

CENTRO UNIVERSITÁRIO UNIFACVEST  
CURSO DE CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE CONCLUSÃO DE CURSO  
MARCO AURÉLIO CEVEY

**Business Card: O contato entre colaboradores facilitado**

LAGES

2013

MARCO AURÉLIO CEVEY

**Business Card: O contato entre colaboradores facilitado**

Projeto apresentado à Banca Examinadora do Trabalho de Conclusão do Curso de Ciência da Computação para análise e aprovação.

LAGES

2013

MARCO AURÉLIO CEVEY

**Business Card: O contato entre colaboradores facilitado**

Trabalho de Conclusão do Curso de Ciência da Computação apresentado ao Centro Universitário UNIFACVEST com parte dos requisitos para obtenção do título de bacharel em Ciência da Computação.

Prof. Msc. Márcio José Sembay

Lages, SC \_\_\_\_/\_\_\_\_/2013. Nota\_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

---

LAGES

2013

## **EQUIPE TÉCNICA**

### **Acadêmico**

Marco Aurélio Cevey

### **Professor Orientador**

Profº. Márcio José Sembay

### **Coordenador de TCC**

Profº. Márcio José Sembay

### **Coordenador do Curso**

Profº. Márcio José Sembay

Dedico,

À minha família, por sempre me apoiar e estar por perto toda a minha vida, desde o âmbito acadêmico ao pessoal, e por me dar força para nunca desistir dos meus sonhos, objetivos e ambições.

## **AGRADECIMENTOS**

A Deus, por sempre me dar forças, iluminar e me preparar para eu poder alcançar meus objetivos, tal como, viver plenamente meus ideais.

A minha família, pelo apoio, pela compreensão, oportunidade, atenção e por fazer do impossível, o possível para que eu possa ter todas as oportunidades, como esta de me tornar um cientista da computação.

Aos meus amigos, que são verdadeiros companheiros, pois me suportaram nos momentos mais difíceis desta jornada, e juntos, superamos todos os obstáculos que se apresentaram diante nós.

Aos professores, que se dedicaram a transmitir os seus conhecimentos, tal como, compartilhar a sua própria vida, para que nós possamos viver melhor a nossa.

Aos profissionais que disponibilizaram o seu tempo, conhecimento e experiências de forma verbal ou documentada e por todos os meios que a atualidade permite, pois foram essenciais para o desenvolvimento deste trabalho, como o meu próprio.

A tecnologia e a ciência, por permitir que tudo o que escolhi fazer na minha vida seja possível.

*“Make things as simple as possible, but not simpler.”*

“Faça tudo da forma mais simples o possível, mas não de forma simplista.”

Albert Einstein

## SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS .....	11
LISTA DE FIGURAS .....	13
LISTA DE QUANDROS E TABELAS .....	15
RESUMO .....	16
ABSTRACT .....	17
1 INTRODUÇÃO .....	18
1.1 Justificativa .....	19
1.2 Importância .....	20
1.3 Objetivo do Trabalho .....	20
1.3.1 Objetivo Geral .....	20
1.3.2 Objetivos específicos.....	21
1.4 Metodologia .....	21
1.4.1 Estudo de Caso .....	21
1.4.2 Estudo Bibliográfico .....	23
1.4.3 Cronograma .....	23
1.5 Estrutura do Trabalho.....	24
2 FUNDAMENTAÇÃO TEÓRICA.....	25
2.1 Sistemas de Informação .....	25
2.2 Sistema Web.....	26
2.2.1 Características do Processo de Sistemas Web .....	27
2.2.2 Principais vantagens do Sistema Web.....	27
2.2.3 Principais Desvantagens do Sistema Web .....	29
2.3 Microsoft .Net Framework.....	30
2.4 Linguagem de Programação C#.....	31
2.5 ASP.NET.....	33
2.5.1 Web Forms .....	33
2.5.2 Composição de uma Página .....	36
2.5.3 Funcionamento .....	37

2.5.4	Funcionamento dos Eventos .....	38
2.6	Microsoft Visual Studio 2010 .....	41
2.7	HTML.....	42
2.8	Java Script .....	44
2.9	Serviço do Windows .....	45
3	PROJETO.....	46
3.1	Diagrama de Classes .....	48
3.2	Diagrama de Caso de Uso .....	50
3.3	Diagrama de Entidade Relacionamento .....	51
3.4	Interfaces do sistema Business Card.....	52
3.4.1	Visão Funcionários.....	52
3.4.2	Visão Fotos.....	54
3.4.3	Visão Cartão de Negócio.....	55
3.4.4	Visão Pesquisa.....	56
3.4.5	Visão Logon .....	57
3.4.6	Visão Editar Dados.....	58
3.4.7	Visão Alterar Foto .....	59
3.4.8	Visão Editar Foto .....	60
3.4.9	Visão Logon Administrativo.....	61
3.4.10	Visão Opções do Administrador .....	62
3.4.11	Visão Novo Cadastro .....	63
3.4.12	Visão Editar Cadastro .....	64
3.4.13	Visão Remover Cadastro .....	65
3.4.14	Visão Novos Colaboradores.....	66
3.4.15	Visão Aniversariantes .....	67
3.5	E-mails enviados pelo sistema .....	68
3.5.1	E-mail de Boas-vindas ao colaborador .....	68
3.5.2	E-mail de Novos Colaboradores .....	69
3.5.3	E-mail dos Aniversariantes do Mês .....	70
3.5.4	E-mail de Feliz Aniversário ao Colaborador .....	71
3.5.5	E-mail de Cadastro Incompleto.....	72

4	CONSIDERAÇÕES FINAIS.....	73
5	REFERÊNCIAS BIBLIOGRÁFICAS.....	74
6	ANEXOS.....	77
6.1	Anexo A - ValidationExceptions.xml.....	77
6.2	Anexo B – Código do Serviço do Windows de envio de e-mails automático .....	77

## LISTA DE ABREVIATURAS E SIGLAS

C# – *CSharp*, linguagem de programação

TI – Tecnologia da Informação

ASP – *Active Server Pages*

ASP.Net – Plataforma integrada ao *Framework .NET* para o desenvolvimento de aplicações *web*.

S/A – Sociedade Anônima

TCC – Trabalho de Conclusão de Curso

POO – Programação Orientada a Objetos

SQL – *Structured Query Language*

IDE – *Integrated Development Environment*

SGBD – Sistema Gerenciador de Banco de Dados

IIS – *Internet Information Services*

RAM – *Random Access Memory*

SO – Sistema Operacional

CPU – *Central Processing Unit*

SI – Sistemas de Informação

HTML – *HyperText Markup Language*

HTTP – *Hypertext Transfer Protocol*

XML – *Extensible Markup Language*

CLR – *Common Language Runtime*

MSIL – *Microsoft Intermediate Language*

VB – *Visual Basic*

C – Linguagem de programação

C++ – Linguagem de programação

PHP – *Personal Home PageTools*

## LISTA DE FIGURAS

Figura 1 – Código C# para a geração de uma aplicação de linha de comando .....	32
Figura 2 – Execução do código C# para a geração de uma aplicação de linha de comando .....	33
Figura 3 – Estrutura de arquivos de um <i>Web Form</i> .....	34
Figura 4 – Arquivo Default.aspx .....	35
Figura 5 – Arquivo Default.aspx.cs.....	36
Figura 6 – View State do Web Form.....	38
Figura 7 –PostBack da página ASP.NET .....	39
Figura 8 – Processamento da página ASP.NET no servidor .....	40
Figura 9 – Página ASP.NET modificada renderizada .....	41
Figura 10 – Interface do Microsoft Visual Studio 2010.....	42
Figura 11 – Exemplo de código HTML .....	43
Figura 12 – Exemplo de página HTML .....	43
Figura 13 – Script de Java Script.....	44
Figura 14 – Execução do <i>script</i> no navegador .....	45
Figura 15 - Serviço do Windows sendo executado .....	46
Figura 16 – Diagrama de Classe (1) .....	48
Figura 17 – Diagrama de Classe (2) .....	49
Figura 18 – Caso de Uso .....	50
Figura 19 – Diagrama de Entidade Relacionamento .....	51
Figura 20 – Visão Funcionários.....	53
Figura 21 – Visão Fotos .....	54
Figura 22 – Visão Cartão de Negócios .....	55
Figura 23 – Visão Pesquisa .....	56
Figura 24 – Visão Logon.....	57
Figura 25 - Visão Editar Dados .....	58
Figura 26 – Visão Alterar Fotos .....	59
Figura 27 – Visão Editar Foto .....	60
Figura 28 – Visão Logon Administrativo.....	61
Figura 29 – Visão Opções do Administrador .....	62
Figura 30 – Visão Novo Colaborador .....	63

Figura 31 – Visão Editar Cadastro .....	64
Figura 32 – Visão Remover Usuário .....	65
Figura 33 – Visão Novos Colaboradores.....	66
Figura 34 – Visão Aniversariantes .....	67
Figura 35 – E-mail de boas-vindas ao colaborador .....	68
Figura 36 – E-mail de novos colaboradores .....	69
Figura 37 – E-mail de aniversariantes do mês.....	70
Figura 38 – E-mail de feliz aniversário ao colaborador .....	71
Figura 39 – E-mail de cadastro incompleto.....	72

## **LISTA DE QUANDROS E TABELAS**

Tabela 1 – Cronograma do TCC .....	23
------------------------------------	----

## RESUMO

O presente trabalho de conclusão de curso apresenta um sistema que facilita o contato entre os colaboradores no meio de trabalho na empresa NDDigital Software S/A, tal como suprir a necessidade de que a empresa possui de manter um repositório de dados atualizado, com as principais informações dos seus colaboradores, pois quando se fizer necessário contatar algum colaborador por exemplo, ocorra um contato rápido e eficaz entre as partes envolvidas. Para que o Sistema de Informação aqui referido, sempre esteja acessível de maneira rápida e eficiente, foram realizadas pesquisas bibliográficas referentes aos meios de distribuição de *software*, e assim, utilizou-se no sistema o conceito de aplicação *Web* (*sistema acessado com um navegador pela internet ou redes privadas*), e para tal, foi utilizada a plataforma de desenvolvimento *Web* ASP.NET, que é integrada ao *Framework* (*Conjunto de códigos para prover uma funcionalidade genérica*) Microsoft .NET, e a linguagem de programação C#, o que tornou o desenvolvimento e a manutenção do sistema fácil, confiável e flexível.

**Palavras chaves:** Contato facilitado, aplicação *Web*, Microsoft .NET.

## ABSTRACT

This work of completion presents a system that facilitates contact between employees in the work environment in the company NDDigital Software S/A as meet the need that the company has to maintain a data repository updated, with the main information of its employees, because when you do need to contact any employee for example, occurs a rapid and effective contact between the parties involved. For the information system referred to herein, to always be accessible quickly and efficiently, literature searches relating to the distribution of software were performed, and thus the system used the concept of Web application ( system accessed with a web browser by internet or private networks ), and to this end the ASP.NET Web development platform has been used, which is the Framework had integrated ( set of codes to provide a generic ) functionality Microsoft .NET, and C # programming language, which became the development and maintenance of the system easy., reliable and flexible

**Keywords:** Contact facilitated, Web application, Microsoft .Net.

# 1 INTRODUÇÃO

Um dos grandes desafios dos Sistemas de Informação é assegurar a qualidade e agilidade da informação, imprescindível para os usuários que o utilizam.

Do mesmo modo, que atualmente, no mercado globalizado em que as empresas de TI precisam proporcionar suportes aos seus clientes vinte e quatro horas por dia, de modo ágil e confiável, e que para tal qualidade almejada seja alcançada, a empresa tem que manter um repositório atualizado e completo com informações pertinentes aos seus colaboradores, para que quando se faça necessário, ocorra uma comunicação dinâmica e clara entre os colaboradores envolvidos.

A principal vantagem proporcionada pela tecnologia aos Sistemas de Informação é a capacidade de processar um gigantesco número de dados simultaneamente, tornando a disponibilização das informações demandadas, praticamente on-line. Mas de pouco adianta esse potencial, se os sistemas (rotinas, processos, métodos) não estiverem muito bem coordenados e analisados. Informatizar sistemas ruins traz novos problemas e nenhuma solução, além de nublar as possíveis causas dessas falhas. Essa situação infelizmente é bastante comum nas empresas, pois existe uma grande confusão sobre análise de sistemas operacionais/corporativos/informativos e a programação desses sistemas.

Mas com a aplicação das técnicas de engenharia de software é possível construir um software de maior qualidade, que satisfaça todas as necessidades de clientes e usuários, sem apresentar falhas ou erros, e seja fácil de usar, modificar e manter.

Além disso, utilizando padrões de desenvolvimento com a correta tecnologia, a produtividade pode ser aumentada e o tempo reduzido, atendendo às exigências das empresas, e levando à qualidade ao usuário.

Projetar software orientado a objeto é difícil, mas projetar software reutilizável orientado a objetos é ainda mais complicado. Você deve identificar objetos pertinentes, fatorá-los em classes no nível correto de granularidade, definir as interfaces das classes, as hierarquias de herança e estabelecer as relações-chave entre eles. O seu projeto deve ser específico para o problema a resolver, mas também genérico o suficiente para atender problemas e requisitos futuros. Também deseja evitar o reprojeto, ou pelo menos minimizá-lo. (GAMMA et al., 2000).

Dessa forma, o sistema aqui apresentado foi planejado, modelado, desenvolvido e distribuído utilizando tecnologias e meios que possibilitam a entrega eficiente das

funcionalidades atribuídas ao sistema para o usuário final, tal quão genérico o suficiente no seu domínio (O meio e a abrangência em que o Sistema de Informação atua), para atender a possíveis problemas, novas iterações ou projetos futuros da maneira mais eficiente.

## 1.1 Justificativa

Com as novas leis trabalhistas vigentes, a empresa NDDigital Software S/A, que possuía um sistema de cartão ponto próprio desenvolvido em ASP.Net, pelo aumento de funcionários, teve que utilizar o relógio de ponto biométrico, para registrar o ponto dos colaboradores, por este motivo, o sistema antigo foi descontinuado, mas esta mudança acarretou um problema, pelo fato de que o sistema antigo possuía as informações pertinentes aos horários de trabalhos dos colaboradores, como a empresa que o colaborador pertence ( O espaço físico da NDDigital Software S/A, é dividido com mais outras Empresas, sendo estas: N5 Software Limitada, HANT e I5INVEST Empreendimentos), necessitava de um meio mais prático e eficiente para acessar as informações de seus colaboradores.

Para suprir esta demanda, a empresa NDDigital Software S/A necessitava de um repositório contendo todas as informações relativas ao contato dos colaboradores, tal qual a empresa que o colaborador pertence, para otimizar o processo de localização dos colaboradores quando necessários, pois a empresa NDDigital Software S/A possui atuação global no mercado de Tecnologia de Informação e possui suporte vinte e quatro horas para os seus cliente, necessita manter um meio eficiente de contato com todos os seus colaboradores.

De modo que para suprir as necessidades da empresa NDDigital Software S/A, um Sistema de Informação baseado em aplicação *Web*, acessado pela rede interna do grupo de empresas, contendo os contatos dos colaboradores, seu horário de trabalho e empresa de atuação é o mais indicado, pois é de fácil acesso, e permite que os colaboradores possam identificar e localizar outros colaboradores, ou manter as suas próprias informações atualizadas, ou através do sistema, conhecer os outros colaboradores, o que é essencial para uma boa convivência no ambiente de trabalho.

## 1.2 Importância

Na atualidade o mundo vive na era da informação, exigindo das organizações uma gestão estratégica eficiente, a qual pode ser facilitada pela utilização de recursos inteligentes oferecidos pela tecnologia de informação e sistemas de informação.

A tecnologia de informação oferece recursos tecnológicos e computacionais para a geração e gestão de informações, e os sistemas de informação estão cada vez mais sofisticados, propondo mudanças nos processos, estrutura e estratégia de negócios. Não se admite hoje uma empresa de Tecnologia de Informação, que queira competir com vantagem, sem a utilização dessas ferramentas. Estes fatos abrem lacunas para que os gestores, com novas visões busquem o aperfeiçoamento contínuo para suas empresas.

Então por estes e outros motivos, que o sistema aqui proposto e desenvolvido foi destinado, para que todos os colaboradores da empresa NDDigital Software S/A, possam ter um meio confiável, atualizado e ágil de acessar todas as informações seus colegas de trabalho, o que acaba gerando uma melhoria em todas as atividades relativas a localização e contato entre os colaboradores dentro da empresa NDDigital Software S/A.

## 1.3 Objetivo do Trabalho

### 1.3.1 Objetivo Geral

Desenvolver um Sistema de Informação, para o armazenamento, consulta e gestão das informações dos colaboradores da empresa NDDigital Software S/A, distribuído em uma Aplicação *Web*, capaz de cadastrar, consultar, validar e remover as respectivas informações, como apresentar as últimas atualizações das informações do colaboradores e gerar e enviar *e-mails* automaticamente ou através da visão administrativa do sistema.

### 1.3.2 Objetivos específicos

Os objetivos específicos consistem no desenvolvimento de um Sistema de Informação para o armazenando das informações dos colaboradores da empresa NDDigital Software S/A com as seguintes funcionalidades:

- a) Armazenar e gerenciar as informações dos colaboradores da empresa NDDigital Software S/A.
- b) Disponibilizar as informações dos colaboradores para consulta.
- c) Geração e envio de *e-mails* para os colaboradores através da interface administrativa do sistema.
- d) Geração e envio de *e-mails* automatizados para os colaboradores através de uma aplicação de serviço.
- e) Apresentar na interface do sistema as últimas 20 atualizações ocorridas nas informações dos colaboradores.

Assim, o sistema proporcionará uma maneira facilitada de ocorrer o contato de modo geral dos colaboradores da empresa NDDigital Software S/A, tal qual, seja facilitando a localização do contato do colaborador, atualizando os colaboradores com as últimas atualizações das informações dos outros usuários do sistema ou enviando *e-mails* com os novos colaboradores da empresa.

## 1.4 Metodologia

### 1.4.1 Estudo de Caso

O desenvolvimento deste trabalho ocorreu na empresa NDDigital Software S/A, que foi fundada em 2003 na cidade de Lages, tem como atividade principal a criação e desenvolvimento de *Softwares*. Seu crescimento e sucesso ao longo do tempo é resultado da dedicação aos clientes e parceiros, bem como a qualidade de produtos, sobre tudo, ao profissionalismo e competência de toda equipe de trabalho.

A NDDigital Software S/A, opera atualmente no Brasil, América Latina, Estados Unidos da América, Europa e Coreia do Sul; é reconhecida como a maior desenvolvedora de *software* focado em impressão na América Latina.

A empresa possui quatro verticais de negócio, sendo estas:

- **Impressão:** Solução para impressão corporativa, que otimiza a utilização de recursos através do gerenciamento de políticas, normas, métodos e regras voltadas para o ambiente corporativo; com captura de dados, monitoramento e rastreabilidade em tempo real;
- **Documentos eletrônicos:** Soluções que atendem aos processos de emissão, recepção e gestão de documentos eletrônicos, como nota fiscal eletrônica, conhecimento de transporte eletrônico e nota fiscal de serviço eletrônico;
- **Pagamento eletrônico de frete:** Solução para gestão de pagamento eletrônico de frete, coleta e triagem de documentos pertinente ao processo de logísticas das empresas.
- **Gestão da cadeia de suprimentos (HANT):** Solução voltada para o gerenciamento de cadeias de compras de empresas de engenharia, para a construção pesada e para a compra de serviços para transportadoras, reduzindo o custo final da compra.

