correspondentes.

Gerenciador de Loja Virtual - GonLo

Buscar

Categorias

- Roupas
- Sapatos
- Acessórios
- Tecnologia
- CD / DVD
- Decoração
- E-books

Usuário:

Senha:

Acesso

[Novo Usuário](#)



Figura 16. Tela de Acesso

A figura abaixo mostra a tela após o acesso quando o usuário e senha forem autenticados como administrador.

O usuário administrador tem a permissão para criar novas lojas virtuais e controlar as existentes. Tem, também, a opção de gerenciar as lojas virtuais cadastradas, controlar as imagens, cadastrar categorias, cidades e usuários.

Gerenciador de Loja Virtual - GonLo

Gerenciar Lojas
Imagens
Categorias
Cidades
Usuários

Nome Fantasia:

Categoria:

Proprietário:

CNPJ:

Endereço:

Cidade: Assis

CEP: Bairro:

Telefone: Celular:

Email: Senha:

Status: Ativa

Contrato: Sim

Foto:

Buscar

Categorias
Roupas
Sapatos
Acessórios
Tecnologia
CD / DVD
Decoração
E-books



Figura 17. Criar Loja - Tela visualizada após a autenticação

A figura 18 mostra a tela após o acesso quando o usuário e senha forem autenticados como usuários. Com este tipo de permissão é possível controlar uma loja virtual e seus produtos, suas vendas e promoções.

Gerenciador de Loja Virtual - GonLo

Gerenciar Produtos
Imagens
Categorias
Cidades
Vendas

Produto:

Peso:

Valor:

Detalhes:

Quantidade:

Status: Ativa

Foto:

Buscar

Categorias
Roupas
Sapatos
Acessórios
Tecnologia
CD / DVD
Decoração
E-books

Código	Descrição	Preço Unit.	Estoque		
1	Caneta	2,50	50	<input type="checkbox"/>	<input type="checkbox"/>
2	Lápis	1,20	20	<input type="checkbox"/>	<input type="checkbox"/>
3	Borracha	3,50	10	<input type="checkbox"/>	<input type="checkbox"/>
4	Pen drive	45,50	20	<input type="checkbox"/>	<input type="checkbox"/>
5	Impressora	170,00	5	<input type="checkbox"/>	<input type="checkbox"/>



Figura 18. Minha Loja – Tela visualizada após a autenticação

A tela abaixo apresenta a opção de Contato, onde qualquer pessoa pode acessar e enviar críticas, sugestões e dúvidas.

Figura 19. Tela de Contato

4.4.2. Programação do Aplicativo

O aplicativo foi desenvolvido usando as interfaces de programação Microsoft Visual Studio 2008 e Microsoft Blend. As telas da interface mostradas anteriormente foram desenvolvidas utilizando Microsoft Blend.

Foi utilizado em sua construção o conceito de programação em três camadas, ou seja, para efetuar um cadastro é necessária a criação de três classes. A divisão em diferentes camadas é utilizada para separar as responsabilidades de cada parte do sistema. Este modelo é utilizado quando um sistema cliente-servidor é desenvolvido retirando-se a camada de negócio do lado do cliente.

O modelo de três camadas possui três partes: camada de apresentação, camada de negócio e camada de dados.

A camada de apresentação é a chamada GUI (Graphical User Interface), ou simplesmente interface. Esta camada interage diretamente com o usuário, é através

dela que são feitas as requisições como consultas, por exemplo. A camada de negócio, conhecida como Lógica empresarial, regras de negócio ou funcionalidade é onde fica as funções e regras de todo o negócio do sistema.

A camada de dados é a terceira camada, é o repositório de informações. Esta camada recebe as requisições da camada de negócios e seus métodos executam essas requisições em um banco de dados. Alterando o banco de dados alteraria apenas as classes da camada de dados, e o restante das camadas não seriam afetados por essa alteração.

