

CONTROL MEDICAL CLINICS: SISTEMA CLÍNICO

Diego Gabriel De Souza Rignel¹

diegorignel@hotmail.com

Gabriel Pupin Chenci²

gabriel_gabrielpc@hotmail.com

Carlos Alberto Lucas³

profcarloslucas@gmail.com

RESUMO

Objetiva-se neste artigo apresentar um Sistema Clínico intitulado "Control Medical Clinics", desenvolvido em um trabalho de graduação, bem como descrever históricos e conceitos dos recursos que foram utilizados na aplicação, tais como: o Sistema Operacional Windows, Sistema Gerenciador de Banco de Dados SQL Server, a linguagem de programação Orientada a Objetos C#, e o tipo de arquitetura utilizado na fase de desenvolvimento, em Camadas. Para relatar as funcionalidades do sistema, serão exibidas imagens retiradas da aplicação ao longo do estudo, as quais serão analisadas e aclaradas para que o leitor possa fazer um juízo de valor do sistema baseado nas ilustrações. É consciencioso que o sistema não faz todas as operações envolvidas em um processo clínico, no entanto, as que são explicitadas ilustrativamente são funcionais. Devido a isso, ao decorrer do trabalho é exposto um tópico no qual são relatadas as limitações do sistema. Esta maneira de apresentação será feita para que o leitor saiba do que ele é, e não é capaz de operar junto à aplicação. Para descrever os recursos utilizados ao longo do desenvolvimento, foi feita uma revisão bibliográfica, por meio da qual se abordam informações relativas a históricos e conceitos de tais recursos. Por ser uma revisão bibliográfica, o estudo não contém resultados expressos em forma de números, sendo assim, pode-se considerar como resultado a importância desse sistema para o panorama universo da saúde, e a maneira com que cada recurso utilizado contribuiu na fase de desenvolvimento da aplicação. Mantém-se em aberto a possibilidade de um novo estudo que relate eventualmente novas funcionalidades e recursos do sistema.

Palavras-chave: Sistemas de saúde, desenvolvimento em camadas, C#.

ABSTRACT

This article aims to present a clinical system entitled "Control Medical Clinics," developed in a graduate work, and describe historical and concepts of resources that were used in the application, such as: Windows Operating System, SQL Server Database Management System, the programming language Object Oriented C #, and type of architecture used in the development phase, in layers. To report the

¹ Discente do Curso de Bacharelado em Sistemas de Informação do Centro Universitário de Franca Uni-FACEF

² Discente do Curso de Bacharelado em Sistemas de Informação do Centro Universitário de Franca Uni-FACEF

³ Docente do Curso de Bacharelado em Sistemas de Informação do Centro Universitário de Franca Uni-FACEF

features of the system will display images taken from the application during the study, which will be reviewed and clarified so that the reader can make a judgmental about the system based on the illustrations. It is conscious that the system does not do all the operations involved in a clinical process, however, that are functional are explained pictorially. Because of this, in the course of the work, is exposed a topic, which are reported in, the limitations of the system. This manner of presentation will be made so that the reader knows what he is and is not able to operate with the application. To describe the resources used throughout the development, a literature review was done by means of which explores the historical information and concepts of such resources. Being a literature review, the study does not contains the results expressed as numbers, so it can be considered as a result the importance of this system to the picture of health, and the way that each resource used helps in the development phase of the application . It remains open the possibility of a new study that reports possible new features and system resources.

Keywords: *Health Systems, Development inLayers, C#.*

1. INTRODUÇÃO

No presente estudo, objetiva-se abordar as funcionalidades e limitações do sistema, bem como os recursos utilizados para desenvolvimento do mesmo. Busca-se atingir um público alvo das Ciências Exatas, e a todos os interessados em saber um pouco do funcionamento, construção e limitações do sistema. Utiliza-se como metodologia para abordar os recursos presentes no sistema, a pesquisa bibliográfica.

O trabalho está estruturado da seguinte maneira: no item 2, é feito um breve histórico e apresentações de todos os Recursos Utilizados no Sistema, tais como: Windows, como Sistema Operacional, C# (CSharp), como linguagem de programação orientada a objetos, SQL Server, como base de dados e Desenvolvimento em Camadas, como arquitetura do projeto; no item 3 e 4 são abordadas e descritas as Funções de cada tela do sistema e suas limitações, respectivamente; enfim, no item 5 são elaboradas as Considerações Finais do trabalho, baseadas em linhas gerais do estudo propriamente dito.