Para o desenvolvimento do trabalho aqui apresentado foi realizado reuniões com o setor de Recursos Humanos da NDDigital Software S/A para a apuração dos requisitos básicos do sistema responsável para o armazenamento das informações dos colaboradores da empresa em questão. Após este levantamento de requisitos básicos, foi definido que o sistema proposto seria distribuído na rede interna da empresa, pela facilidade de acesso, e a equipe de Recursos Humanos informou quais as informações que seriam mais importantes e detalhes de como os e-mails que o sistema enviaria seriam formatados.

O trabalho foi dividido em quatro etapas, sendo estas:

- 1) Identificação das informações mais pertinentes, para que o sistema facilite o contato entre os colaboradores da empresa junto ao Recursos Humanos;
- 2) Levantamento dos dados atuais de contatos dos colaboradores armazenados em uma planilha, para serem inseridos no sistema aqui apresentado, junto à recepção;

- 3) Obtenção, junto a equipe de Desenvolvimento do antigo sistema de registro de ponto da empresa, das fotos dos colaboradores.
- 4) Análise dos atuais e-mails informativos que eram gerados e enviados manualmente aos colaboradores da NDDigital Software S/A;

### 1.4.2 Estudo Bibliográfico

Para a realização deste trabalho foram pesquisados em livros, periódicos e páginas da *web*, os conceitos de Sistemas de Informação, quais as suas características, tal como o funcionamento do Sistemas *Web*, suas vantagens e desvantagens.

E também pesquisou-se a viabilidade e a praticidade de acesso que os Sistemas *Web* possuem, tal como as facilidades na implantação das aplicações, mas ao mesmo tempo, os desafios que a arquitetura *web* no leva a enfrentar pelo fato que não há padronização de como os navegadores interpretam as páginas HTML.

### 1.4.3 Cronograma

O seguinte cronograma foi utilizado para o desenvolvimento do trabalho aqui apresentado.

Atividades Realizadas	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
Pesquisa						
Revisão Bibliográfica						
Desenvolvimento						
Testes						
Implantação						
Entrega do TCC						
Defesa da Banca						

**Tabela 1 – Cronograma do TCC**  
**Fonte: Próprio autor**

## 1.5 Estrutura do Trabalho

Para o desenvolvimento do trabalho aqui apresentado, foram cumpridas as seguintes etapas: pesquisa de material bibliográfico, revisão bibliográfica, desenvolvimento, testes e implantação.

Na primeira fase aqui relatada, foi realizada uma pesquisa bibliográfica, com a coleta de informações e dados para o início do trabalho. Enquanto que na revisão bibliográfica ocorreu a fundamentação teórica, e proporcionou o embasamento técnico, assim justificando os tópicos abordados no TCC como as tecnologias e meios de distribuição do sistema aqui apresentado.

Para o desenvolvimento e testes do sistema, foi utilizado o *Framework* Microsoft .NET 4.0, mais especificamente o ASP.Net 4.0, que é a plataforma integrada ao *Framework* Microsoft .NET para o desenvolvimento de aplicações *web*, com a linguagem de programação POO C#. A IDE utilizada para a codificação foi o Microsoft *Visual Studio 2010*, o SGBD utilizado foi o Microsoft *SQL Server 2008* e o Navegador utilizado foi o *Internet Explorer 9*.

*Hardware* utilizado:

- Computador *Desktop* para o desenvolvimento e testes;
- Processador Inter® Core™ i3 CPU 540, 3.07 GHz;
- 8 GB de memória RAM;
- 500 GB de disco rígido;

Para a fase de implantação, foi utilizado o SO Microsoft *Windows Server 2003 R2 Enterprise Edition*, e o servidor *web* utilizado foi o IIS 6.0 e o SGBD utilizado foi o Microsoft *SQL Server 2008*.

*Hardware* utilizado:

- Servidor para a aplicação *web* e o SGBD;
- Intel® Xenon® CPU X3430, 2.40 GHz;
- 3 GB de memória RAM;

- 126 GB de disco rígido;

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Sistemas de Informação**

Para Matsuda (2007), Sistemas de Informação são processos administrativos que envolvem processos menores que interagem entre si. O sistema é dividido em subsistemas que podem ser: produção/serviço venda distribuição, materiais, financeiro, recursos humanos e outros, dependendo do tipo de empresa. O departamento de informática da empresa cruza esses subsistemas, o que leva a uma abordagem sistemática integrativa, envolvendo questões de planejamento estratégico da empresa.

De acordo com DALFOVO (2000), a definição sobre informação, é todo o esclarecimento, fornecimento de dados, notas, argumentos, fator qualitativo que designa a posição de um sistema, eventualmente transmissível a outro sistema. Sistemas de Informação (SI) são conjuntos de componentes inter-relacionados que coletam informações, manipulam e difundem estas informações fornecendo, além disto, um mecanismo de opinião. Este mecanismo permite efetuar ajustes nas atividades de entrada e ou processamento.

FREITAS (1992) acrescenta a informação é o produto da análise dos dados existentes nas empresas que transmite conhecimento e pode auxiliar o executivo na tomada de decisão. Conforme DALFOVO (2000), o uso eficaz da informação nas organizações passa a ser um patrimônio, que é considerado um fator chave para o sucesso das organizações. Este fator torna-se mais expressivo quando as organizações se defrontam com as mudanças de mercado e avanços das tecnologias.

Conforme OLIVEIRA (1992), um sistema é considerado um ambiente como sendo um conjunto de informações, sendo que, eventuais mudanças de qualquer ordem podem mudar ou alterar as informações do sistema. Esse, por sua vez, é um conjunto de partes com mútua dependência, que juntas formam um todo para exercer uma determinada função. Portanto, pode-se afirmar que as entradas, processamentos e saídas são partes que formam um sistema.

Segundo RODRIGUES (1996), sem se preocupar com o histórico da evolução dos Sistemas de Informação, pode-se dizer que, a partir de 1985, a informação passou a ser utilizada, mais orientadamente, como recurso estratégico. A partir desta época, os Sistemas de Informação começaram a ser visto como *commodity* pelo sentido e papel a eles atribuídos pelas organizações.

Sistemas de Informação devem apresentar informações claras, sem a interferência de dados que não são importantes e deve possuir um alto grau de precisão e rapidez para não perder sua razão de ser em certos momentos críticos (DALFOVO, 2000).

Conforme REZENDE (2003), sistemas de informação foram divididos de acordo com as funções administrativas, que, a mercê de suas características próprias, foram sendo tratadas de forma individualizada, resultando na criação de vários sistemas para ajudar os executivos nos vários níveis hierárquicos, a tomarem decisões.

## 2.2 Sistema Web

A *Web* é, atualmente, o principal veículo para a prestação destes serviços, permitindo atingir um número cada vez maior e mais diversificado de usuários e isto torna a demanda por sistemas baseados na *Web* cada vez maior e urgente, um sistema ou aplicação *Web* é desenvolvido para adicionar funcionalidades de negócio para alcançar o objetivo deste determinado negócio (CONALLEN, 2003).

Nos seus termos mais simples, uma aplicação *Web* é um sistema que permite a seus usuários executar a regra ou lógica do negócio com um navegador *Web*. Existe uma distinção sutil entre um sistema *Web* e um simples *Site Web*. Um sistema *Web* é um site em que a entrada do usuário no sistema afeta o estado do negócio. Essencialmente, um sistema baseado na *Web* usa um *Site Web* como o *front end* para aplicação de negócio. Em quase todas as aplicações *Web* mais simples, o usuário precisa fornecer mais do que apenas informações de solicitação de navegação. Normalmente, os usuários de um sistema *Web* inserem uma gama variada de dados, textos simples, seleções de caixas de seleção ou, até mesmo, informações binárias ou de arquivo. Esses dados serão armazenados e de alguma forma o sistema vai ter um estado diferente em algum determinado lugar quando o usuário terminar de usar o sistema, ou seja, o usuário afetou de alguma forma o estado do negócio.

A arquitetura de um sistema *Web* é simples e direta. Ela contém os mesmos componentes principal de um *Site Web*: um servidor *Web*, uma conexão de rede e navegadores cliente. A inclusão do servidor permite que o sistema gerencie o estado e a regra do negócio.

### **2.2.1 Características do Processo de Sistemas Web**

Segundo PRESSMAN (2002) as seguintes características guiam o processo de sistemas baseados na *Web*:

a) Imediatismo: aplicações baseadas na *Web* têm um imediatismo que não é encontrado em nenhum outro tipo de software. Isto é, o prazo de colocação no mercado e disponibilização de novas informações de um site pode ser uma questão de semanas.

Os desenvolvedores precisam usar métodos para planejamento, projeto, implementação e testes;

b) Segurança: como as aplicações *Web* estão disponíveis através de acesso à rede, é difícil limitar a população de usuários finais que podem ter acesso à aplicação. A fim de proteger o conteúdo reservado e fornecer modos seguros de transmissão de dados, boas medidas de segurança precisam ser implementadas na infraestrutura da aplicação propriamente dita;

c) Estética: uma inegável parte da atração de uma aplicação ou sistema *Web* é o seu aspecto. Quando uma aplicação é projetada para o mercado, para vender produtos e ideias, ou para que os usuários se sintam bem e fiquem à vontade no uso da aplicação a estética pode ter tanto a ver com o sucesso quanto o projeto técnico.

### **2.2.2 Principais vantagens do Sistema Web**

A seguir estão citadas algumas vantagens da utilização dos sistemas que são baseados na plataforma *Web* (CONALLEN, 2003):

a) Sistemas na *Web* podem ser executados a partir de qualquer navegador da internet;

b) Sistemas baseados na *Web* podem ser acessados de qualquer lugar do mundo, para isto basta apenas o usuário possuir um computador com conexão à internet e um navegador, pois o sistema *Web* fica acessível em qualquer computador seja na rede local ou na *Web* facilitando aos usuários o acesso ao aplicativo onde quer que estejam. Imagine um cenário onde o gerente de um determinado departamento esteja de férias e este departamento necessite solicitar ao almoxarifado uma mesa e algumas cadeiras novas e que apenas o gerente tem autorização para proceder com tal pedido. Basta este gerente conectar na internet através de um *cyber* café ou escritório virtual e efetuar o pedido entrando com sua senha. Ou seja, ele não precisa passar sua senha para outro funcionário, não precisa estar fisicamente na empresa e não precisa delegar este tipo de poder temporariamente a ninguém.

c) Interface HTML reconhecida por uma grande gama de usuários já acostumados com o funcionamento dos navegadores;

d) Atualização dos dados e informações em tempo real para todos os usuários do sistema;

e) Desenvolvimento, manutenção e atualização centralizada da aplicação. Não é necessário sair instalando o sistema em diversos equipamentos diferentes. Basta colocá-lo no servidor para que os usuários obtenham acesso, gerando minimização dos custos, pois em qualquer situação, seja de atualização e ou alteração do sistema basta fazer apenas no servidor e a partir de então todos os usuários do sistema desfrutarão das mudanças efetuadas;

f) Redução dos custos de comunicação, sendo que se existem escritórios dispersos e o sistema de informação se baseia na Internet, o custo em conversações pode ser substancialmente reduzido;

g) Exportação de dados entre usuários remotos usando o protocolo HTTP é mais simples e mais fácil do que usar outro protocolo;

h) Não é exigida muita memória e nem poderosos processadores para a execução do sistema nos terminais, pois o sistema é todo executado no servidor;

i) Escalabilidade no processamento. Se houver necessidade de aumentar o poder de processamento, basta fazer isto no servidor;

j) Pode proporcionar potencialmente, uma melhor produção para a empresa, pois o funcionário pode trabalhar em qualquer local onde o acesso à Internet é disponibilizado.

### 2.2.3 Principais Desvantagens do Sistema Web

A seguir estão citadas algumas desvantagens da utilização dos sistemas que são baseados na plataforma Web (CONALLEN, 2003):

a) Não há uma padronização entre os diversos navegadores e o sistema poderia ser exibido de uma maneira diferente dependendo do navegador e da versão deste navegador também;

b) A entrada de uma grande massa de dados é prejudicada na interface HTML, pois não existe uma maneira padrão de criar máscaras de entrada de dados;

c) Tempo de processamento da execução das tarefas depende da velocidade da conexão, entre cliente e servidor;

d) Os sistemas baseados na *Web* dependem dos recursos do navegador usado para visualizar a aplicação. Como eles possuem recursos diferentes, existem dificuldades para prever como a aplicação vai se comportar;

e) Desenvolver páginas dinâmicas e formulários com interface HTML para entrada de dados é muito mais trabalhoso e complicado que em aplicações comuns;

f) A manipulação das variáveis é um trabalho muito mais complicado, tendo em vista a possibilidade que o usuário tem de abrir e fechar janelas e “navegar” para onde bem entender;

g) Desenvolvimento mais complicado, pois envolvem três camadas onde é necessário o servidor de banco de dados, servidor de aplicativos (regras de negócios) que será o servidor de Internet e o *front end* com diversas validações no próprio navegador;

h) Interface HTML não é rica em controles gráficos e peca no quesito posicionamento. O visual da aplicação pode não ficar tão elegante em relação ao momento do desenvolvimento;

i) Difícil gerenciamento do estado do cliente no servidor. A natureza sem conexão das comunicações do cliente e do servidor não proporciona um modo fácil do servidor controlar a solicitação de cada cliente e associá-la à solicitação anterior, visto que cada e toda solicitação de página *Web* estabelece e, em seguida, interrompe um conjunto completamente.

## 2.3 Microsoft .Net Framework

Microsoft .NET (comumente conhecido com .NET *Framework* em inglês dotNet) é uma iniciativa da Microsoft em que visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações. Todo e qualquer código gerado para .NET, pode ser executado em qualquer dispositivo que possua um framework de tal plataforma. (Disponível em: <<http://social.msdn.microsoft.com>>. Acesso em 20/11/2013).

Com ideia semelhante à plataforma Java, o programador deixa de escrever código para um sistema ou dispositivo específico, e passa a escrever para a plataforma .NET.

A plataforma .NET, é executada sobre uma CLR (*Common Language Runtime*) interagindo com uma Coleção de Bibliotecas Unificadas, que juntas são o próprio framework. Esta CLR é capaz de executar, atualmente, mais de vinte diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem.

A Arquitetura .NET, A plataforma .NET se baseia em um dos princípios utilizados na tecnologia Java (*Just In Time Compiler - JIT*), os programas desenvolvidos para ela são duplo-compilados, ou seja são compilados duas vezes, uma na distribuição e outra na execução.

Um programa que é escrito em qualquer das mais de vinte linguagens de programação disponíveis para a plataforma .NET, sendo que a linguagem mais comum atualmente é o C# (*CSharp*), o código fonte gerado pelo programador é então compilado pela linguagem escolhida gerando um código intermediário em uma linguagem chamada MSIL (*Microsoft Intermediate Language*).

Este novo código fonte gera um arquivo chamado de *Assembly*, de acordo com o tipo de projeto:

- a) EXE - Arquivos Executáveis, Programas
- b) DLL - Biblioteca de Funções
- c) ASPX - Página Web
- d) ASMX - Web Service

No momento da execução do programa ele é novamente compilado, desta vez pelo JIT (*Just In Time Compiler*), de acordo com a utilização do programa, por exemplo: Temos um *Web Site* desenvolvido em ASP.NET, ao entrar pela primeira vez em uma página o JIT irá compila-la, nas outras vezes que algum outro usuário acessar esta página, ele usará esta compilação.

Também é possível, através de ferramentas específicas, "pré-compilar" o código para que não se tenha o custo da compilação JIT durante a execução.

## 2.4 Linguagem de Programação C#

De acordo com Sihler, Andrew L., o C# (*CSharp*) é uma linguagem de programação orientada a objetos criada pela Microsoft, faz parte da sua plataforma .Net. A companhia baseou C# na linguagem C++ e Java.

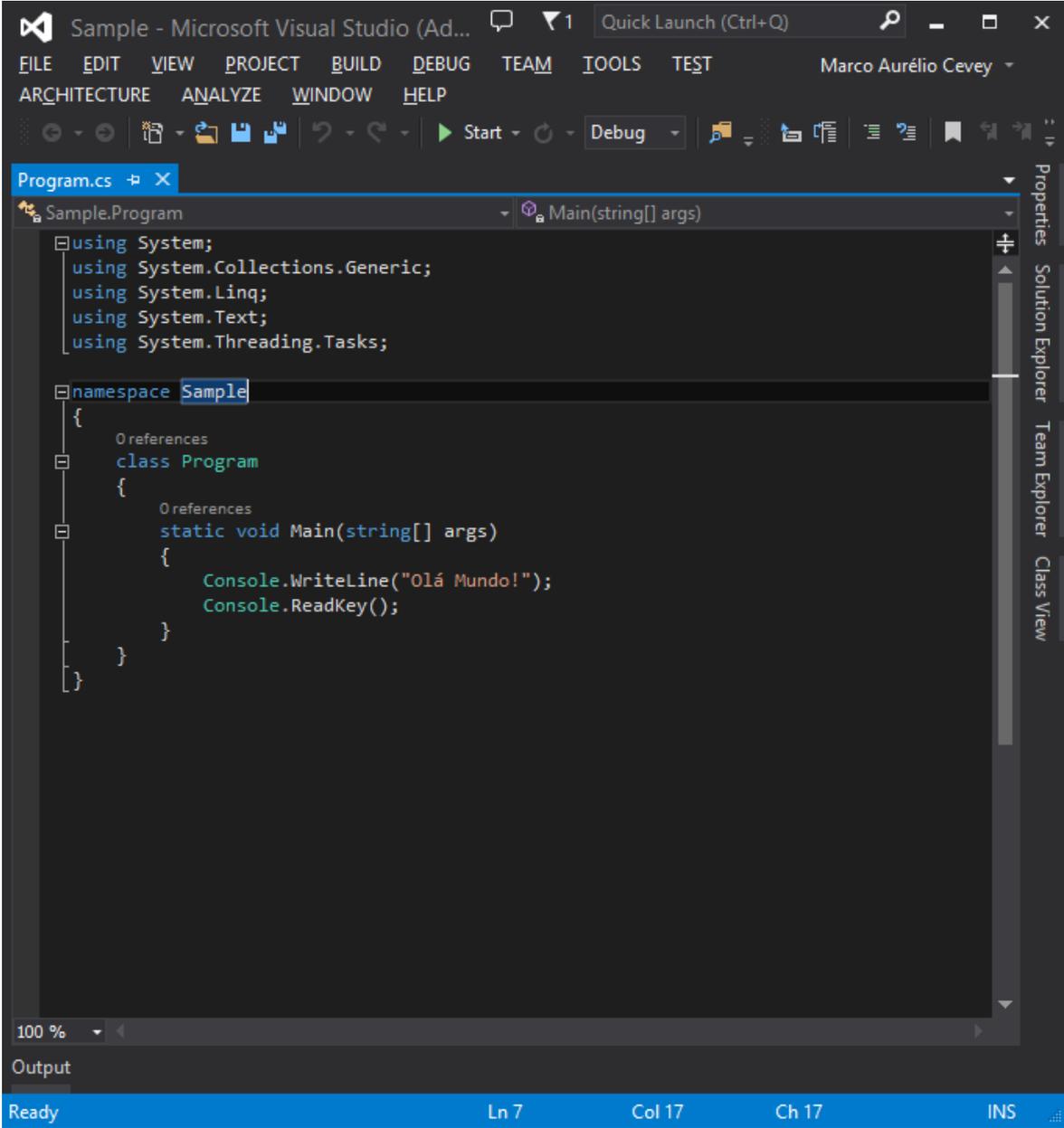
A Microsoft desenvolveu a linguagem C# em conjunto com a arquitetura .NET, sendo assim uma linguagem de programação especialmente desenvolvida para esta plataforma. Sabe-se que a plataforma .NET suporta uma gama de linguagens de programação como C#, VB.NET, J#, C++, Delphi for .NET entre outras, o que torna essa arquitetura produtiva e independente de linguagem, ou seja, o desenvolvedor pode escolher, dentre as linguagens disponíveis, a que mais o agrada.