Como dito anteriormente, Silverlight é desenvolvido utilizando a linguagem XAML, derivada de XML. A figura abaixo mostra a programação da interface gráfica, ou seja, o código XAML da tela inicial do aplicativo.

```

1 <navigation:Page
2   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6   mc:Ignorable="d"
7   xmlns:navigation="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Navigation"
8   xmlns:dataInput="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data.Input"
9   x:Class="tcc.Login"
10  Title="Login Page"
11  d:DesignWidth="1024" d:DesignHeight="768">
12
13  <Grid x:Name="LayoutRoot" >
14    <Rectangle x:Name="backgroundPanel" Fill="#FFFFFF" Stroke="#FFC1C1" StrokeLineJoin="Round" StrokeThic
15    <Rectangle x:Name="TitleTop" RadiusX="22" RadiusY="22" Height="71" Margin="6,7,0,0" VerticalAlignment="To
16      <Rectangle.Fill>
17        <LinearGradientBrush StartPoint="0.518981,1.0233" EndPoint="0.518981,-0.00193842">
18          <GradientStop Color="#FF0A3752" Offset="0"/>
19          <GradientStop Color="#FF0B3E5D" Offset="0.694"/>
20          <GradientStop Color="#FF497189" Offset="0.716"/>
21          <GradientStop Color="#FF9CB6C5" Offset="0.966102"/>
22        </LinearGradientBrush>
23      </Rectangle.Fill>
24    </Rectangle>
25    <TextBlock x:Name="PanelTitle" Height="50" Margin="21,18,0,0" Text="Gerenciador de Loja Virtual - GonLo"
26      <TextBlock.Foreground>

```

Figura 20. Código XAML da tela inicial – Camada de Apresentação

Para que seja possível a inclusão e visualização dos dados na interface é necessário a implementação de duas classes, no exemplo, uma que contém um produto e outra que contém uma lista de produtos.

A classe que contém um produto é implementada com os mesmos atributos da tabela no banco de dados e os métodos *get* e *set* para cada atributo. Isso é feito

para que seja possível consultar os dados por outras classes, a figura é apresentada abaixo.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 ///Classe que contém 1 produto
6 namespace MOR
7 {
8     public class ProdutoMOR
9     {
10         private int _Codigo;
11         private string _Produto;
12         private string _Peso;
13         private string _Detalhes;
14         private float _Quantidade;
15         private float _Valor;
16
17         public float Valor
18         {
19             get { return _Valor; }
20             set { _Valor = value; }
21         }
22
23         public float Quantidade
24         {
25             get { return _Quantidade; }
26             set { _Quantidade = value; }
27         }
28
29         public string Produto
30         {
31             get { return _Produto; }
32             set { _Produto = value; }
33         }
34
35         public string Peso
36         {
37             get { return _Peso; }
38             set { _Peso = value; }
39         }
40
41         public string Detalhes
42         {
43             get { return _Detalhes; }
44             set { _Detalhes = value; }
45         }
46
47         public int Codigo
48         {
49             get { return _Codigo; }
50             set { _Codigo = value; }
51         }
52     }
53 }
54

```

Figura 21. Classe que contém um produto

Abaixo é apresentada a classe que contém a lista de produtos, com essa classe é possível fazer a seleção dos produtos.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 ///Classe que contém uma lista de Produtos
6 namespace MOR
7 {
8     public class ProdutosMOR: List<ProdutoMOR>
9     {
10
11     }
12 }

```

Figura 22. Classe que contém uma lista de produtos

A figura abaixo mostra a camada de negócios da aplicação, é nesta camada onde serão tratadas as exceções e chamados os métodos de inserção, atualização, visualização e exclusão.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 ///Camada de Regras de Negócios - Business Logic Layer
6 namespace BLL
7 {
8     public class ProdutosBLL
9     {
10         public MOR.ProdutosMOR Select ()
11         {
12             DAL.Produtos dPrd = new DAL.Produtos ();
13             return dPrd.Select ();
14         }
15         public void Insert (MOR.ProdutoMOR wprd)
16         {
17             DAL.Produtos dPrd = new DAL.Produtos ();
18             dPrd.Insert (wprd);
19         }
20         public void Update (MOR.ProdutoMOR wprd)
21         {
22             DAL.Produtos dPrd = new DAL.Produtos ();
23             dPrd.Update (wprd);
24         }
25         public void Delete (MOR.ProdutoMOR wprd)
26         {
27             DAL.Produtos dProd = new DAL.Produtos ();
28             dProd.Delete (wprd);
29         }
30     }
31 }

```

Figura 23. Camada de negócio

A camada de dados é onde é feita a conexão com o banco de dados da aplicação. Não é necessário fazer a ligação direta desta camada com a camada de aplicação, pois é isso que a camada de negócios faz. A imagem abaixo faz a conexão com o banco de dados através da *ConnectionString* e tem o método de *Selec*, que é usado para mostrar os dados, neste caso dos produtos, na interface gráfica.