Contextualizando, esse é o primeiro estudo do software. Portanto, mantém-se em aberto a possibilidade de um novo trabalho que aborde eventualmente novas funcionalidades do sistema.

2. RECURSOS UTILIZADOS NO SISTEMA

Alguns recursos foram utilizados ao decorrer do desenvolvimento da aplicação. Abaixo estão listados os principais, tais como: o Sistema Operacional (Windows), o Banco de Dados (SQL Server), a linguagem de programação (C# ou CSharp) e a arquitetura do projeto (Camadas):

2.1 Windows

Fundada em 1975 por Bill Gates e Paul Allen, dois amigos de colégio, a Microsoft inicialmente tinha em seus planos desenvolver em linguagem BASIC um software para o computador da MITS, o Altair 8800. Já em 1977 foi lançada uma nova versão do software, escrita em FORTAN para computadores baseados em CP/M, do inglês, Control Program/Microcomputer - sistema operacional em disco desenhado para os processadores Intel 8080, Intel 8085 e Zilog Z80 - (AVILA, 2007).

Em meados de 80, a Microsoft teve seu serviço contratado pela IBM, com a finalidade de desenvolver um sistema operacional para computador pessoal com sistema CP/M. No entanto, Gates e Allen não possuíam um sistema operacional completo, então decidem comprar o Q-DOS – sistema operacional da empresa Seattle Computer Products - pela quantia de 50 mil dólares. Após passar por melhoras, tornou-se na época, o que hodiernamente conhecemos por MS-DOS (PERON, 2009).

Contemporâneo a isso, a Apple desenvolvia uma interface gráfica para o Apple LISA, e já tinha interesse em utilizar um mouse terceirizado, da empresa XEROX. Não demorou muito, e Gates conseguiu convencer Steve Jobs a deixá-lo participar do projeto do novo computador da Apple – o Macintosh - e em 1982 a Microsoft inicia o desenvolvimento das aplicações. Todavia, Gates não só formou a parceria com Jobs, mas aproveitou também para copiar a interface gráfica do Macintosh e vende-la anos depois, em uma versão similar, no Oriente Médio, mais precisamente no Japão. Ciente do que Gates havia feito, Jobs quebra a aliança entre as duas empresas (PERON, 2009).

De acordo com (AVILA, 2007), a história do Windows inicia-se em 1981, antes mesmo do lançamento do Apple LISA, entretanto, o produto chega ao mercado somente em novembro de 85, intitulado Windows 1.0- versão vendida na época pela quantia de 100 dólares- que incluía a possibilidade de utilizar acessórios como mouse, e um recurso impressionante para o contexto histórico, que era a multitarefa, isto é, a capacidade de repartir a utilização do processador entre várias tarefas simultâneas. Igualmente, era acompanhado também de alguns jogos, e aplicativos como calendário, bloco de notas, calculadora, relógio, entre outros.

Ainda consoante (AVILA, 2007), abaixo segue o cronograma de surgimento das principais versões do Windows, até o Vista, lançada em janeiro de 2007:

- Windows 1.0, novembro de 1985;
- Windows 2.0, novembro de 1987;
- Windows 2.1/286 e Windows 2.1/386, maio de 1988;
- Windows 2.11, março de 1989;
- Windows 3.0, maio de 1990;
- Windows 3.1, abril de 1992;
- Windows for Workgroups 3.1, outubro de 1992;

- Windows for Workgroups 3.11, novembro de 1993;
- Windows 95, agosto de 1995;
- Windows 98, junho de 1998;
- Windows 98 SE, em maio de 1999;
- Windows Me, setembro de 2000;
- Windows NT 3.1, julho de 1993;
- Windows NT 3.5, setembro de 1994;
- Windows NT 3.51, maio de 1995;
- Windows NT 4.0, julho de 1996;
- Windows 2000, fevereiro de 2000;
- Windows XP, outubro de 2001. Versão que recebeu as atualizações SP1 e SP2;
- Windows Server 2003, abril de 2003 e;
- Windows Vista, janeiro de 2007.