Essa liberdade de escolha da linguagem de programação também aumenta a produtividade de um time de desenvolvimento, pois enquanto parte dos desenvolvedores estão construindo a interface gráfica em VB .NET um outro grupo pode estar desenvolvendo as classes de negócio com C#, por exemplo. Depois é só integrar as duas partes do projeto.

O C# sofre influência de várias linguagens, mais notadamente do Delphi, C++ e Java. A linguagem C# na verdade é a reunião das principais vantagens dessas linguagens, corrigindo seus defeitos e/ou limitações e acrescentando alguns outros recursos, o que faz

do C# uma linguagem muito poderosa e atrativa aos desenvolvedores que desejam migrar para a plataforma Microsoft .NET, pois é de fácil assimilação.

A figura 1 mostra o código C# para a geração de uma aplicação de linha de comando e a figura 2 mostra a execução deste código.



The image shows a screenshot of the Microsoft Visual Studio IDE. The main window displays a C# file named 'Program.cs'. The code defines a namespace 'Sample' containing a class 'Program' with a static method 'Main'. The 'Main' method prints 'Olá Mundo!' to the console and then reads a key from the console. The IDE interface includes a menu bar (FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, TEST), a toolbar with icons for file operations and execution, and a status bar at the bottom showing 'Ready', 'Ln 7', 'Col 17', 'Ch 17', and 'INS'. The right sidebar shows the 'Solution Explorer', 'Team Explorer', and 'Class View' panes.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Olá Mundo!");
            Console.ReadKey();
        }
    }
}
```

**Figura 1 – Código C# para a geração de uma aplicação de linha de comando**  
Fonte: Próprio autor

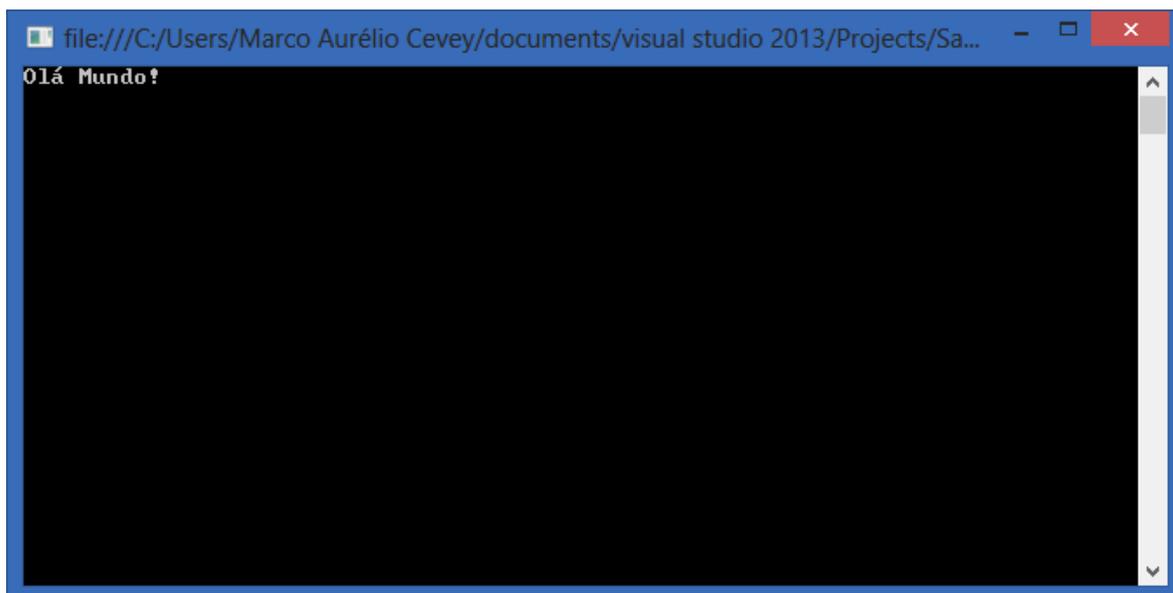


Figura 2 – Execução do código C# para a geração de uma aplicação de linha de comando  
Fonte: Próprio autor

## 2.5 ASP.NET

ASP.NET é a plataforma da Microsoft para o desenvolvimento de aplicações *Web* e é o sucessor da tecnologia ASP. É um componente do IIS que permite através de uma linguagem de programação integrada no *.NET Framework* criar páginas dinâmicas. Não é nem uma linguagem de programação como VBScript, PHP, nem um servidor Web como IIS ou Apache.

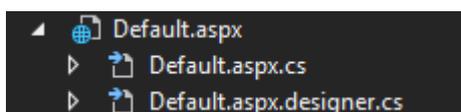
O ASP.NET é baseado no *Framework .NET* herdando todas as suas características, por isso, como qualquer aplicação *.NET*, as aplicações para essa plataforma podem ser escritas em várias linguagens, como C# e Visual Basic *.NET*.

### 2.5.1 Web Forms

As páginas ASP.Net (conhecidas com *Web Forms*) são uma parte vital de uma aplicação ASP.NET, pois são estas páginas que providenciam a páginas HTML para o navegador depois que o cliente fez a requisição.

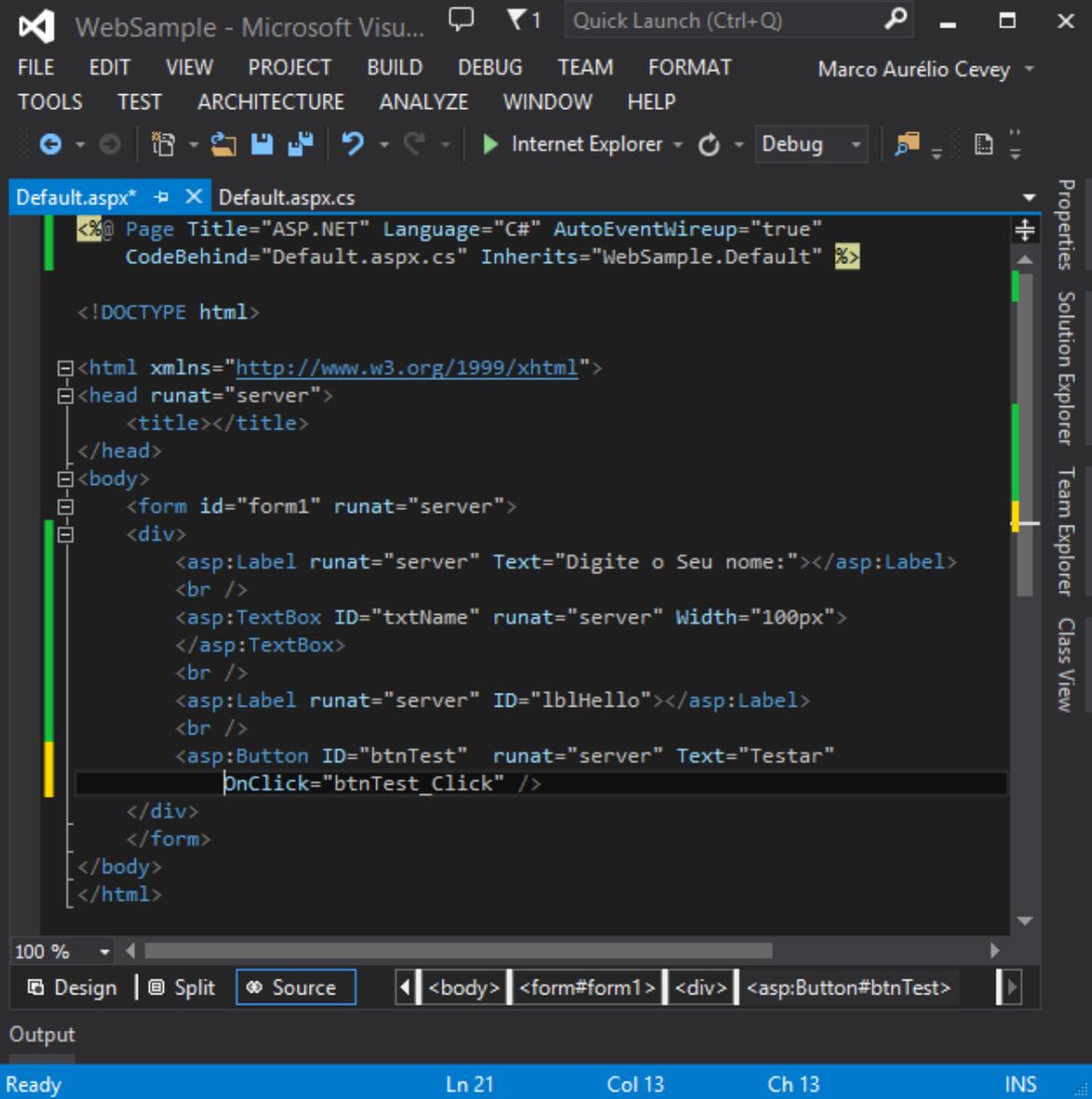
Este conceito de *Web Forms* foi criado para a migração de programadores das tecnologias de cliente (*Windows Forms*) para a *Web*, usando as mesmas facilidades da *interface* Windows, como arrastar botões.

A figura 03 mostra a estrutura de arquivos do *Web Form*, Sendo que o arquivo `Default.aspx` contém a marcação HTML e ASP.NET, enquanto que o arquivo `Default.aspx.cs` possui os códigos C# do *Web Form* que será executado no servidor, e por último, `Default.aspx.designer.cs` contém todas as informações dos controles da página. (Botões, caixas de texto, entre outros):



**Figura 3 – Estrutura de arquivos de um *Web Form***  
**Fonte: Próprio autor**

A figura 4 apresenta o arquivo `Default.aspx` e a figura 5 exhibe o arquivo `Default.aspx.cs`, que são os principais arquivos da estrutura de um *web form*:



The screenshot displays the Microsoft Visual Studio IDE with the file 'Default.aspx' open. The code is as follows:

```
<%@ Page Title="ASP.NET" Language="C#" AutoEventWireup="true"
CodeBehind="Default.aspx.cs" Inherits="WebSample.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label runat="server" Text="Digite o Seu nome:"></asp:Label>
<br />
<asp:TextBox ID="txtName" runat="server" Width="100px">
</asp:TextBox>
<br />
<asp:Label runat="server" ID="lblHello"></asp:Label>
<br />
<asp:Button ID="btnTest" runat="server" Text="Testar"
OnClick="btnTest_Click" />
</div>
</form>
</body>
</html>
```

The interface includes a menu bar (FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, FORMAT, TOOLS, TEST, ARCHITECTURE, ANALYZE, WINDOW, HELP), a toolbar with 'Internet Explorer' and 'Debug' buttons, and a right-hand sidebar with 'Properties', 'Solution Explorer', 'Team Explorer', and 'Class View'. The status bar at the bottom shows 'Ready', 'Ln 21', 'Col 13', 'Ch 13', and 'INS'.

**Figura 4 – Arquivo Default.aspx**  
Fonte: Próprio autor

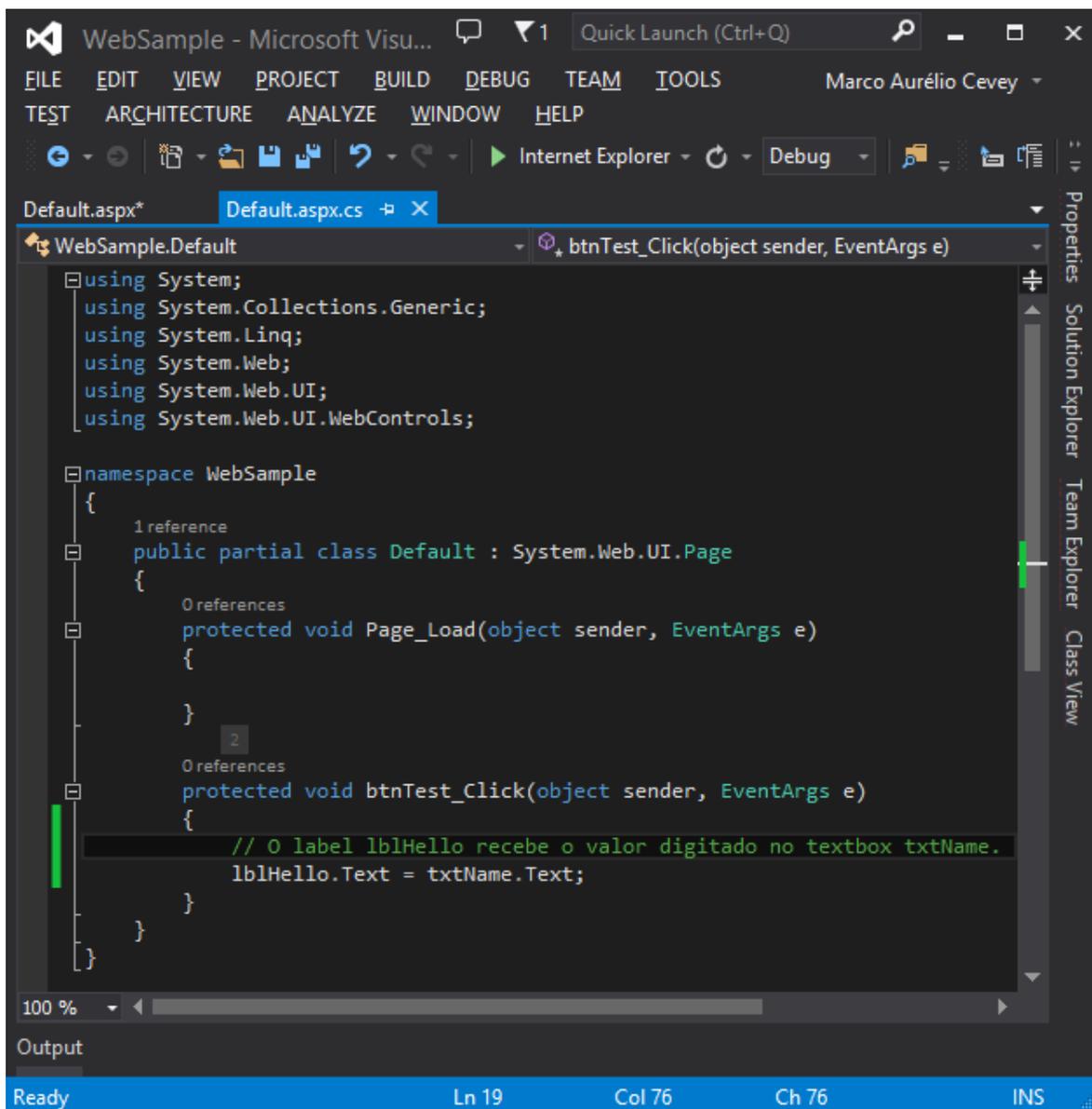


Figura 5 – Arquivo Default.aspx.cs  
Fonte: Próprio autor

## 2.5.2 Composição de uma Página

Um página ASP.NET é:

- Igual a uma página HTML;
- Pode conter HTML, XML e scripts;
- Os scripts serão executados no servidor;

- Tem a extensão “aspx”;
- Todos os controles têm de estar contidos dentro de um “<form>”;
- O “<form>” tem de conter uma *tag* “runat=server”;
- O atributo “runat=server”, significa que o *form* deve ser processado no lado do servidor. Também indica que os controles incluídos nesta *form* estão acessíveis do lado do script do servidor.

### 2.5.3 Funcionamento

As aplicações são executadas no servidor: Imagina-se que uma página que possibilita que os clientes gravem comentários. O usuário faz estas operações no navegador, mas para gravar o comentário precisa-se fazer um *update* na base de dados, para isso o código tem que ser executado no servidor. O ASP.NET trata destes eventos com uma técnica chamada *PostBack*, que envia a página (*Form*) quando certas ações são executadas. Quando o ASP.NET recebe a página, pode executar os eventos correspondentes do lado do servidor.

As aplicações Web são “*stateless*” (não guardam estado): Por outras palavras, assim que uma página é renderizada em HTML, os objetos são destruídos e todas as informações do cliente são descartadas. Este modelo é muito bom para aumentar a escalabilidade e o tráfego das aplicações, mas causa alguns problemas para oferecer uma experiência semelhante a uma aplicação Windows. Para preencher esta lacuna o ASP.NET oferece várias ferramentas, a mais notável é o mecanismo chamado *View State*, que automaticamente coloca informação na página num elemento HTML *hidden field*, para que as informações não sejam perdidas.

O ASP.NET utiliza um modelo de eventos, neste modelo, adiciona-se controles à página e escolhe-se que eventos. Cada “*event handler*” é um método discreto que mantém o código limpo é executado no momento em que o evento é disparado.

A figura 6 exibe a *View State* do *Web Forms*:

```
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"  
value="SWEJ9G41391gTa+TurjVpFnwRFShLcKQQPT5dFrgvnpGfBnexTv2GC5RegU+  
0Hgm7hIwyr5IvkpPZ/WRTfjQXtufJUjB8bu0Av7Dcb93Fr3vJ8EEMai/FtY8CzJk10TY2edjkHfxwn+q13Mwclp5nA==" />
```

**Figura 6 – View State do Web Form**  
**Fonte: Próprio autor**

## 2.5.4 Funcionamento dos Eventos

A página é executada pela primeira vez, o ASP.NET cria a página e os controles associados, o código é executado no servidor, a página é renderizada em HTML e retornada para o cliente. Os objetos são também libertados da memória do servidor.

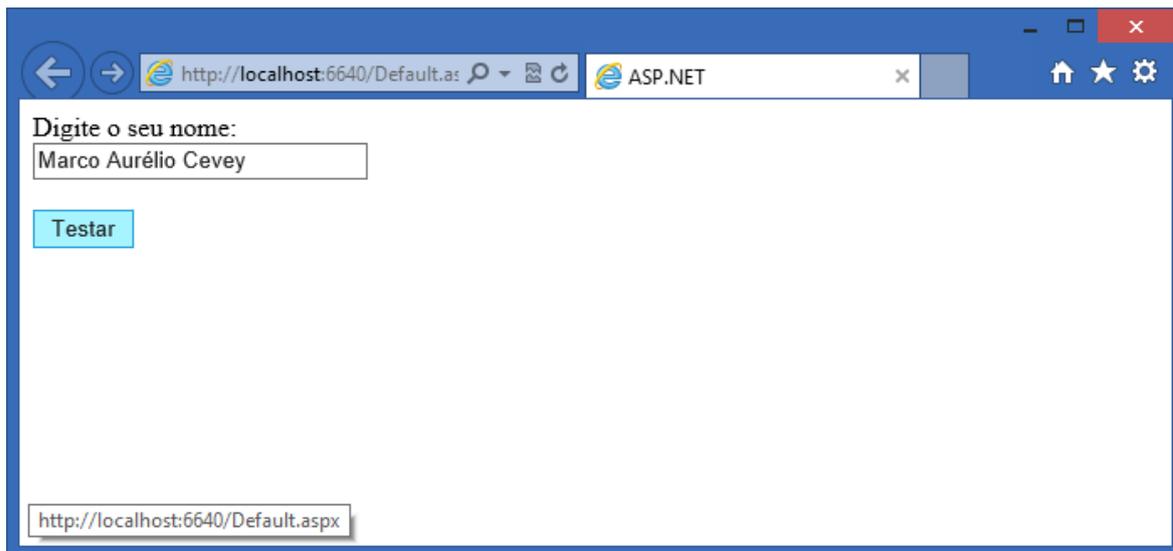
Num determinado ponto o usuário executa uma ação que origina um *PostBack*, como clicar em um botão. Toda a informação da página é enviada para o servidor.

O ASP.NET intercepta a página e cria de novo os objetos no servidor, colocando-os no estado em que estavam a última vez que foram enviados para o cliente.

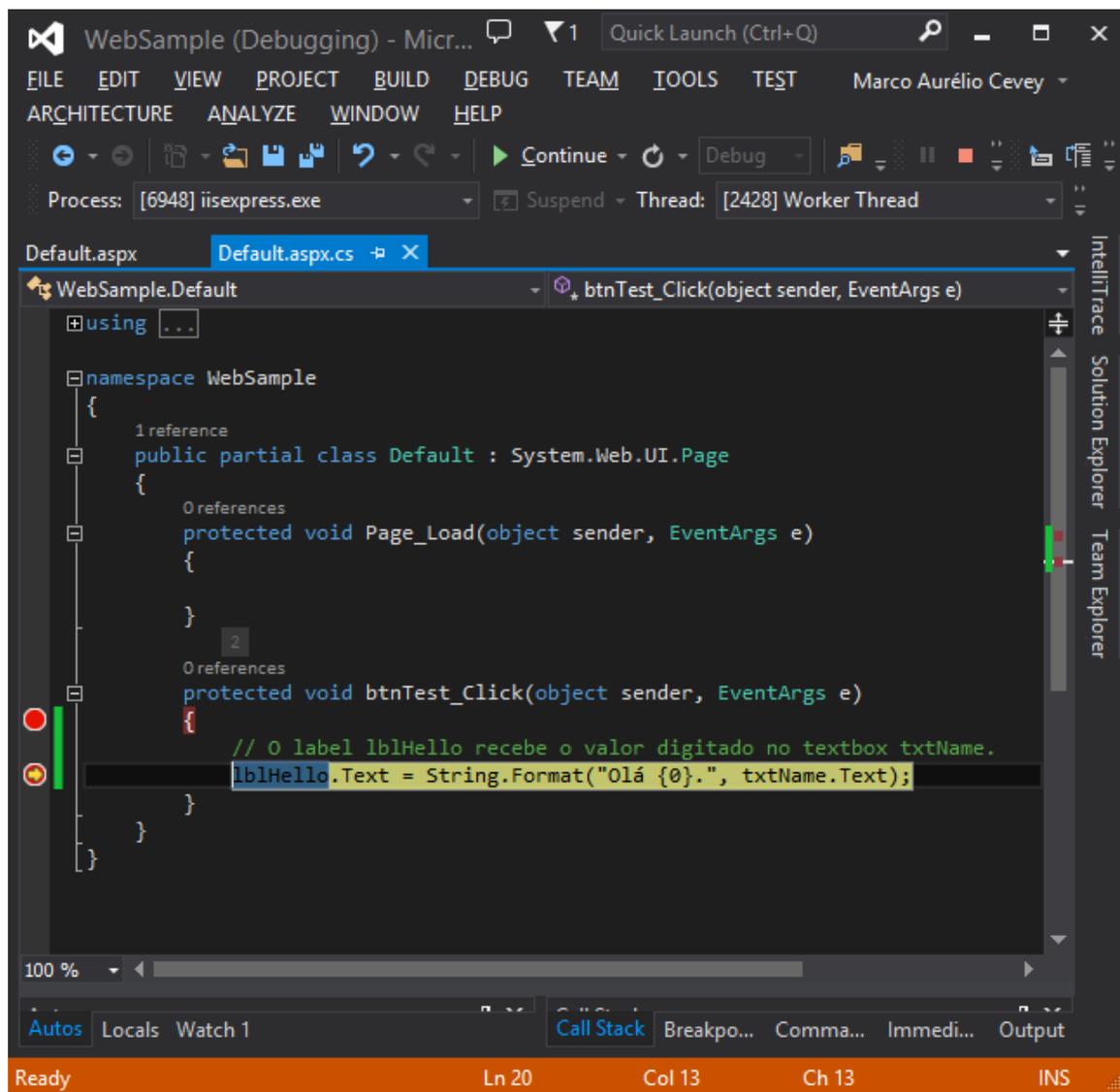
Depois o ASP.NET verifica qual foi o evento no cliente causador do *PostBack* e dispara o evento correspondente no código C# no servidor. Neste momento o ASP.NET executa o código associado ao evento e atualizamos a informação referente aos objetos da página.

A página modificada é renderizada em HTML e devolvida ao cliente com os novos dados e os objetos são retirados da memória do servidor.

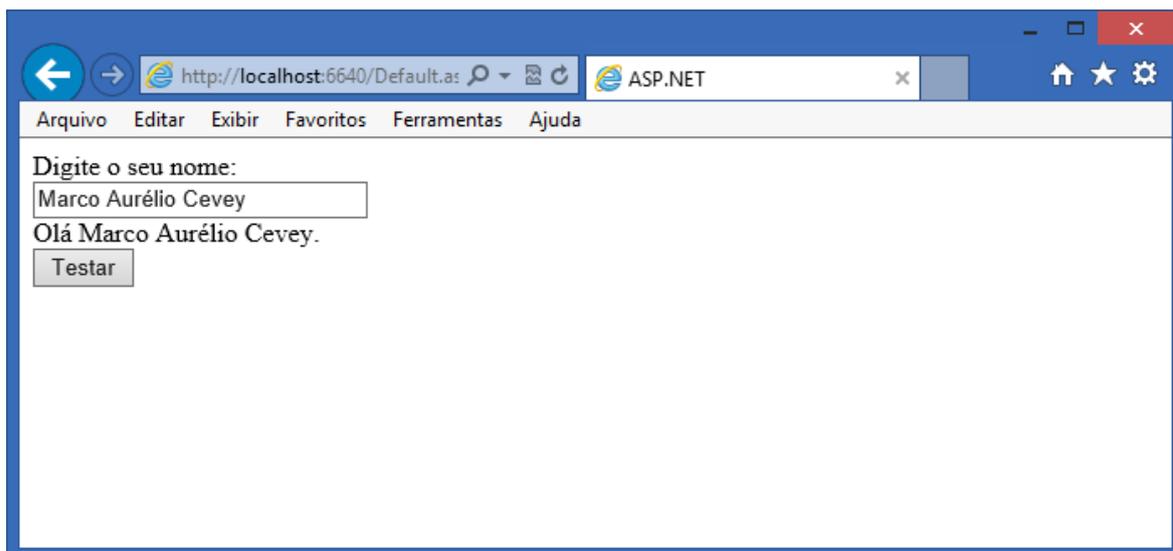
A figura 7 exibe o *PostBack* da página que ocorrerá com o clique do botão “Testar”, para ser processada no servidor o evento atrelado ao clique do botão “Testar”; a imagem 8 mostra o evento do clique do botão “Testar” sendo executado no servidor e pôr fim a figura 9 mostra a página modificada renderizada pelo navegador:



**Figura 7 – PostBack da página ASP.NET**  
**Fonte: Próprio autor**



**Figura 8 – Processamento da página ASP.NET no servidor**  
Fonte: Próprio autor

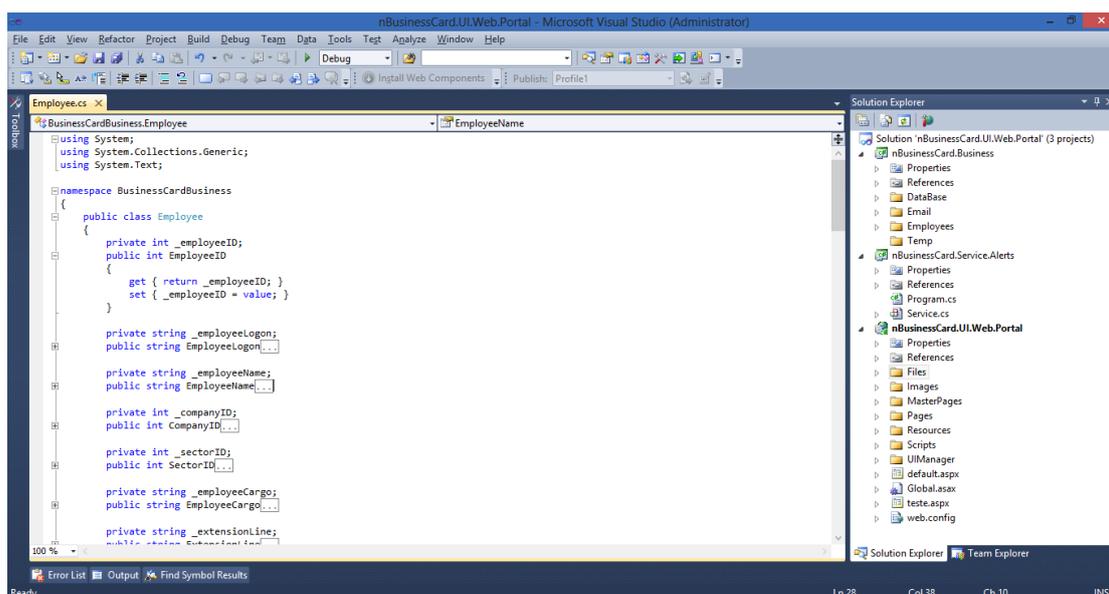


**Figura 9 – Página ASP.NET modificada renderizada**  
Fonte: próprio autor

## 2.6 Microsoft Visual Studio 2010

O Microsoft Visual Studio é um pacote de programas da Microsoft, para desenvolvimento de software, especialmente dedicado, ao framework .NET e às linguagens VB, C, C++, C# e J#. Também é um grande produto de desenvolvimento na área *web*, usando a plataforma do ASP.NET. As linguagens com maior frequência nessa plataforma são: VB.NET e o C#.

A figura 10 apresenta a interface do Microsoft Visual Studio 2010.



**Figura 10 – Interface do Microsoft Visual Studio 2010**  
**Fonte: Próprio autor**

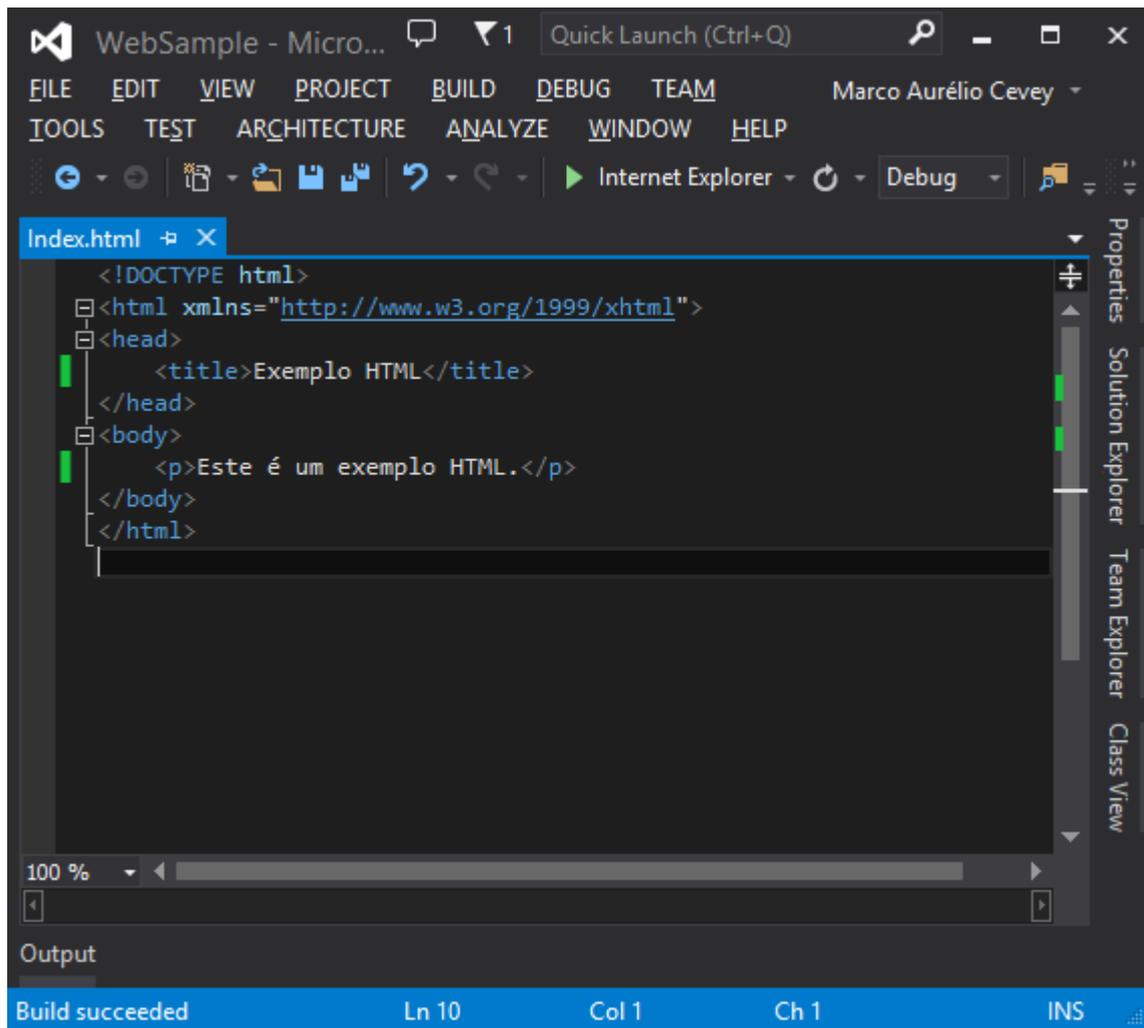
## 2.7 HTML

De acordo com MARINHO (2000), para que as informações possam ser publicadas e distribuídas globalmente, através da Internet, é necessário que se utilize uma formatação que seja entendida pelos diversos computadores e sistemas. Para tanto é necessário que se desenvolva e se adote um padrão; o padrão desenvolvido e adotado na *Web* é o HTML.

HTML significa *Hyper Text Markup Language* (Linguagem de Formatação de Hiper-texto) é a linguagem padrão para apresentação de documentos estruturados na Internet.

Uma página HTML, não passa de um arquivo texto comum (com extensão .HTM ou .HTML), com alguns comandos que são interpretados pelos navegadores, transformando o texto em uma página.

A seguir a Figura 11 apresenta um código HTML e a Figura 12 apresenta a execução do mesmo código no navegador:

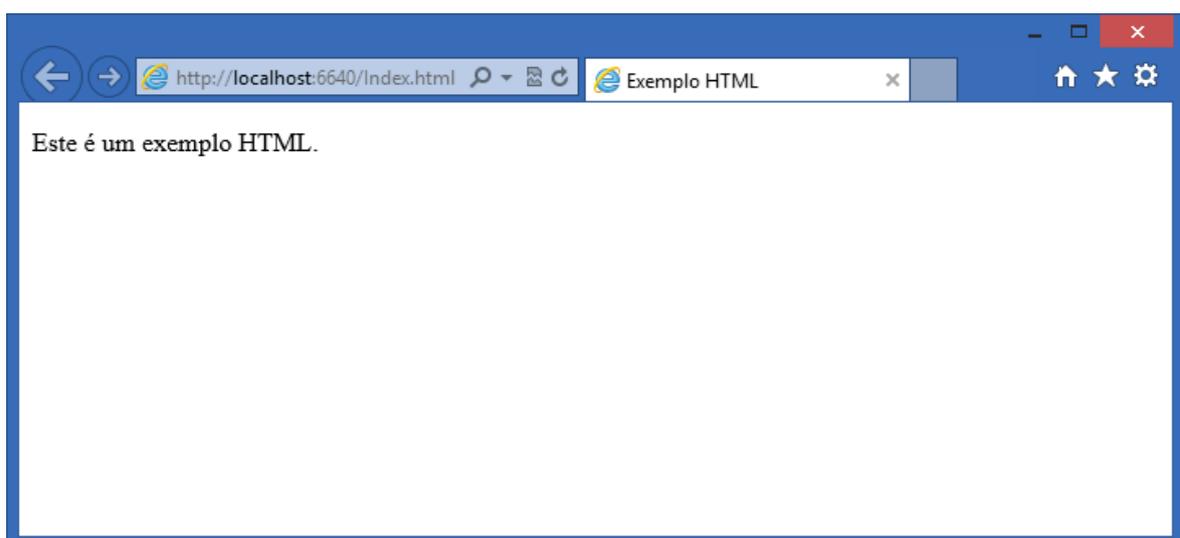


The screenshot shows the Visual Studio Code editor interface. The title bar reads 'WebSample - Micro...' and 'Quick Launch (Ctrl+Q)'. The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, TOOLS, TEST, ARCHITECTURE, ANALYZE, WINDOW, and HELP. The user name 'Marco Aurélio Cevey' is visible. The toolbar shows navigation and execution icons, with 'Internet Explorer' and 'Debug' buttons. The main editor area displays the code for 'Index.html':

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exemplo HTML</title>
</head>
<body>
  <p>Este é um exemplo HTML.</p>
</body>
</html>
```

The status bar at the bottom indicates 'Build succeeded', 'Ln 10', 'Col 1', 'Ch 1', and 'INS'.

**Figura 11 – Exemplo de código HTML**  
Fonte: Próprio autor

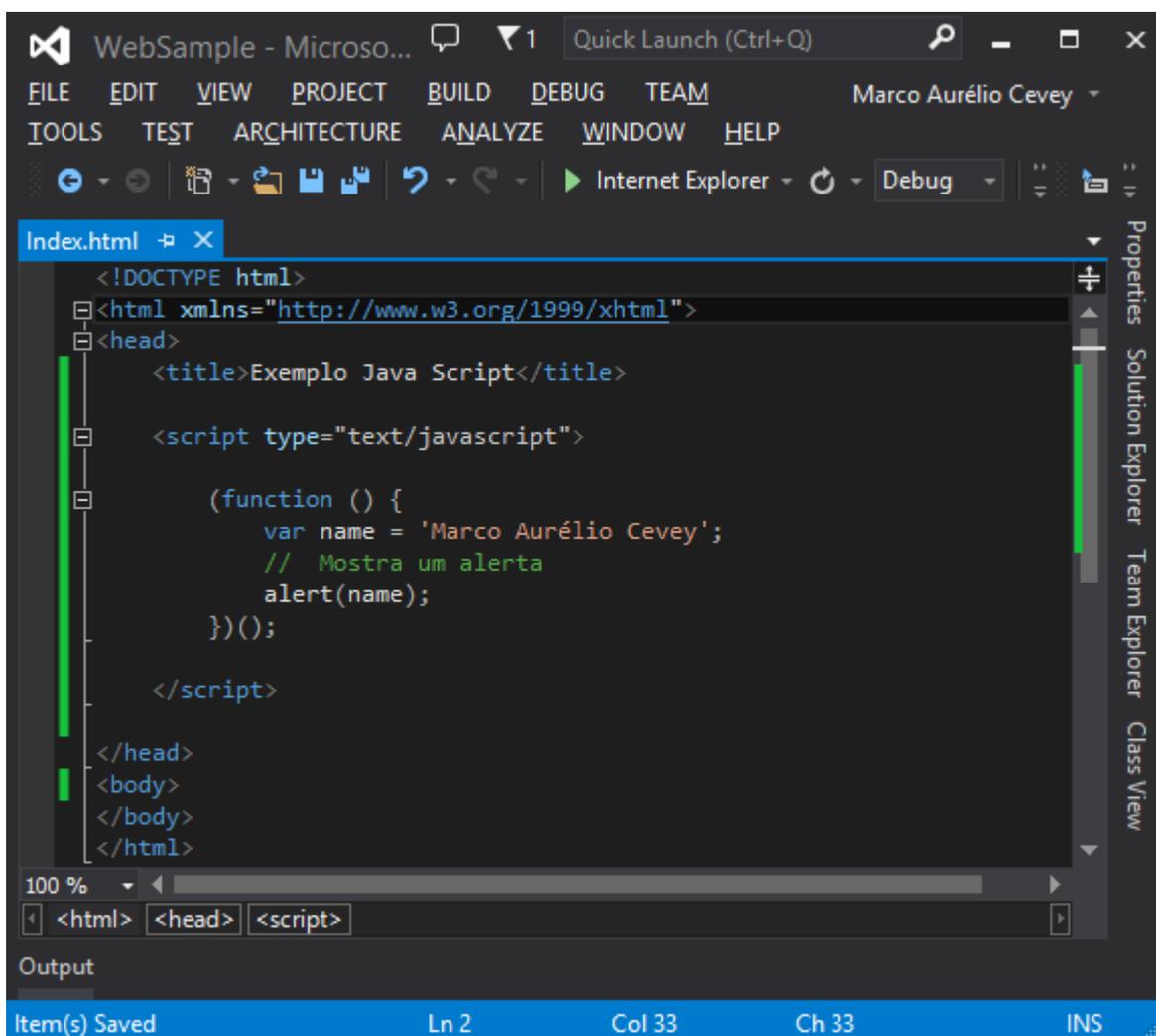


**Figura 12 – Exemplo de página HTML**  
Fonte: Próprio autor

## 2.8 Java Script

Segundo Feather (1997), Java Script é uma linguagem de *script*, e sua sintaxe é bastante similar a C, C+, Pascal e Delphi. Os comandos e funções de Java Script são inseridos dentro de um documento da *Web* junto com outros *tags* HTML. Quando o *browser* de usuário acessa o documento, ele executa o programa ali contido e executa as operações apropriadas.