```
16 namespace DAL
17 {
18     public class Produtos
19     {
20         private string ConStr = @"Data Source=.\SQLEXPRESS;Initial Catalog=TC;Integrated Security=True";
21
22         public MOR.ProdutosMOR Select()
23         {
24             MOR.ProdutosMOR wProds = new MOR.ProdutosMOR();
25             SqlConnection Con = new SqlConnection(ConStr);
26             Con.Open();
27             string SQL = "select * from Produtos";
28             SqlCommand Cmd = new SqlCommand(SQL, Con);
29             SqlDataReader rd = Cmd.ExecuteReader(CommandBehavior.CloseConnection);
30             try
31             {
32                 while (rd.Read())
33                 {
34                     MOR.ProdutoMOR wPrd = new MOR.ProdutoMOR();
35                     wPrd.Codigo = int.Parse(rd[0].ToString());
36                     wPrd.Produto = rd[1].ToString();
37                     wPrd.Peso = rd[2].ToString();
38                     wPrd.Detalhes = rd[3].ToString();
39                     wPrd.Quantidade = Convert.ToSingle(rd["Quantidade"].ToString());
40                     wPrd.Valor = float.Parse(rd[5].ToString());
41                     wProds.Add(wPrd);
42                 }
43             }
44             finally
45             {
46                 rd.Close();
47             }
48             return wProds;
49         }
50     }
}
```

Figura 24. Camada de dados

5. CONCLUSÃO

A nova tecnologia Microsoft, o Silverlight mostrou-se muito rica, no entanto, é uma tecnologia não aproveitada por não ser muito estudada e utilizada por desenvolvedores, tendo seus recursos e funcionalidades pouco explorados. Os estudos realizados mostram que essa tecnologia possui efeitos em seus componentes que torna clara a satisfação dos usuários.

O Gerenciador Online de Lojas Virtuais - GonLo surgiu para demonstrar o estudo relacionado ao uso desta tecnologia juntamente com a programação em C# .NET. O visual rico e o *designer* moderno tornam o uso da ferramenta mais agradável aos usuários. O objetivo do GonLo é garantir uma maior usabilidade para clientes das lojas virtuais. Como o cadastramento de várias lojas virtuais é possível o aplicativo se torna um ambiente único com várias opções de compras para seus clientes, sendo assim, um Shopping Virtual, onde o cliente tem a opção de visualizar várias lojas.

A dificuldade encontrada no desenvolvimento do trabalho foi a incompatibilidade entre as versões 2 e 3 do Silverlight. Inicialmente o projeto foi desenvolvido utilizando a versão 2 da tecnologia, com o decorrer do tempo surgiu a atualização para a versão 3, com esta atualização o aplicativo deixou de funcionar, sendo desenvolvido novamente utilizando a versão 3.

Como sugestão para trabalhos futuros ficam as melhorias no Gerenciador Online de Lojas Virtuais – GonLo, ou seja, a implementação de novas funcionalidades, tais como a integração do gerenciador com cartões crédito e bancos para concretização da compra, pois este aplicativo foi desenvolvido para demonstrar os estudos feitos em relação a nova tecnologia Microsoft Silverlight, e a implementação de módulos para tornar o aplicativo desenvolvedor de uma única loja virtual, ou seja, o proprietário da loja virtual terá a opção de deixar sua loja em um shopping virtual ou a opção de criar uma loja separadamente das outras no gerenciador.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTIN, A. L. **Comércio eletrônico**: modelo, aspectos e contribuições de sua aplicação. São Paulo: Atlas, 2004.

ALLAIRE, J. **Macromedia Flash MX**: a next-generation rich client. Disponível em: <<http://www.adobe.com/devnet/flash/whitepapers/richclient.pdf>>. Acesso em: 17 mar. 2009.