2.2 SQL Server

O SGBD (Sistema Gerenciador De Banco de Dados) SQL Server é um produto pertencente à Microsoft, inicialmente criado em parceria com a Sybase - empresa de software que produz serviços e produtos relacionados ao gerenciamento de informação, mobilidade, ferramentas de desenvolvimento, data warehousing e OLAP - em 88. Já em meados da década de 90, a parceria terminou, e a Microsoft aprimorou novas funcionalidades ao aplicativo.

No hodierno, o SQL Server é um dos SGBD mais usados em todo mundo, competindo com alguns sistemas como o MySQL e Oracle (PACIEVITCH, 2011).

Segundo (PACIEVITCH, 2011), a adesão do SGBD pode ser de forma gratuita ou paga, atribuindo-se mais funcionalidades a versão paga. Apesar de seu preço ser um dos mais baixos dentro do mercado, o SQL Server não perde a qualidade e a integridade com que trata as informações.

Abaixo estão listadas as principais características de armazenamento que o sistema SQL Server (versão R2 2008) utiliza:

- HIERARCHY ID: É um modelo de programação flexível, que ajuda a organizar a hierarquia dos dados;

- Dados FILESTREAM: Permite o armazenamento de grandes dados binários diretamente no sistema NTFS de arquivos;
- Compressão de Backup: o backup exige menos I/O (entrada/saída) de disco, o que diminui os custos para manter o backup online e;
- Computações de Bloqueio: As Computações de Bloqueio permitem aos usuários aumentar a profundidade das hierarquias assim como a complexidade da computação (PACIEVITCH, 2011).

Na contemporaneidade, o SQL Server, apesar de ser um excelente SGBD, vem perdendo espaço no mercado, devido a dois fatores:

- Por ser de código privado, e ter como concorrente o MySQL, aplicativo de código aberto e;
- Pela dificuldade de instalação em SO (Sistemas Operacionais) que não sejam Windows.

2.3 C# (CSharp)

No ano de 1999, a Microsoft escolheu uma equipe, comandada por Anders Hejlsberg, para iniciar o desenvolvimento de uma linguagem chamada Cool. Passado um ano, em uma conferência de desenvolvedores, tal linguagem é apresentada, no entanto, com um novo nome: C# (C Sharp).

De acordo com (CARVALHO, 2011), embora a criação da linguagem tenha sido elaborada por vários desenvolvedores, o título de criador foi atribuído a Hejlsberg, criador também de duas outras linguagens: o Turbo Pascale e Delphi.

Sua criação foi baseada em JAVA e C++, sendo orientada a objetos, e considerada a linguagem símbolo do .NET(dot NET) por ter sido criada praticamente do zero, sem nenhum legado de bibliotecas e funções.

Em dezembro de 2001, a Microsoft submeteu o C# à ECMA, do inglês, (European Computer Manufacturers Association), para uma padronização, tendo sua solicitação aceita e recebendo no ano de 2003 um padrão ISO (CARVALHO, 2011).

2.4 Arquitetura em Camadas

A separação de sistemas complexos em camadas é uma das técnicas mais utilizadas no desenvolvimento de software. É um conceito que foi adaptado da

arquitetura de computadores, que utilizam as camadas para fazer chamadas ao sistema operacional e instruções do processador (CHIBA, 2007).

O termo camada pode significar separação física ou lógica, podendo ser também considerado como uma referência à separação de responsabilidades para a produção de softwares.

Precedentemente, os sistemas eram desenvolvidos em uma única camada, a qual recebia todas as funcionalidades da aplicação, tornando-se robusta e dificultando a manutenção. As inúmeras operações, de entrada, saída, verificação e acesso a base de dados, eram feitas no mesmo lugar. Na figura 1 é ilustrado este processo, mais conhecido como Camada Monolítica (MACORATTI, 2008).



Figura 1 – Exemplo de Camada Monolítica, onde todas as operações são feitas em uma única camada

Fonte: (MACORATTI, 2008)

De acordo com (MACORATTI, 2008) e (CHIBA, 2007), posteriormente, houve a necessidade de vários usuários compartilharem o acesso à base de dados simultaneamente, surgindo assim uma segunda camada. Nesse novo modelo, a base foi isolada em uma única máquina, e o usuário do sistema que necessitasse do acesso aos dados, compartilhava-o com tal máquina. Porém, logo surgiu um novo problema: o gerenciamento de versões; Uma vez que para cada alteração feita, os aplicativos dos usuários (Clientes) necessitavam de atualização. A figura 2 exemplifica o processo de separação da camada de acesso de dados.