A figura 13 exibe um *script* de Java Script no documento HTML e a figura 14 exibe a execução do *script* no navegador:

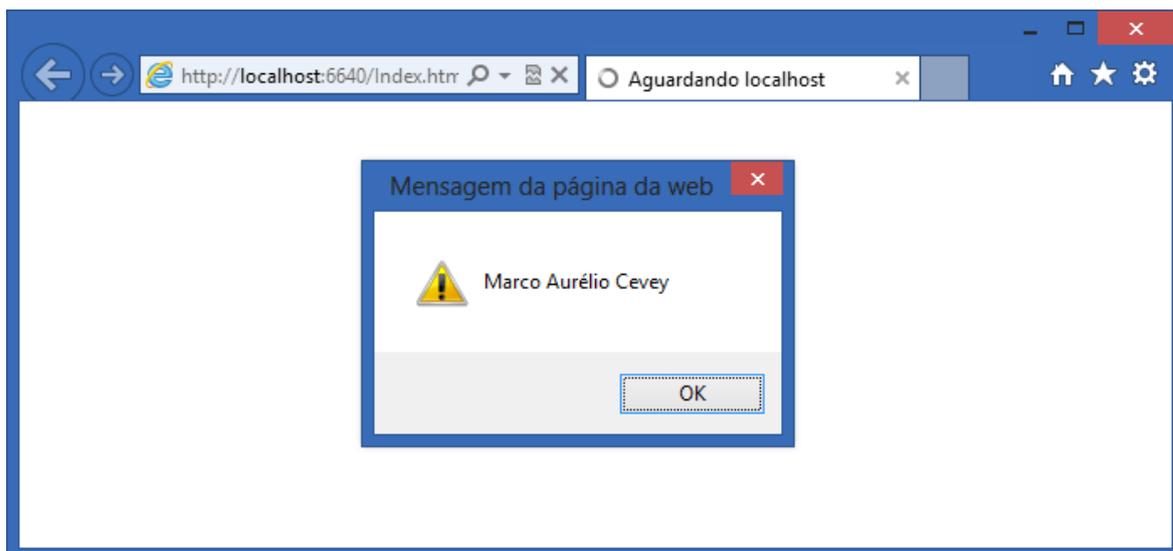


The image shows a screenshot of the Visual Studio Code editor. The main window displays a file named 'Index.html' with the following HTML code:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Exemplo Java Script</title>
  <script type="text/javascript">
    (function () {
      var name = 'Marco Aurélio Cevey';
      // Mostra um alerta
      alert(name);
    })();
  </script>
</head>
<body>
</body>
</html>
```

The editor interface includes a menu bar (FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM), a toolbar with icons for navigation and execution, and a status bar at the bottom showing 'Item(s) Saved', 'Ln 2', 'Col 33', 'Ch 33', and 'INS'. The right sidebar shows the 'Properties', 'Solution Explorer', 'Team Explorer', and 'Class View' panels.

Figura 13 – Script de Java Script  
Fonte: Próprio autor

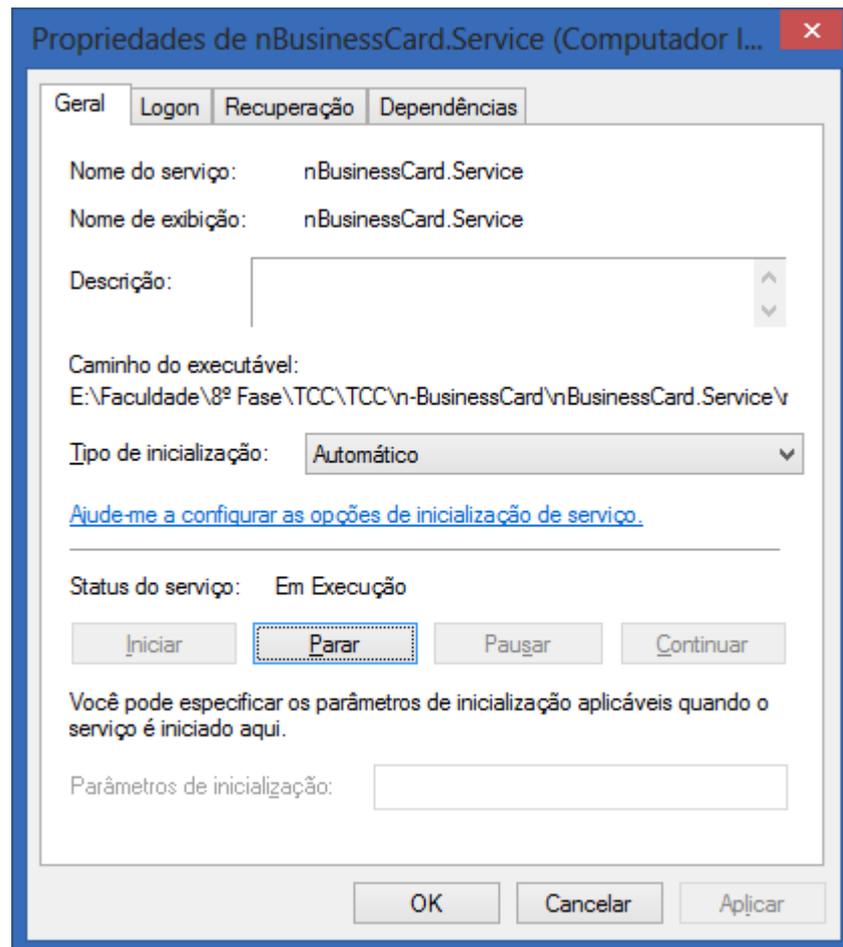


**Figura 14 – Execução do *script* no navegador**  
**Fonte: Próprio autor**

## 2.9 Serviço do Windows

Um serviço do Windows é uma aplicação que inicia quando o Windows é carregado e roda em segundo plano enquanto o Windows estiver em execução. Um aplicação do tipo serviço do Windows é útil quando deseja-se monitorar constantemente alguns arquivos, o teclado, criar arquivos de *log*, ou efetuar qualquer tarefa em segundo plano enquanto o SO está ocupando com o seu trabalho.

A figura 15 exhibe um serviço do Windows instalado que está sendo executado em segundo plano no SO:



**Figura 15 - Serviço do Windows sendo executado**  
**Fonte: Próprio autor**

### 3 PROJETO

Para que o desenvolvimento do sistema aqui descrito, como em qualquer outro sistema computacional, é necessário um estudo prévio para definir a abrangência do sistema, suas funcionalidades, requisitos e o *hardware* necessário para que o sistema promova os resultados esperados.

Segundo LARMAN (2000) o desenvolvimento de uma aplicação necessita de uma descrição do problema e dos seus requisitos. Para identificar o problema e o que o sistema precisa fazer, deve-se realizar uma análise, a qual enfatiza uma investigação do problema e de como a solução será definida. Porém, é necessário ter descrições detalhadas da solução

lógica e de como ela atende aos requisitos e restrições existentes, para tanto se realiza o projeto.

Ao modelar e projetar o Business Card, o principal objetivo foi facilitar a vida dos colaboradores da NDDigital Software S/A, evitando assim que os colaboradores, por exemplo, percam muito tempo na localização de outros colaboradores, para transferir uma ligação interna ou externa, ou apenas para conhecer melhor, os colaboradores, e também eliminar o retrabalho necessário, a cada vez que ocorre o envio dos e-mails ao colaboradores, que eram gerados manualmente.

Fez-se necessário a escolha de uma plataforma de desenvolvimento *Web*, que se baseie em uma linguagem de programação orientada a objetos, foi assim escolhido o ASP.Net, por ser integrada ao *framework* Microsoft .NET, e pelo fato de se poder utilizar a linguagem POO compilada, o C# (*C Sharp*), e ser uma plataforma robusta e gratuita, que possibilita desde o desenvolvimento de aplicações *web*, até aplicativos de serviço (aplicações que são executadas no SO quando o sistema é inicializado, executando assim, determinadas tarefas em um intervalo de tempo, ou quando um evento é disparado, por exemplo).

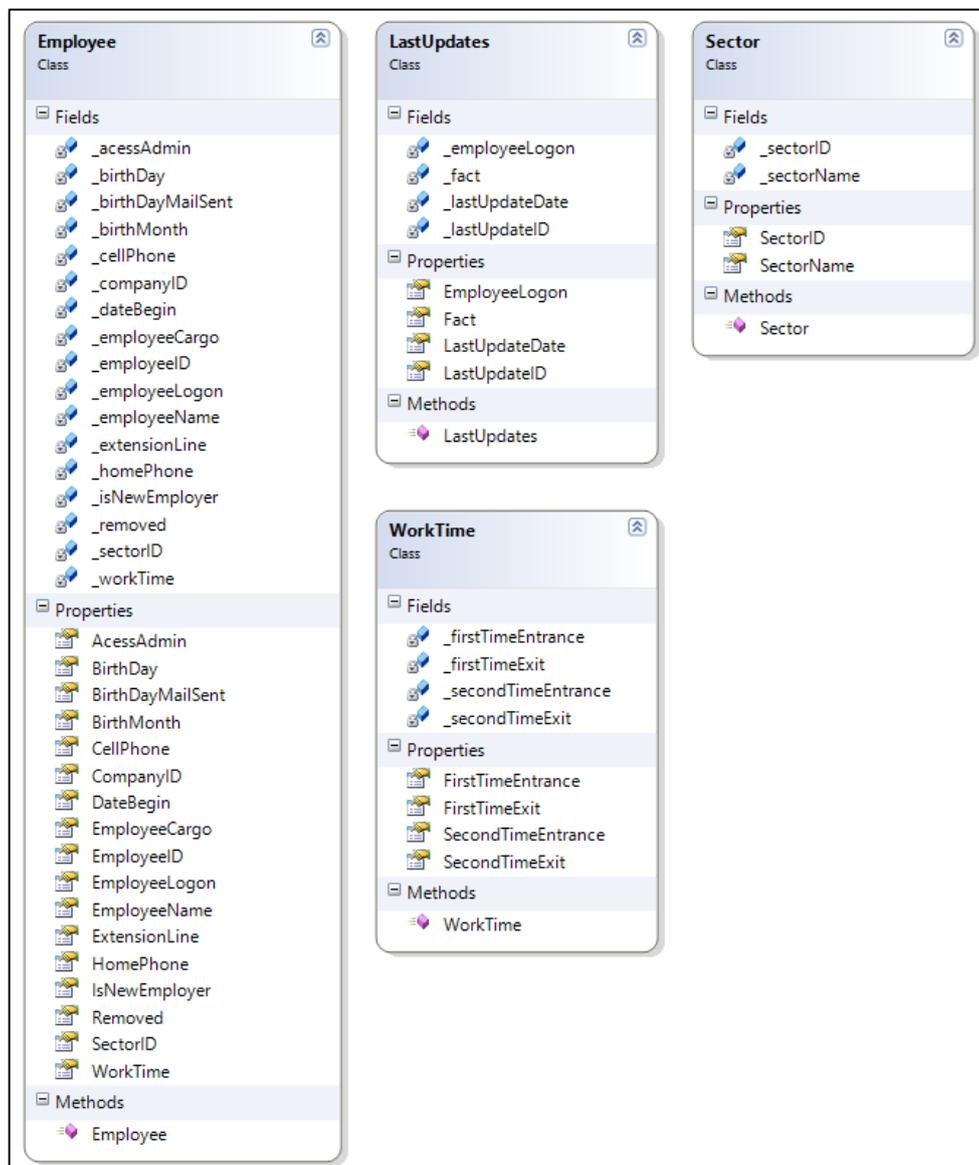
A funcionalidade do sistema consiste de que, o colaborador da empresa NDDigital Software S/A, através do sistema, poderá tanto acessar os dados de contato de todos os colaboradores, tão como manter o seus dados atualizados, sendo estas as principais funcionalidades:

- Cadastro das informações dos colaboradores da empresa NDDigital Software S/A, tanto novos quanto os que já estão empregado;
- Consulta dos dados dos colaboradores, tanto dados de contatos quanto fotos dos colaboradores;
- Somente os Administradores podem cadastrar novos contatos ou remover colaboradores do sistema;
- O sistema deverá ter uma sistema com envio de e-mails automáticos, que enviará e-mails a todos os colaboradores ou apenas específicos;

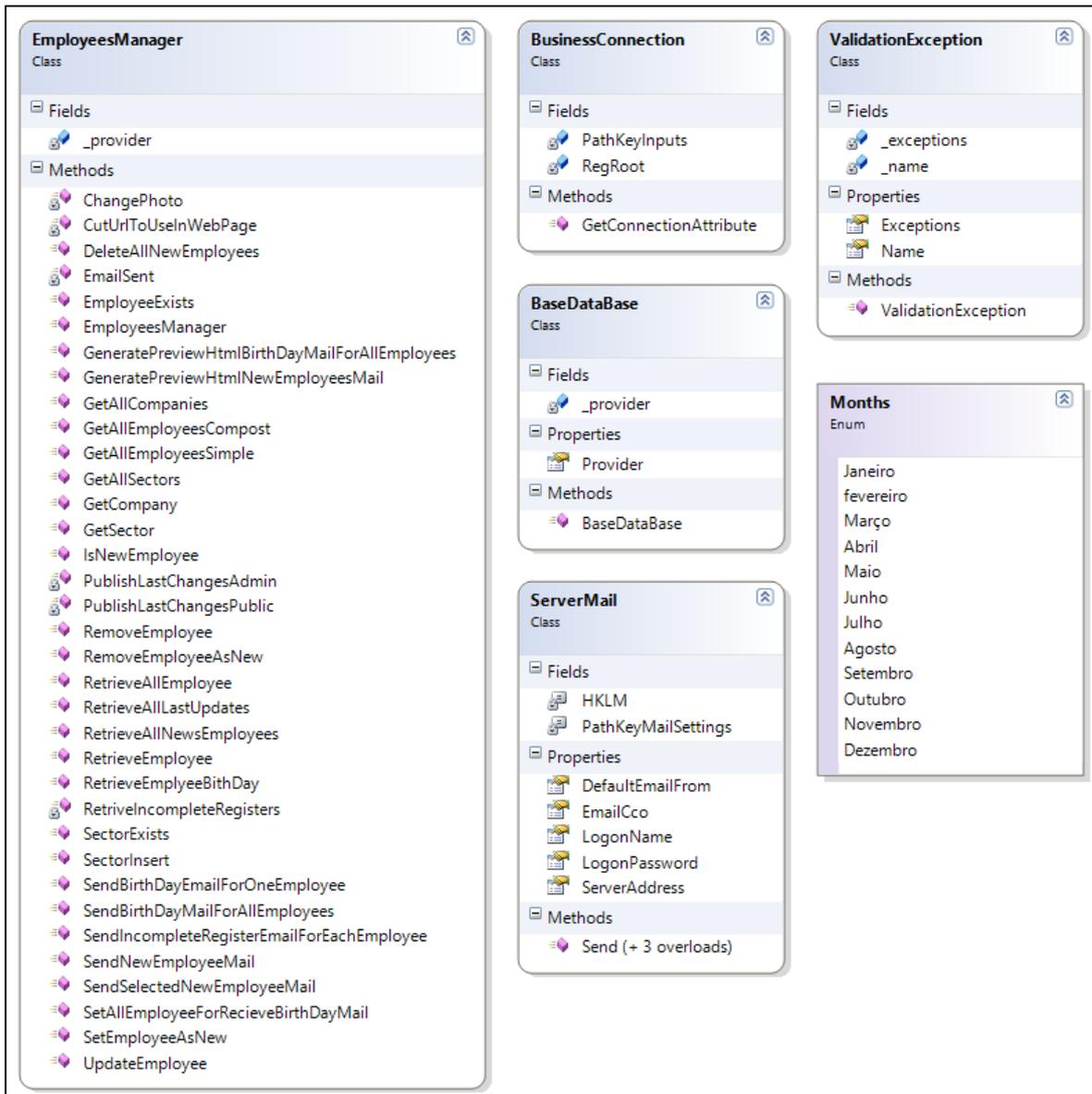
### 3.1 Diagrama de Classes

Segundo FURLAN (1998) trata-se de uma estrutura lógica estática em uma superfície de duas dimensões mostrando uma coleção de elementos declarativos de modelo como classes, tipos e seus respectivos conteúdos e relações.

O diagrama de classe contém a representação das classes necessárias para todo o funcionamento correto do sistema. Aqui estão todas as classes de regra de negócios, de acesso a camada de dados e as classes complexas que reapresentam as abstrações do mundo real.



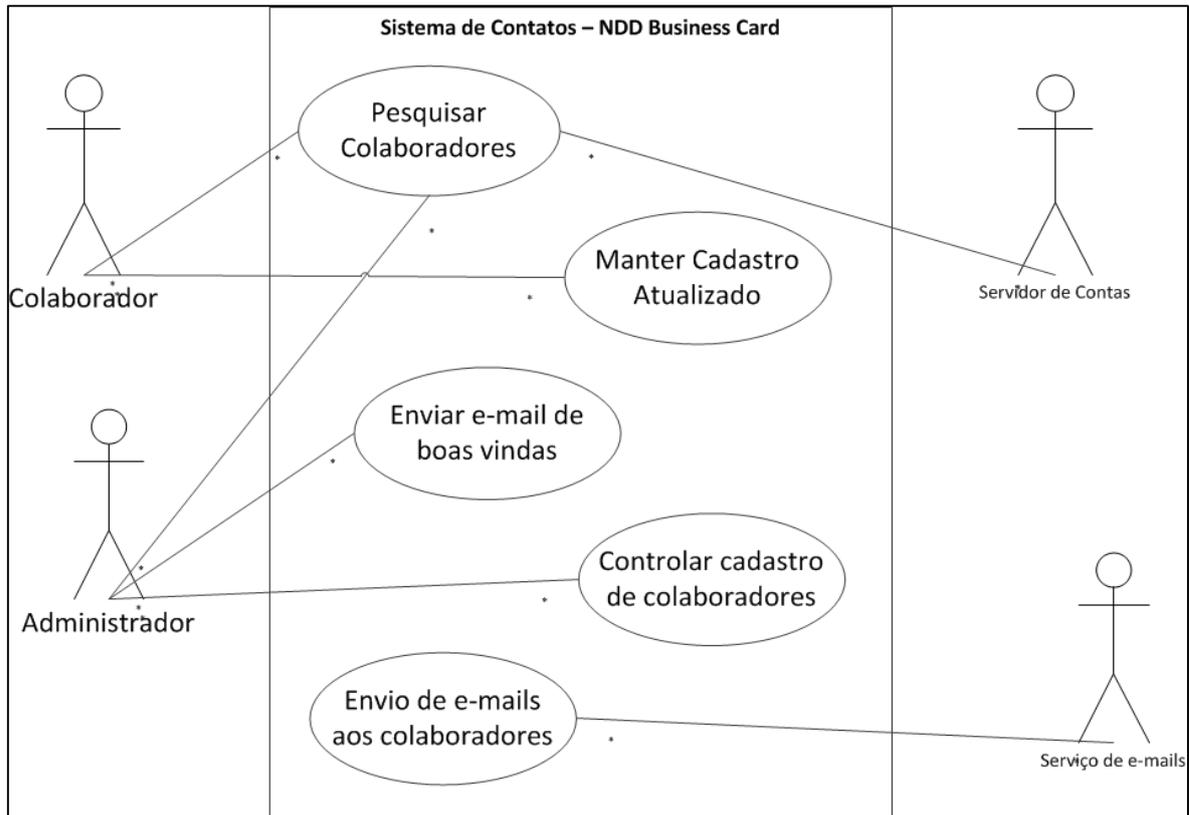
**Figura 16 – Diagrama de Classe (1)**  
**Fonte: Próprio autor**



**Figura 17 – Diagrama de Classe (2)**  
**Fonte: Próprio autor**

### 3.2 Diagrama de Caso de Uso

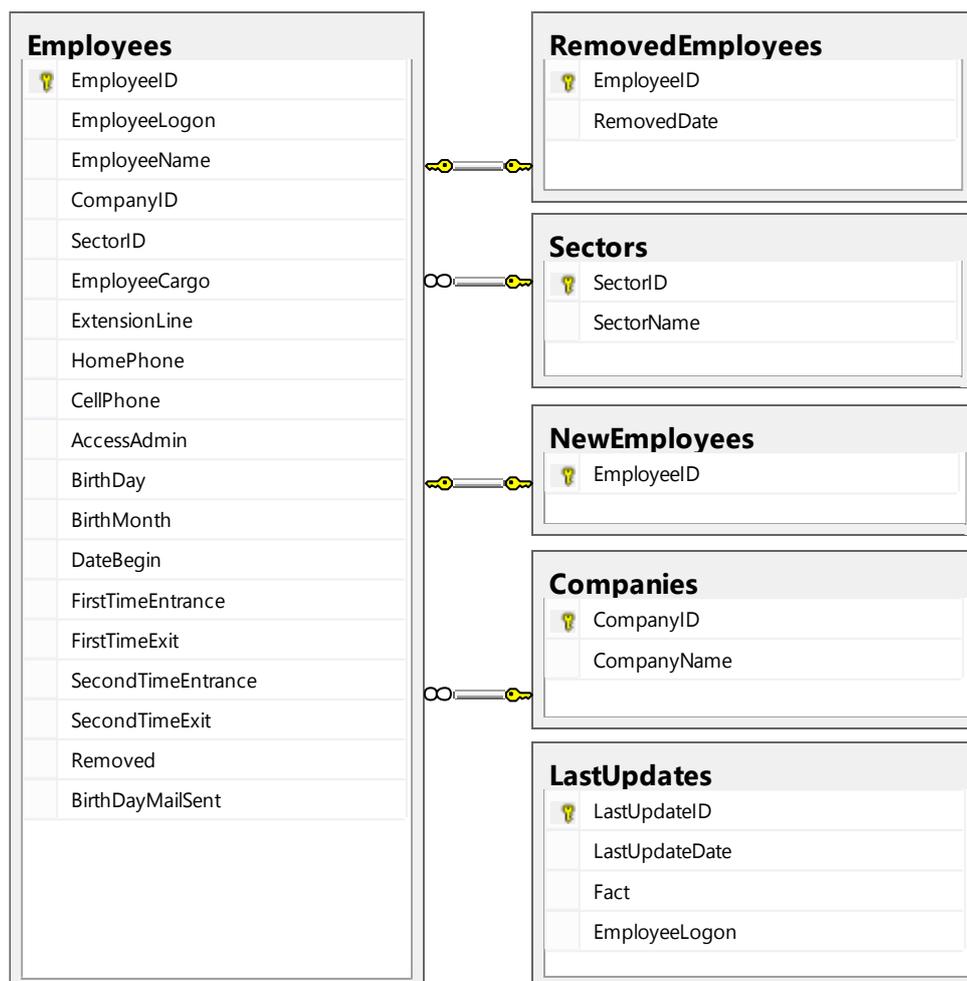
O diagrama de caso de uso informa de maneira mais clara para o usuário a abrangência e funcionalidades do sistema.



**Figura 18 – Caso de Uso**  
**Fonte: Próprio autor**

### 3.3 Diagrama de Entidade Relacionamento

O Diagrama de Entidade Relacionamento é um modelo que descreve os dados do sistema com alto nível de abstração, e tem como objetivo principal mostrar o relacionamento entre as tabelas do sistema.



**Figura 19 – Diagrama de Entidade Relacionamento**  
**Fonte: Próprio autor**

### **3.4 Interfaces do sistema Business Card**

Aqui serão apresentadas as interfaces gráficas do sistema, sendo que estas serão precedidas pelo termo “Visão” seguido do nome da mesma.

#### **3.4.1 Visão Funcionários**

Contém as 20 últimas atualizações feitas nas informações dos colaboradores, e as informações reduzidas dos funcionários, para o rápida pesquisa e localização da informação, e o acesso para que o colaborador possa atualizar as suas informações.

**NDD** NDDigital Business Card (beta)

Fotos Editar Minhas Informações

Pesquisa:

Nome	Setor	Ramal
Fulana de Tal	Direção	8000
Maicon Ciclano	Desenvolvimento	8000
Marco Aurélio Cevey	Desenvolvimento	8000
Maria Teste da Silvas	Suporte	8000

**Últimas Atualizações**

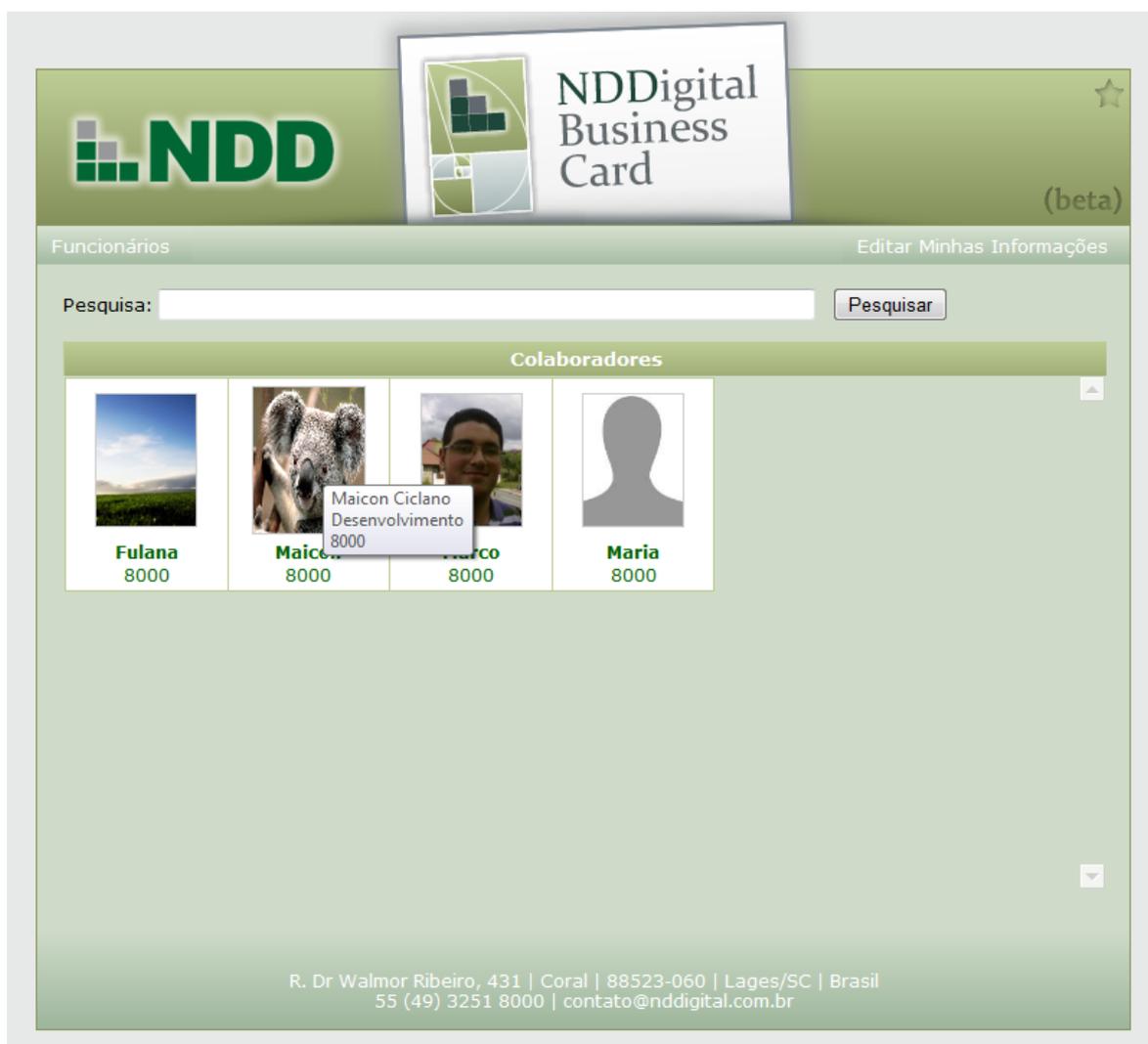
- Novo funcionário cadastrado: **Maria Teste da Silvas.**
- Maicon Pereira** teve seu nome alterado para **Maicon Ciclano.**
- Novo funcionário cadastrado: **Maicon Pereira.**
- Novo funcionário cadastrado: **Fulana de Tal.**
- Marco Aurélio Cevey** teve sua empresa alterada para **NDDigital.**
- Novo funcionário: **Marco Aurélio Cevey.**

R. Dr Walmor Ribeiro, 431 | Coral | 88523-060 | Lages/SC | Brasil  
55 (49) 3251 8000 | contato@nndigital.com.br

**Figura 20 – Visão Funcionários**  
**Fonte: Próprio autor**

### 3.4.2 Visão Fotos

Possui as fotos dos colaboradores, que mostra as fotos pra identificação dos colaboradores pela face, como pesquisa e acesso ao seus cadastros.



**Figura 21 – Visão Fotos**  
**Fonte: Próprio autor**

### 3.4.3 Visão Cartão de Negócio

Contém todas as informações do Colaborador, esta visão é acessada clicando no nome, do colaborador, foto ou atualização do mesmo.



The screenshot shows a web interface for NDDigital Business Cards. At the top, there is a logo for NDD and the text "NDDigital Business Card" with a "(beta)" label. Below this, there are navigation options like "Fotos" and "Editar Minhas Informações". A search bar is visible with the text "Pesquisa: m" and "Resulta". The main content area displays a business card for Marco Aurélio Cevey. The card includes a photo of the employee, his name, email address ([marco.cevey@nddigital.com.br](mailto:marco.cevey@nddigital.com.br)), and his sector: "Desenvolvimento". Contact information includes "Ramal: 8000", "Residencial: (49) 3222-2754", and "Celular: (49) 9957-3676". His work hours are listed as "Horário de trabalho: 08:00 às 12:00 e 13:30 às 18:18". The card also features the NDD logo and the text "Admissão: Junho/2011" and "Aniversário: 16 de Abril". At the bottom of the page, there is contact information for the company: "R. Dr Walmor Ribeiro, 431 | Coral | 88523-060 | Lages/SC | Brasil" and "55 (49) 3251 8000 | contato@nddigital.com.br".

**Figura 22 – Visão Cartão de Negócios**  
**Fonte: Próprio Autor**

### 3.4.4 Visão Pesquisa

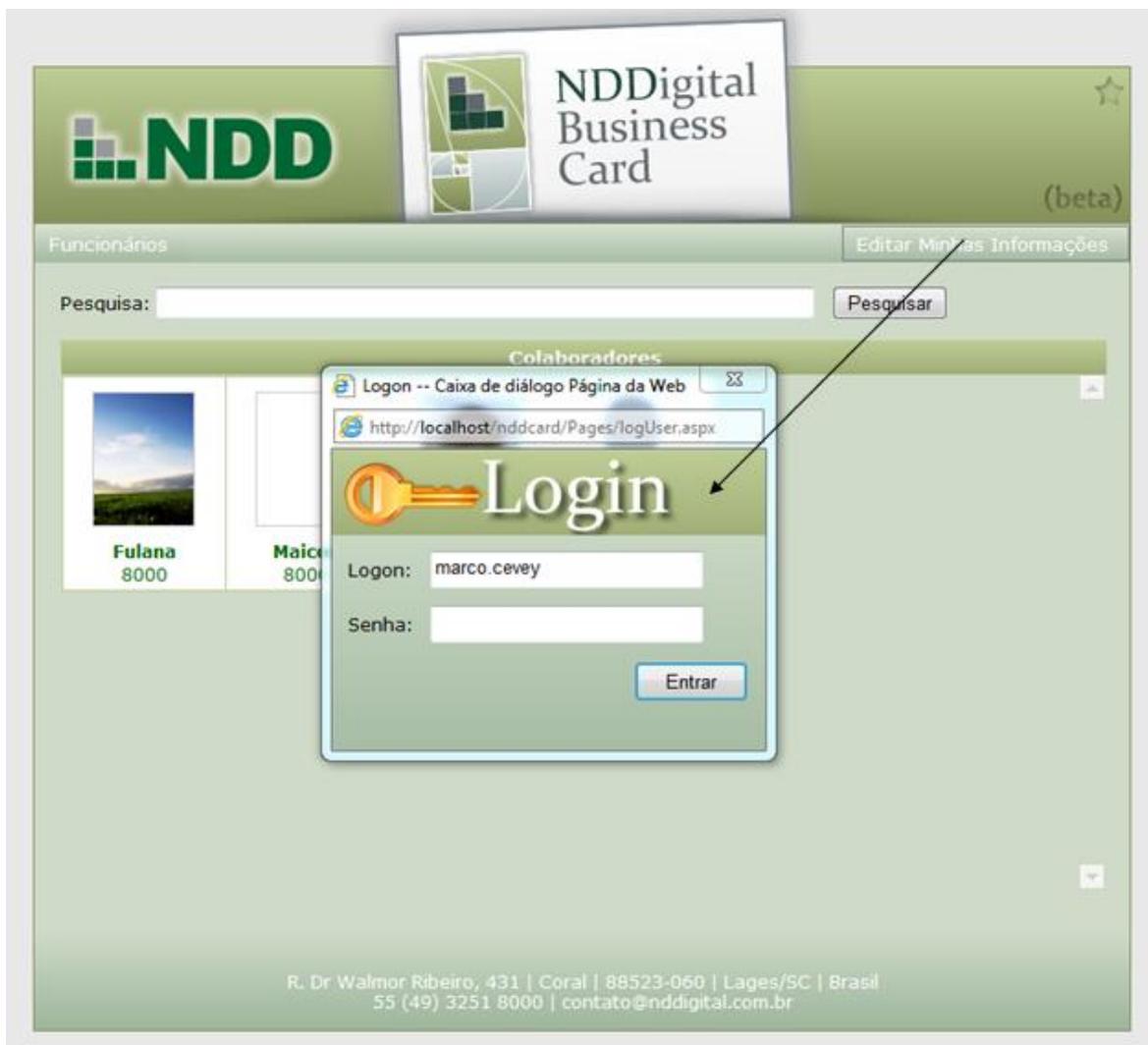
O usuário pode executar uma pesquisa, com o nome, ramal ou setor do colaborador, para facilitar o acesso às informações.



**Figura 23 – Visão Pesquisa**  
**Fonte: Próprio Autor**

### 3.4.5 Visão Logon

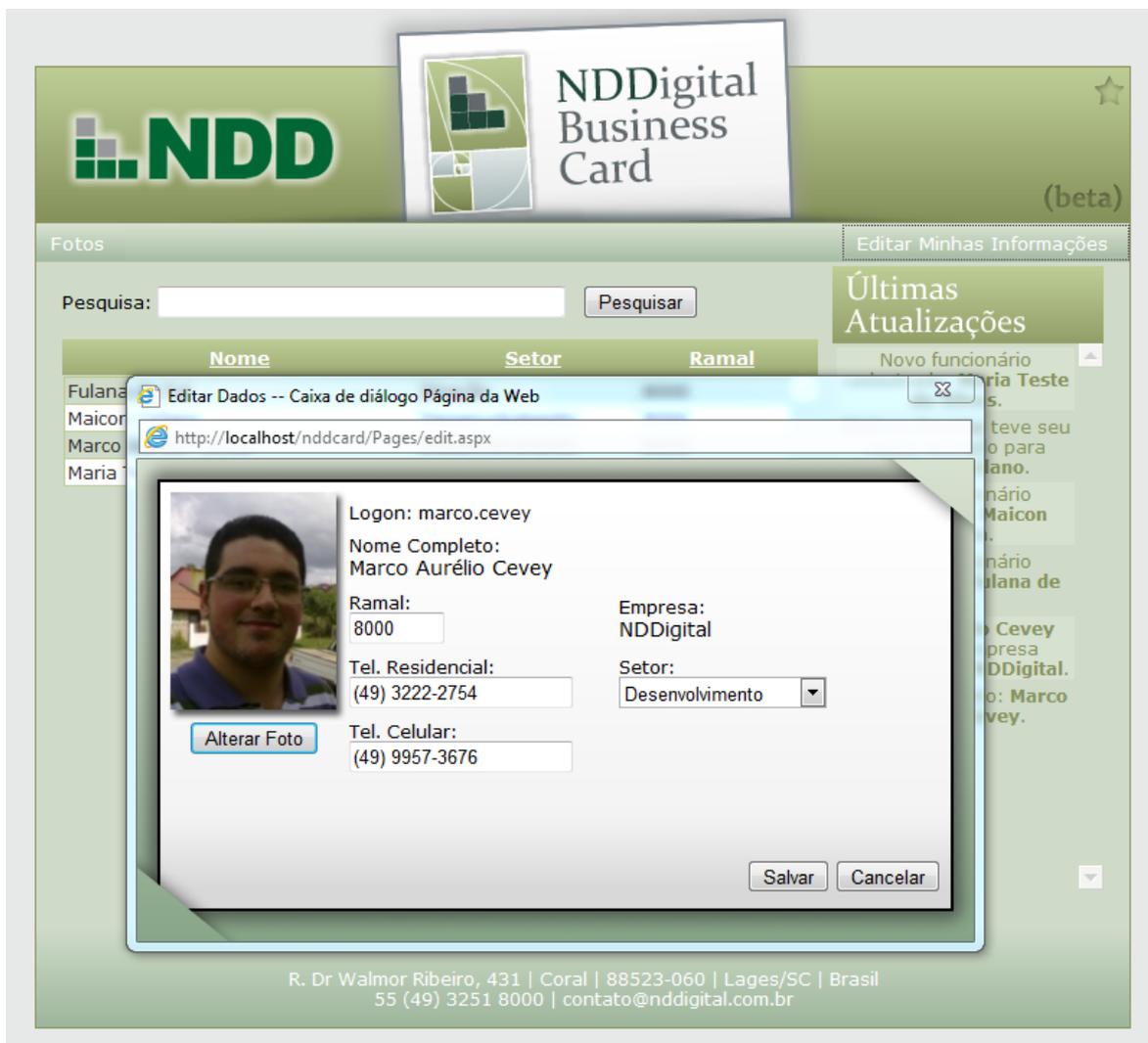
O usuário pode acessar e alterar os seus dados.



**Figura 24 – Visão Logon**  
**Fonte: Próprio autor**

### 3.4.6 Visão Editar Dados

O usuário tem acesso aos seus dados para atualização.

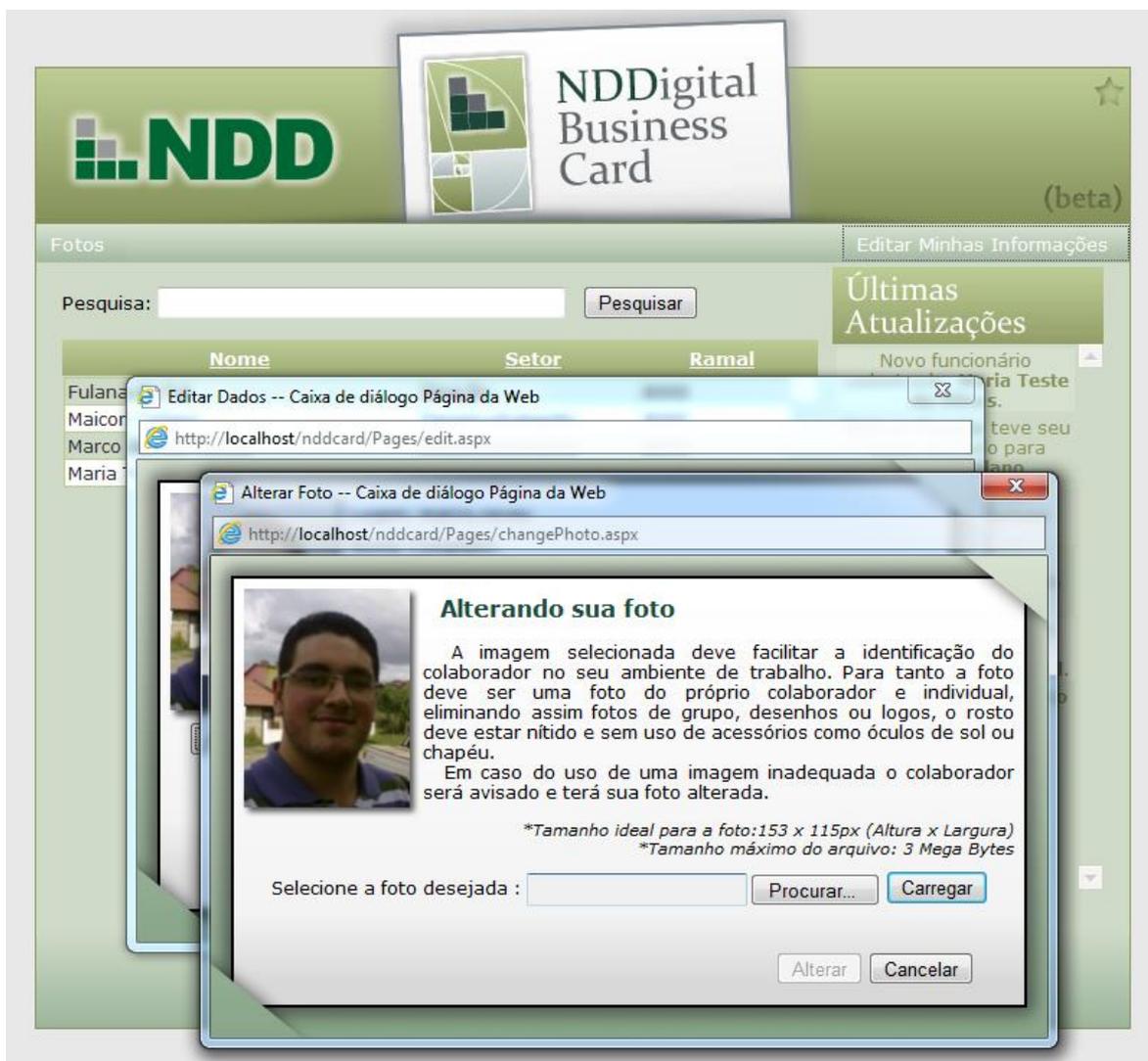


**Figura 25 - Visão Editar Dados**

Fonte: Próprio autor

### 3.4.7 Visão Alterar Foto

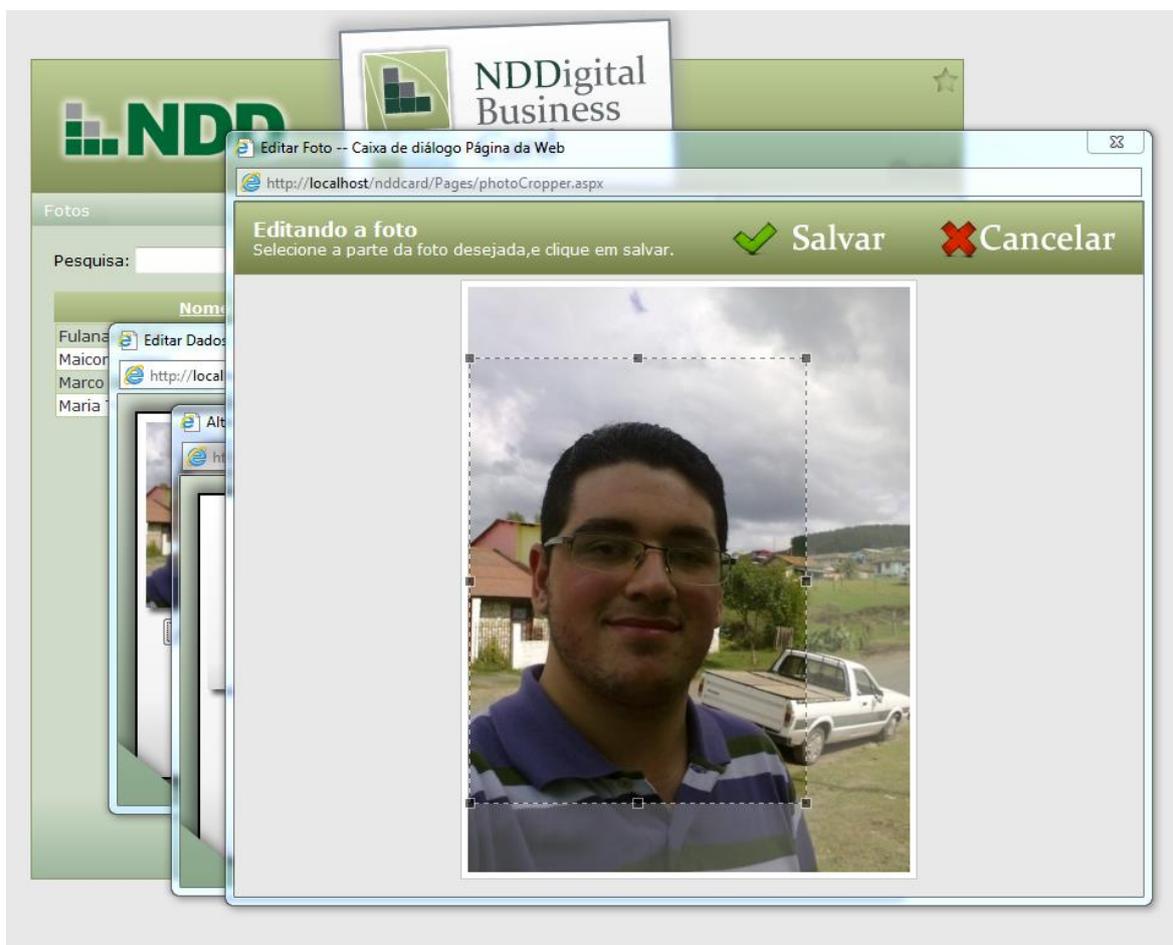
O usuário pode selecionar uma foto desejada, seguindo normas de conduta da empresa NDDigital Software S/A.



**Figura 26 – Visão Alterar Fotos**  
Fonte: Próprio autor

### 3.4.8 Visão Editar Foto

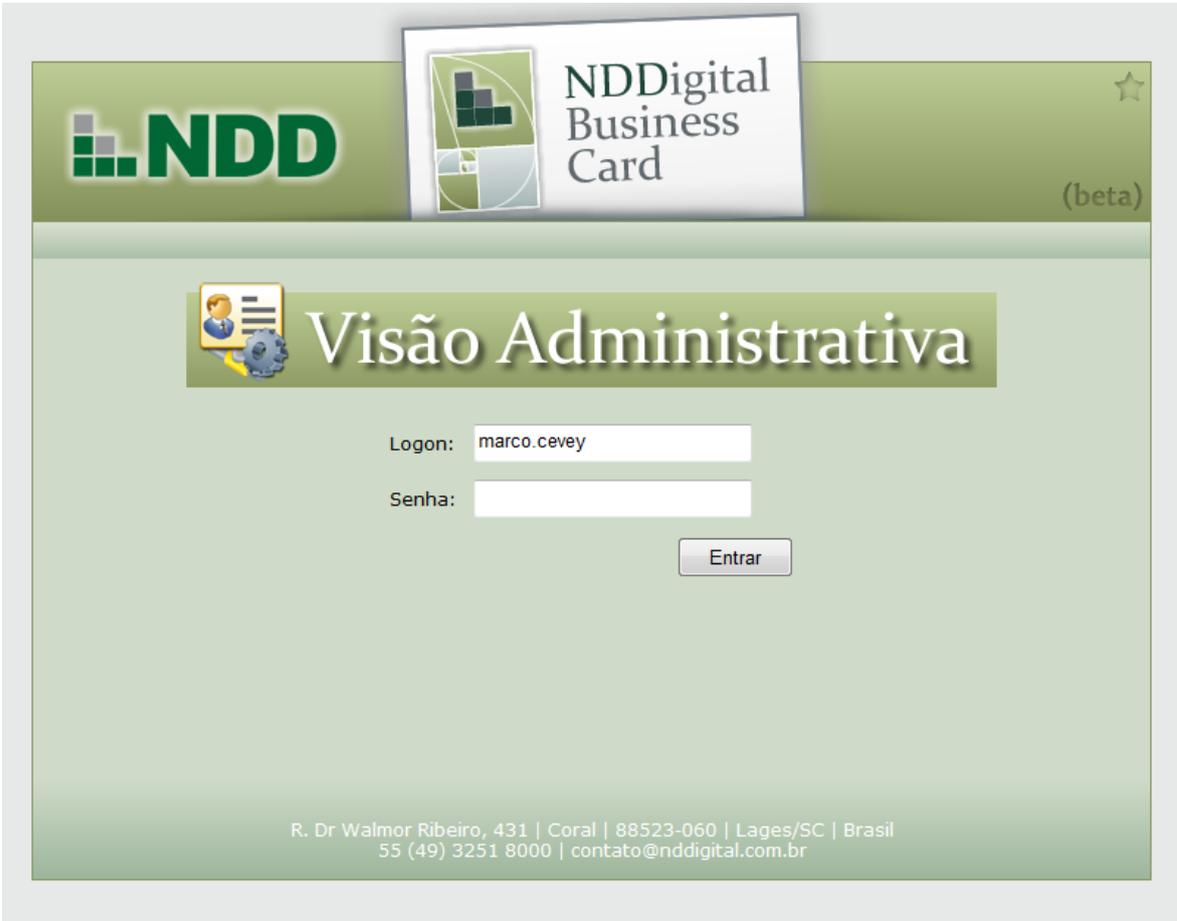
Quando o usuário clicar no botão “Carregar” na visão Editar Foto, caso a foto selecionada possua um tamanho maior que 153 *pixels* de altura por 115 *pixels* de largura, esta visão será aberta para que o usuário possa selecionar a parte desejada da foto, desta maneira, todas as fotos armazenadas no sistema possuem a mesma relação de altura e largura, para que as fotos não fiquem distorcidas, tal qual, reduzindo o espaço de armazenamento de cada foto no servidor.



**Figura 27 – Visão Editar Foto**  
**Fonte: Próprio autor**

### 3.4.9 Visão Logon Administrativo

Nesta visão, somente os usuários administradores possuem acesso, para gerenciarem o sistema.



The image shows a web browser window displaying the administrative login page for NDD Digital Business Card. The page has a green and white color scheme. At the top left is the NDD logo. To its right is a box containing the text "NDDigital Business Card" and a small graphic. In the top right corner, there is a star icon and the text "(beta)". The main heading "Visão Administrativa" is centered, accompanied by an icon of a person and a gear. Below the heading are two input fields: "Logon:" with the text "marco.cevey" and "Senha:". A button labeled "Entrar" is positioned below the password field. At the bottom of the page, contact information is provided: "R. Dr Walmor Ribeiro, 431 | Coral | 88523-060 | Lages/SC | Brasil" and "55 (49) 3251 8000 | contato@nndigital.com.br".

**Figura 28 – Visão Logon Administrativo**  
**Fonte: Próprio autor**

### 3.4.10 Visão Opções do Administrador

Possui as opções de pesquisa, para fácil acesso ao cadastro dos colaboradores, onde pode: cadastrar, alterar e excluir o cadastro de um colaborador, como enviar um e-mail de boas-vindas e visualizar e enviar os *e-mails* de Boas Vindas ao novos colaboradores e Aniversariantes do mês.