CHEN, L. **Consumer acceptance of virtual stores**: a theoretical model and critical success factors for virtual stores. 2000. 144 p. Tese (Doutorado) – Universidade de Memphis, 2000.

CLEMENTE, A. **Projetos Empresariais e Públicos**. São Paulo: Atlas, 1998.

COAD, P. Object-Oriented Patterns. **Communications of the ACM**, v. 35, n. 9, set. 1992, p. 152-159.

COMÉRCIO eletrônico. In: **WIKIPÉDIA**: a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/E-commerce>>. Acesso em: 31 mar. 2009.

DAYLEY, B.; DAYLEY, L. D. **Silverlight 2 Bible**. Indianapolis: Wiley Publishing, Inc., 2008.

DUHL, J. Rich Internet Applications White Paper. In: **IDC Opinion**. Disponível em: <http://www.adobe.com/platform/whitepapers/idc_impact_of_rias.pdf>. Acesso em: 19 maio 2009.

DURÃES, R. Introdução ao ASP .NET. In: SPAKI, E. et al. **Desenvolvendo para Web usando Visual Studio 2008**. Rio de Janeiro: Brasport, 2008.

FELIPINI, D. **Loja virtual**: como vender na Internet. [S.l. : s.n.], [ca. 1999].

FOWLER, M. et al. **UML essencial**: um breve guia para a linguagem-padrão. Porto Alegre, RS: Bookman, 2005.

GUEDES, G. T. A. **UML**: uma abordagem prática. 2. ed. São Paulo: Novatec, 2006.

GUEDES, G. T. A. **UML 2: guia prático**. São Paulo: Novatec, 2009.

JOHNSON, R. **How to develop frameworks**: tutorial notes of ECOOP 96. Austria: Linz, 1996.

JOHNSON, R.; FOOTE, B. Designing Reusable Classes. **Journal of Object Oriented Programming – JOOP**, v. 1, n. 2, jun./jul. 1988, p. 22-35.

KALAKOTA, R.; WHINSTON, A.B. **Frontiers of electronic commerce**. United States: Addison-Wesley Publishing Co. Inc. ,1996.

KONO, R. Desenvolvendo aplicações ricas para web usando Silverlight 2.0. In: SPAKI, E. et al. **Desenvolvendo para Web usando Visual Studio 2008**. Rio de Janeiro: Brasport, 2008.

LOJAS Virtuais. In: **E-COMMERCE. ORG**: tudo sobre comércio eletrônico. Disponível em: <http://www.e-commerce.org.br/lojas_virtuais.php>. Acesso em: 18 mar. 2009.

MAYA, P. C.; OTERO, W. I. **A Influência do consumidor na era da internet**. Revista da FAE, v.5, n.1, jan./abr. 2002, p.71-81.

MORITZ, F. **Rich Internet Applications (RIA): a convergence of User Interface Paradigms of Web and Desktop Exemplified by JavaFX**. 2008. 140 p. Monografia (Graduação) - Universidade de Ciências Aplicadas Kaiserslautern, Zweibrücken, Alemanha, 2008.

MSDN. **Biblioteca MSDN**. Disponível em: <<http://msdn.microsoft.com/en-us/library/default.aspx>>. Acesso em: 10 maio 2009.

PAGSEGURO. In: **Guia do Desenvolvedor PagSeguro**. Disponível em: <<http://visie.com.br/pagseguro>>. Acesso em: 17 jun. 2009.

PONTES, J. O. **Aplicações ricas para internet**. 2008. 47 p. Monografia (Graduação) - Fundação Educacional do Município de Assis. Assis, São Paulo, 2008.

ROGERS, G. F. **Framework-Based Software Development in C++**. New Jersey: Prentice-Hall, 1997.

SHAW, M. J. Eletronic Commerce. **Review of Research**. Ineus Information Sipstens Frontius, 1999.

SILVA, R. P. **Suporte ao Desenvolvimento e uso de Frameworks e Componentes**. 2000. 262 p. Tese (Doutorado) – Universidade Federal do Rio Grande do Sul. Porto Alegre, Rio Grande do Sul, 2000.

SPAKI, E. et al. **Desenvolvendo para web usando Visual Studio 2008**. Rio de Janeiro: Brasport, 2008.

STADZISZ, P. C. **Projeto de Software usando a UML**. 2002. Centro Federal de Educação Tecnológica do Paraná.

W3C (World Wide Web Consortium). **Padrões Web**. Disponível em: <<http://www.w3.org/>>. Acesso em: 23 maio 2009.

YAMASHITA, S. S.; GOUVÊA, M. A. Impactos e implicações da internet sobre o marketing de relacionamento. **Anais do Encontro Anual da Associação Nacional de Programas de Pós-Graduação em Administração**, 2004. Curitiba.