Figura 2 – Exemplo de Aplicação em duas Camadas, onde a camada de acesso de dados é separada das demais

Fonte: (MACCORATTI, 2008)

Com a ascensão da internet, surge uma nova necessidade - separar a lógica de negócio da interface gráfica com usuário- ideia *sinequa non* os usuários da Web não conseguem acessar as aplicações a menos que as instalem em suas máquinas locais. Na figura 3 é mostrado este processo de separação da lógica de apresentação e lógica de negócios (MACORATTI, 2008).



Figura 3 – Exemplo de aplicação em três Camadas, onde a camada de lógica de apresentação é separada da lógica de negócios

Fonte: MACORATTI, 2008

Tais separações tornam as aplicações menos robustas, flexíveis, e de fácil manutenção, com uma possibilidade de alteração de forma independente, na qual se pode apenas alterar a interface gráfica com o usuário, sem modificar a lógica de negócio e o acesso aos dados. Na última figura, mostra-se de maneira convincente, que a separação em três camadas é o ideal para desenvolvimento Web, sobretudo.

3. FUNCIONALIDADES DO SISTEMA

Abaixo, serão aclaradas e ilustradas, respectivamente, todas as funcionalidades da aplicação:

O sistema inicia-se com a Tela de Login, na qual o usuário digita seu nome de usuário e senha, uma vez corretos, será exibida uma mensagem de boas vindas e ocorrerá a abertura da tela principal da aplicação. Caso os dados digitados forem incompatíveis com os dados previamente cadastrados na base de dados, aparecerá uma mensagem de erro, indicando que o usuário deve digitar novamente seus dados. A figura 4, ilustra este processo.

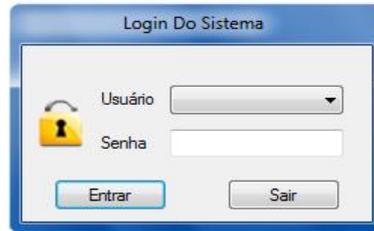


Figura 4 – Login do sistema

Após passar pela validação da tela de Login, é iniciada a tela principal da aplicação, na qual se percebe um menu principal no canto superior esquerdo, surge ao lado, o nome do usuário que está logado no sistema, e ainda no canto superior direito, observa-se a data e hora do computador. Abaixo, no canto inferior esquerdo, têm-se os dados do consultório, os quais são configurados no menu Configurações, que será explorado mais adiante. É interessante ressaltar o sistema de Backup da base de dados. Toda vez que o usuário clicar no botão sair ou tentar fechar a aplicação pelo atalho ALT + F4, será exibida uma mensagem de confirmação da saída do sistema, se tal confirmação resultar em “SIM”, uma mensagem será exibida, dizendo: “O backup do Sistema foi feito com sucesso!”. Com isso, toda vez que a aplicação for fechada, será feito o backup no mesmo instante, localizado em uma pasta pré-criada no C:/. Na figura 5, observa-se a tela principal do projeto.



Figura 5 – Tela principal

O Cadastro de Pacientes é uma tela do sistema, na qual o usuário define todos os dados do paciente, devendo obedecer às regras de obrigatoriedade dos campos. Para cadastrar um novo paciente, o usuário deve preencher todos os campos e clicar no botão Salvar, uma vez que todos os campos forem corretamente preenchidos e validados (no caso de CPF e RG), o paciente estará incluso na base de dados. Se a operação desejada for alteração, basta somente o usuário selecionar o paciente através de um combo (Localizar Pacientes), logo os campos referentes àquele paciente estarão preenchidos, bastando somente alterar o que desejar e clicar no botão Alterar. Para a exclusão de um paciente, basta selecioná-lo no combo e clicar no Botão Excluir, logo uma mensagem de confirmação irá aparecer, uma vez que confirmada no botão Sim, o paciente será removido da base de dados. Para limpar os campos é simples, apenas com um clique no botão Limpar e todos os campos retornarão ao estado inicial. Quando o usuário clica no botão Novo, os campos voltam ao estado inicial para que um novo paciente seja preenchido. A figura 6 ilustra exatamente este processo.