U usuário(a): Marco Aurélio Cevey

Pesquisa:

	Nome	Setor	Ramal
✓	Antonio Silva	Financeiro	8500
✓	Fulano Marcon	Pré-Vendas	8402
✓	Jeane Cevey	Direção	8045
✓	Jéssica Cevey	Marketing	8012
✓	Marco Aurélio Cevey	Desenvolvimento	8049
✓	Valdir de Souza Ribeiro	Mercado	8000
✓	Zélia Marcon	Universidade	8000

**Opções**

Selecione um(a) colaborador(a) ao lado:

- Novo
- Editar
- Boas-vindas ao colaborador
- Remover
- Boas Vindas
- Aniversariantes
- Sair

R. Dr Walmor Ribeiro, 431 | Coral | 88523-060 | Lages/SC | Brasil  
55 (49) 3251 8000 | contato@nndigital.com.br

**Figura 29 – Visão Opções do Administrador**  
Fonte: Próprio autor

### 3.4.11 Visão Novo Cadastro

Nesta visão, o administrador poderá cadastrar um novo colaborador.

The screenshot shows a web browser window with the title "Novo Cadastro -- Caixa de diálogo Página da Web" and the URL "http://localhost/nddcard/pages/admNew.aspx". The browser window is overlaid on a web application interface for "NDDigital Business Card (beta)". The application interface includes a logo for "NDD" and a navigation menu with items like "Usuário", "Pesquis", "Fu", "Ma", "Ma", and "Ma". The "Novo Cadastro" form is a modal dialog box with the following fields and values:

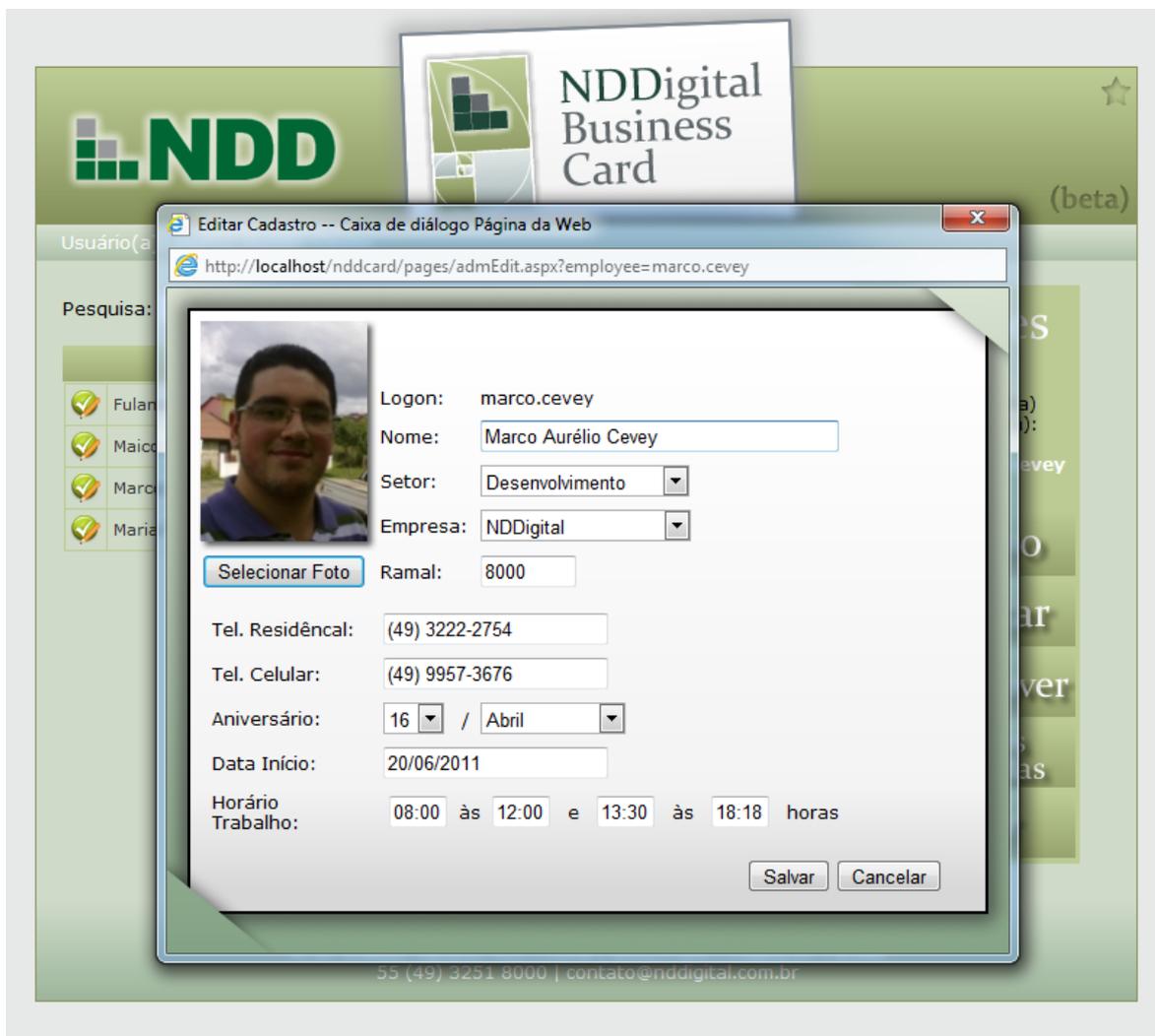
- Logon:
- Nome:
- Setor:
- Empresa:
- Ramal:
- Tel. Residencial:
- Tel. Celular:
- Aniversário:  /
- Data Início:
- Horário Trabalho:

At the bottom right of the form, there are two buttons: "Salvar" and "Cancelar".

**Figura 30 – Visão Novo Colaborador**  
Fonte: Próprio autor

### 3.4.12 Visão Editar Cadastro

Nesta visão, o administrador poderá editar o cadastro de um colaborador.



**Figura 31 – Visão Editar Cadastro**  
Fonte: Próprio autor

### 3.4.13 Visão Remover Cadastro

Nesta visão, o administrador pode remover o cadastro de um colaborador.



**Figura 32 – Visão Remover Usuário**  
Fonte: Próprio autor

### 3.4.14 Visão Novos Colaboradores

Nesta visão, o administrador poderá gerar, visualizar e enviar o *e-mail* com os novos colaboradores, tal como alterar os novos colaboradores para não serem considerados mais como novos colaboradores no sistema.



**Figura 33 – Visão Novos Colaboradores**  
**Fonte: Próprio autor**

### 3.4.15 Visão Aniversariantes

Nesta visão, o administrador poderá gerar, visualizar e enviar o *e-mail* com os aniversariantes do mês.



**Figura 34 – Visão Aniversariantes**  
**Fonte: Próprio autor**

### 3.5 E-mails enviados pelo sistema

Aqui será mostrado os *e-mails* que são gerados pelo sistema, sendo que estas mensagens eletrônicas podem ser geradas e enviadas pelos administradores do sistema, ou pelo serviço de envios automáticos que é executado todo dia as 00:00 horas, gerando e enviando os *e-mails* específicos para colaborador.

#### 3.5.1 E-mail de Boas-vindas ao colaborador

Esta mensagem é enviada ao colaborador selecionado na Visão de Opções do Administrador dando as boas-vindas aos mesmos.

##### **Marco**

Desejamos a você as boas vindas a família NDDigital e sucesso neste seu novo desafio. Acreditamos que você desenvolverá a função que lhe foi confiada com excelência e crescerá nesta organização, realizando-se pessoal e profissionalmente.

Conte sempre com nosso apoio.

Recursos Humanos  
NDDigital

**Figura 35 – E-mail de boas-vindas ao colaborador**  
**Fonte: Próprio autor**

### 3.5.2 E-mail de Novos Colaboradores

Esta mensagem é gerada e enviada pelos administradores do sistema para o setor de Recursos Humanos que reenvia o *e-mail* para todos os colaboradores.



Figura 36 – E-mail de novos colaboradores  
Fonte: Próprio autor

### 3.5.3 E-mail dos Aniversariantes do Mês

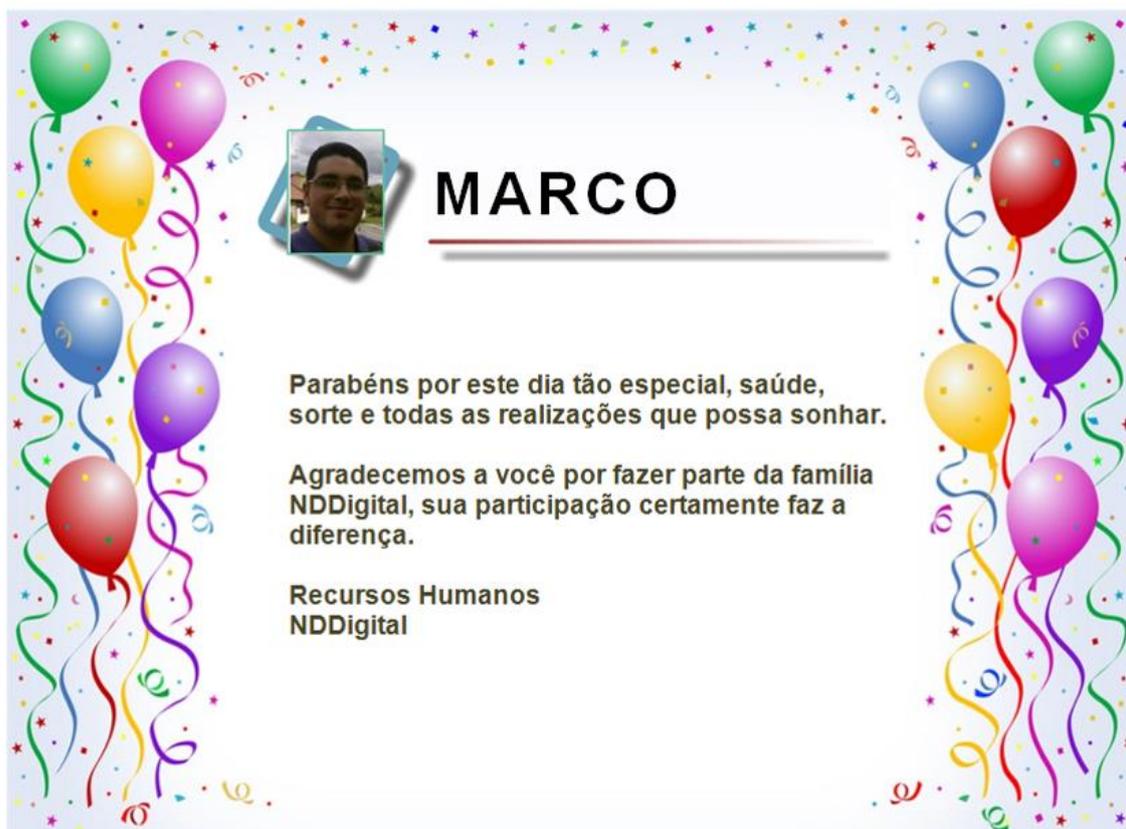
Esta mensagem é gerada e enviada pelos administradores do sistema para o setor de Recursos Humanos que reenvia o *e-mail* para todos os colaboradores.



Figura 37 – E-mail de aniversariantes do mês  
Fonte: Próprio autor

### 3.5.4 E-mail de Feliz Aniversário ao Colaborador

Este *e-mail* é enviado pelo serviço de envio do sistema para cada colaborador no seu dia de nascimento.



**Figura 38 – E-mail de feliz aniversário ao colaborador**  
Fonte: Próprio autor

### 3.5.5 E-mail de Cadastro Incompleto

Esta *e-mail* é enviado pelo serviço automático de envio para cada colaborador que não está com suas informações completas, ou seja, não possua foto, ramal, telefone fixo ou celular, nas exceções, o serviço lê um arquivo XML (Anexo A) com as exceções, e ignora este colaboradores, que por algum motivo, pode ter suas informações incompletas.

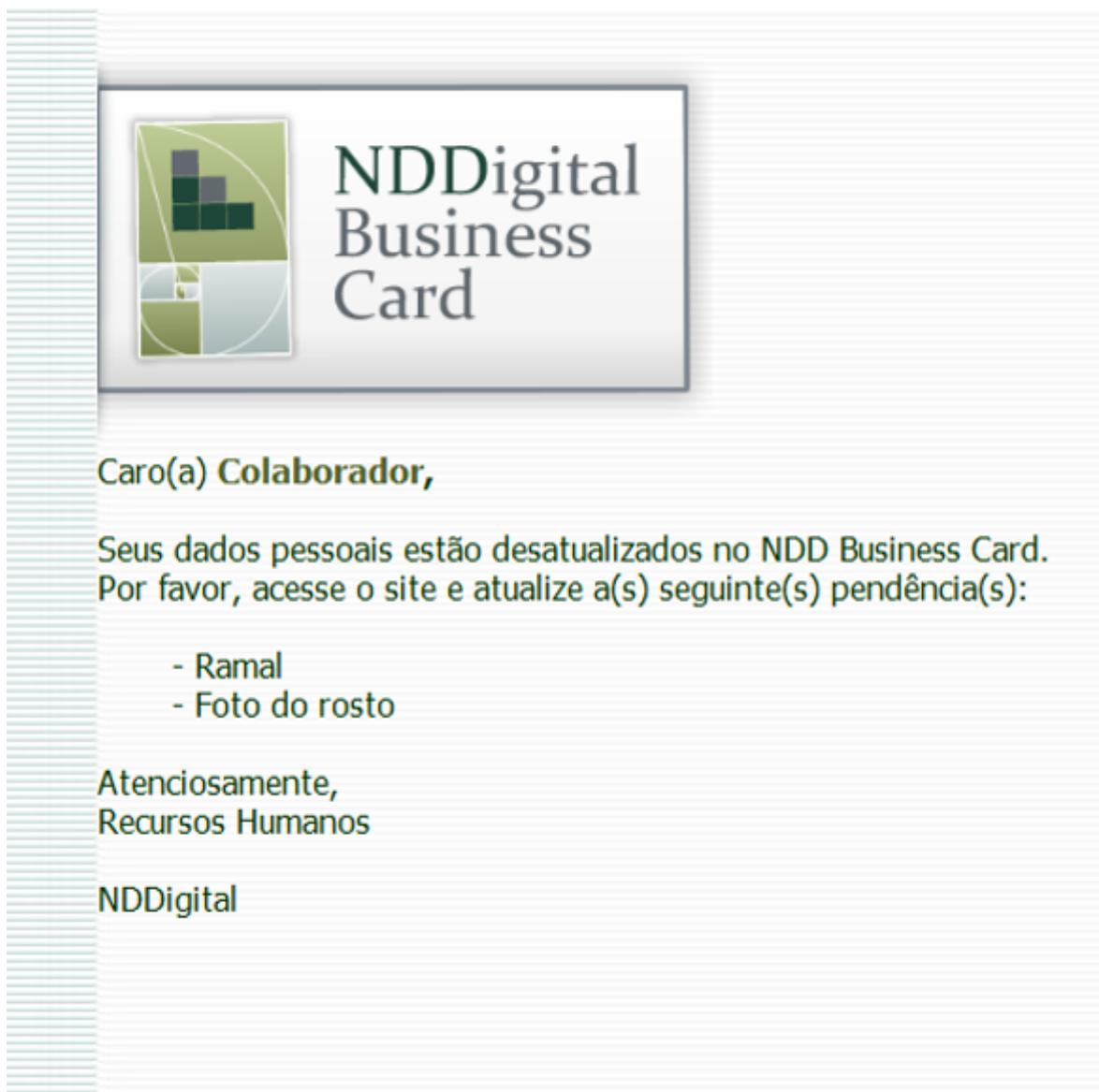


Figura 39 – E-mail de cadastro incompleto  
Fonte: Próprio autor

## 4 CONSIDERAÇÕES FINAIS

Os sistemas *web* mudam totalmente o paradigma de desenvolvimento de programas, de modo que o padrão de desenvolvimento baseado em eventos, como o que ocorre nas aplicações *desktop*, não se aplica de forma natural a arquitetura HTTP, que é uma arquitetura sem eventos, mas uma das grandes facilidades da plataforma ASP.NET, é que a mesma nos proporciona através de algumas modificações na arquitetura HTTP, a possibilidade de termos nas páginas a programação orientada a eventos, o que facilita muito o desenvolvimento de que está iniciando no desenvolvimento de sistemas *Web*.

Mas o que se pode notar, após o desenvolvimento deste sistema que aqui foi apresentado, é que o desenvolvimento *Web*, apesar de seus problemas, facilita muito a rápida implantação e disponibilização das aplicações baseadas no sistema *web*, da mesma maneira, que a manutenção torna-se muito mais fácil, pelo fato de que o sistema está apenas implantado em um local, uma correção de problemas fica mais acessível, e está correção logo está acessível a todos os usuários do sistema.

Atualmente, as aplicações *web*, estão ficando cada vez mais complexas, mas com o surgimento do HTML 5, por exemplo, e a padronização do modo como os navegadores exibem as páginas HTML, as aplicações *web* estão sendo cada vez mais amigáveis, ágeis e fáceis de utilizar, claro que isto será possível apenas se houver uma boa arquitetura envolvida por trás do projeto em questão.

E o sistema que foi implantado na empresa NDDigital Software S/A, está sendo utilizado em sua totalidade, facilitando o contato entre os colaboradores da empresa, sempre que necessário, e também mantém os colaboradores informados sobre os seus novos companheiros de trabalho ou até mesmo os aniversariantes da sua equipe, o que é de grande valia no âmbito de trabalho, pois pessoas que sabem realmente trabalhar em equipe, sabem quanto valioso o entrosamento é necessário.

## 5 REFERÊNCIAS BIBLIOGRÁFICAS

MANZANO, José Augusto N. G., Microsoft SQL Server 2008 Express - Interativo - Guia Prático. **Erica-2009.**

Sihler, Andrew L.. **New Comparative Grammar of Greek and Latin.** illustrated ed. New York: Oxford University Press, 1995. 21 p.

Navathe, S. B. and Elmasri, R.. **Sistemas de banco de dados – Fundamentos e aplicações.** LTC: 2002.

CAMACHO Jr., Carlos Olavo de Azevedo, **Desenvolvimento em Camadas com C####.**  
NET, Visual Books – 2008.

MEDEIROS, Ernani, **Desenvolvendo software com UML 2.0: definitivo /** Ernani Sales de Medeiros. – São Paulo: Pearson Makron Books, 2004.

ACADEMY, Quint W. Redwood. **Conceitos Básicos ITIL para Gerenciamento de Serviços em TI.** São Paulo: Quint, 2003.

CONALLEN, Jim. **Desenvolvendo aplicações web.** Rio de Janeiro: Campus, 2003.

DALFOVO, Oscar; AMORIM, Sammy Newton. **Quem tem informação é mais competitivo.** Blumenau: Acadêmicas Publicações Ltda., 2000.

FEATHER, Stephen. **Java Script em exemplos.** São Paulo: Makron Books, 1997.

FREITAS, Henrique; LESCA, Humberto. **Competitividade empresarial na era da informação**. Revista de Administração, São Paulo, 1992.

HILES, A. N. Service level agreements: panacea or pain? The TQM Magazine, Bradford, v. 6, n. 2, p. 14, 1994.

GAMMA, Erich *et al.* Padrões de Projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000.

MAGALÃES, Ivan Luizio; PINHEIRO, Walfrido Brito. **Gerenciamento de Serviços de TI na Prática: Uma abordagem com base na ITIL**. São Paulo: Novatec, 2007.

OLIVEIRA, Djalma de Pinho Rebouças. **Sistemas de informações gerenciais**. São Paulo: Atlas, 1992.

PEREIRA, Felipe Luiz. CPITIL: **Uma aplicação de apoio ao gerenciamento de problemas baseado na recomendação ITIL**. Itajaí, 2007. [156f]. Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) - Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, Itajaí, 2007.

REZENDE, Denis Alcides – Planejamento de sistemas de informação e informática: guia prático para planejar a tecnologia da informação integrada ao planejamento estratégico das organizações. Edição. São Paulo: Atlas, 2003.

RODRIGUES, Leonel César. Impactos dos sistemas de informação, Jornal de Santa Catarina, Blumenau-SC. Caderno de Economia, p. 2, 30 jun. 1996.

STAIR, Ralph M. **Princípios de sistemas de informação: uma abordagem gerencial**. Rio de Janeiro: LTC, 1998.

STURM, R.; MORRIS, W. JANDER, M. **Service level management: fundamentos do gerenciamento dos níveis de serviço**. Rio de Janeiro: Campus, 2001.

MICROSOFT CORPORATION. **.NET Framework Overview**. 2008. Disponível em: <<http://www.microsoft.com/net/Overview.aspx>>. Acessado em: 14 nov 2013.

MARCELO, Jean. **Introdução ao C#**. 2011. Disponível em: <<http://profjeanmarcelo.files.wordpress.com/2011/04/introduc3a7c3a3o-ao-c.pdf>>. Acessado em: 20 nov 2013.

MICROSOFT CORPORATION. **ASP.Net**. 2013. Disponível em: <<http://www.asp.net/>>. Acessado em: 18 nov 2013.

HADDAD, Renato. **Por que adotar o Visual Studio 2010?** Disponível em: <<http://msdn.microsoft.com/pt-br/library/ff978714.aspx>>. Acessado em: 15 nov 2013.

MACORATTI, José Carlos. **VB 2005 - Criando um Serviço do Windows (Windows Service)**. Disponível em: <[http://www.macoratti.net/vbn5\\_csw.htm](http://www.macoratti.net/vbn5_csw.htm)>. Acessado em: 20 nov 2013.

PPLWARE. **ASP.NET – Introdução e conceitos**. 2010. Disponível em: <<http://pplware.sapo.pt/tutoriais/asp-net-%E2%80%93-introducao-e-conceitos/>>. Acessado em 22 nov 2013.

## 6 ANEXOS

### 6.1 Anexo A - ValidationExceptions.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<ValidationExceptions>
  <ValidationException>
    <Name>WrongExtensionLine</Name>
    <Exceptions>
      <EmployeeLogon>marco.cevey</EmployeeLogon>
      <EmployeeLogon>ana.paula</EmployeeLogon>
    </Exceptions>
  </ValidationException>
  <ValidationException>
    <Name>NoPhoneNumber</Name>
    <Exceptions>
      <EmployeeLogon>fulano.silva</EmployeeLogon>
      <EmployeeLogon>beltrano</EmployeeLogon>
    </Exceptions>
  </ValidationException>
  <ValidationException>
    <Name>NoFacePhoto</Name>
    <Exceptions>
      <EmployeeLogon>ciclano</EmployeeLogon>
    </Exceptions>
  </ValidationException>
</ValidationExceptions>
```