Cadastro De Pacientes

Localizar Pacientes

ID

Nome

Data De Nascimento Sexo

Possui CPF Sim Não Possui RG Sim Não

CPF RG

Endereço

Bairro

Cidade UF

CEP

Telefone Residencial () Telefone Comercial () Celular ()

(*) Campos obrigatórios

Novo Salvar Alterar Limpar Excluir

Figura 6 – Cadastro de Pacientes

Similar ao Cadastro de Pacientes, o Cadastro de Médicos atua da mesma maneira a respeito das operações de Inclusão, Remoção e Alteração. Vale ressaltar que é permitido apenas um médico cadastrado por CRM, ou seja, não são permitidos, cadastros diferentes, ou mesmo iguais, possuírem o mesmo CRM, pois ele é um identificador, logo deve ser único. Observa-se na figura 7, que é um cadastro muito simples para manipulação dos dados e possui poucos campos.

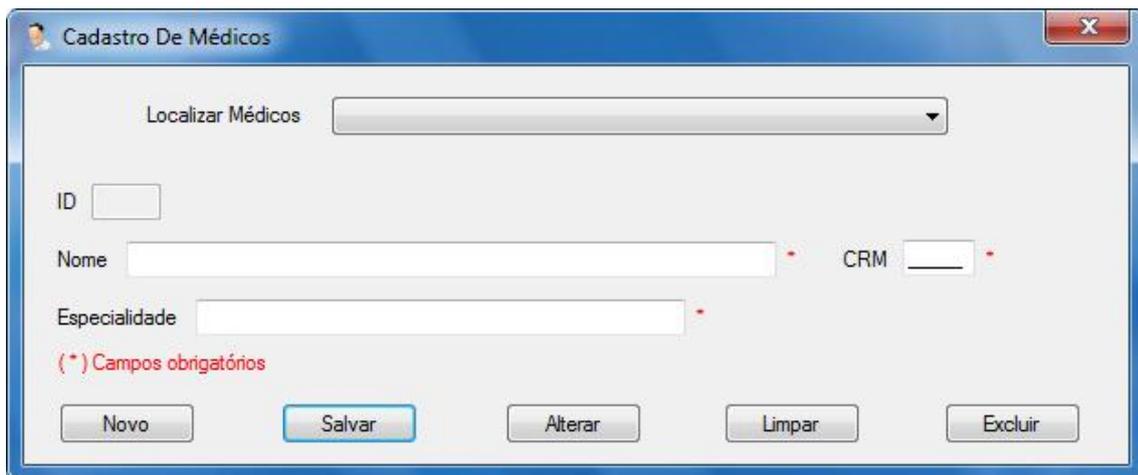


Figura 7 – Cadastro de Médicos

Com as operações idênticas aos Cadastros anteriormente ditos, o Cadastro de Doenças têm uma diferença crucial para o funcionamento adequado do sistema. Quando se fala que em Doenças, existe um padrão denominado CID10 (Cadastro Internacional de Doenças), que é um banco de dados atualizado frequentemente, que contém todas as doenças já descobertas na história da humanidade. O banco possui aproximadamente 14.200 doenças. O diferencial desse sistema, é que já vêm incluso nele todas as doenças pré-cadastradas, já com seus devidos identificadores. A figura 8 ilustra este processo.

Figura 8 – Cadastro de Doenças

O Agendamento de consultas é mais um dos requisitos de um sistema como este. Na figura 9, logo abaixo, percebe-se que é um processo simples, no qual o usuário define o paciente, o médico, e os dados da consulta, tais como: data, hora e o tipo de atendimento (particular ou convênio); e como isso ele pode operar normalmente consoante os Cadastros, com as operações de inclusão, alteração e remoção. Para selecionar um consulta, diferentemente dos Cadastros, o usuário seleciona-a pela tabela e não por um combo, como era feito anteriormente.

	Id_Agendamento	Nome_Paciente	Nome_Medico	Data	Hora	Tipo_atendimento
*						

Figura 9 – Agendamento de Consultas

O Banco de diagnóstico é um dos módulos mais complexos desse sistema. Nele, vê-se que há uma gama de informações a serem preenchidas, e com algumas muito interessantes, como é o caso da imagem de ressonância magnética, a qual o usuário (no caso médico) pode importar do próprio computador a imagem de ressonância que pertencer ao paciente. Tem também alguns campos para preenchimento não obrigatório, nos casos de Exames Laboratoriais e Radiológicos. Vale ressaltar que a imagem não é obrigatória, uma vez que não são todos os pacientes que necessitam passar por esse tipo de exame. As operações de Inclusão, Alteração e Remoção do Diagnóstico são idênticas aos Cadastros referidos anteriormente, porém, nos Cadastros, o esquema de selecionar os dados era através de um combo, já no Banco de Diagnóstico, ele é executado através da tabela, similar ao módulo de Agendamento. O processo está ilustrado na figura 10.