### 6.2 Anexo B – Código do Serviço do Windows de envio de e-mails automático

```
using System;
using System.Collections.Generic;
using System.IO;
using System.ServiceProcess;
using System.Threading;
using BusinessCardBusiness;
using NDDigital.Core.Win32;

namespace nBusinessCard.Service.Alerts
{
  public partial class Service : ServiceBase
  {
    #region Variáveis

    Timer _timer;
    private static RegistryRoot RegRoot = RegistryRoot.HKEY_LOCAL_MACHINE;
    private static string PathKeyInputs = @"Software\NDDigital\BusinessCard\";
```

```

private static EmployeesManager manager = new EmployeesManager();

#endregion Variáveis

#region Construtor

public Service()
{
    InitializeComponent();
}

#endregion Construtor

#region Métodos

private void WriteLog(string fact)
{
    string path = "";
    try
    {
        path = Registry.Read64(RegRoot, PathKeyInputs, "LogDirectory",
String.Empty).ToString();
    }
    catch
    {
        path = @"c:\nBusinessCardServiceLog.txt";
    }

    StreamWriter vWriter = new StreamWriter(path, true);
    vWriter.WriteLine(DateTime.Now.ToString() + " " + fact);
    vWriter.Flush();
    vWriter.Close();
}

public void StartDebug(string[] args)
{
    OnStart(args);
}

#endregion Métodos

#region Eventos

protected override void OnStart(string[] args)
{
    if (!System.Diagnostics.Debugger.IsAttached)
    {
        TimerCallback _timerCallback = new TimerCallback(OnElapsedTime);
        _timer = new Timer(_timerCallback, null, 5000, 14400000);
        WriteLog("Serviço Iniciado");
    }
    else
    {
        OnElapsedTime(null);
    }
}

private void OnElapsedTime(object status)
{
    if (!System.Diagnostics.Debugger.IsAttached)
        _timer.Change(Timeout.Infinite, Timeout.Infinite);
}

```

```

try
{
    DateTime birthDayDay = Convert.ToDateTime(Registry.Read64(RegRoot,
PathKeyInputs, "DayBirthDayLast", "2001-01-01").ToString());
    DateTime weekBirthDay = Convert.ToDateTime(Registry.Read64(RegRoot,
PathKeyInputs, "WeekBirthDay", "2001-01-01").ToString());

    DateTime CurrentYear = Convert.ToDateTime(Registry.Read64(RegRoot,
PathKeyInputs, "YearSetBirthDay", "2001-01-01").ToString());

    #region Send Incomplete Registers Emails

    if (birthDayDay.Date < DateTime.Now.Date)
    {
        List<string> SendEmployeesList =
manager.SendIncompleteRegisterEmailForEachEmployee();

        if (SendEmployeesList.Count > 0)
        {
            foreach (string EmployeeName in SendEmployeesList)
            {
                WriteLog("Email para correção de cadastro enviado para "
+ EmployeeName);
            }
        }
    }

    #endregion Send Incomplete Registers Emails

    #region birth day
    if (birthDayDay.Date < DateTime.Now.Date)
    {
        if (CurrentYear.Year < DateTime.Now.Year)
        {
            Registry.Write64(RegRoot, PathKeyInputs, "YearSetBirthDay",
DateTime.Now.Year + "-01-01");
            manager.SetAllEmployeeForRecieveBirthDayMail();
        }

        List<string> births = manager.SendBirthDayEmailForOneEmployee();

        if (births.Count > 0)
        {
            for (int i = 0; i < births.Count; i++)
            {
                WriteLog("Email de aniversário enviado para " +
births[i]);
            }
        }

        Registry.Write64(RegRoot, PathKeyInputs, "DayBirthDayLast",
DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"));
    }

    #endregion
}
catch (Exception error)
{
    WriteLog(error.Message);
}

```

```
        if (!System.Diagnostics.Debugger.IsAttached)
            _timer.Change(14400000, 14400000);
    }

    protected override void OnStop()
    {
        WriteLog("Serviço Parado");
    }

    #endregion Eventos
}
}
```