(*) Campos obrigatórios

Id_Diagnostico	Nome_Paciente	Nome_Medico	Data	Anamnese	Doenca	Ex_Laboratoriais	Ex_Radiologicos	Caminho_Foto
*								

Figura 10 – Banco de Diagnósticos

O módulo de impressão é dividido em três partes: Receitas, Atestados e Recibos. O usuário deverá preencher os campos obrigatórios e clicar no botão Imprimir. Logo será aberta uma janela para a escolha da impressora e suas características. Após, é necessário somente confirmar no botão Ok. Nas figuras 11, 12 e 13 serão ilustradas as Impressões de Receitas, Atestado e Recibos, respectivamente.

A janela 'Impressão De Receitas' apresenta os seguintes elementos:

- Campos de seleção para 'Pacientes' e 'Médicos'.
- Um campo de texto para 'Crm'.
- Uma área grande e vazia rotulada 'Medicação'.
- Um rodapé com o texto '(*) Campos obrigatórios'.
- Dois botões: 'Imprimir' e 'Limpar'.

Figura 11 – Impressão de Receitas

A janela 'Impressao De Atestado' apresenta os seguintes elementos:

- Campos de seleção para 'Pacientes' e 'Médicos'.
- Um campo de texto para 'Crm'.
- Dois campos de seleção para 'Data e hora da consulta'.
- Dois campos de seleção para 'Período De Afastamento' (De e Até).
- Um rodapé com o texto '(*) Campos obrigatórios'.
- Dois botões: 'Imprimir' e 'Limpar'.

Figura 12 – Impressão de Atestados

Figura 13 – Impressão de Recibos

Dentro do módulo de utilitários, o sistema contém o Cálculo de IMC, um Conversor de Temperatura e uma chamada a Calculadora do Windows, para essa última, basta apenas que o usuário clique em Calculadora, item do menu Utilitários, e através de uma chamada de processo, ela será aberta, podendo ser utilizada quando for necessário realizar algum cálculo. A seguir, estes itens, em ordem, serão listados nas figuras 14, 15 e 16 respectivamente.

Figura 14 – Cálculo de IMC

Figura 15 – Conversor De Temperatura



Figura 16 – Calculadora do Windows

O menu Configurações pode ser acessado somente por usuários que têm o privilégio de administrador do sistema, pois nele é definido através de seus itens, os dados do Consultórios, os usuários do sistema e a senha interna utilizada para acessar o Banco de Diagnósticos.

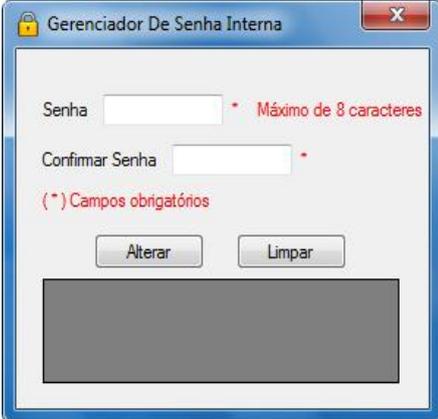
Os Dados do Consultório já vem pré-cadastrados no sistema, havendo somente a possibilidade de alteração. Não é permitida a inclusão de novos dados, tampouco, remoção. Esses, por sua vez, são definidos como: nome do consultório, endereço, telefone, entre outros. São eles que os que identificam a clínica na tela inicial da aplicação. Este processo está ilustrado na figura 17.

A screenshot of a web application form titled "Dados Do Consultório". The form contains several input fields: "Nome", "Endereço", "Bairro", "Cidade", "UF" (a dropdown menu), and "Telefone" (with a small icon for international dialing). Each field has a red asterisk to its right, indicating it is mandatory. Below the fields is a red text label "(*) Campos obrigatórios". At the bottom of the form are two buttons: "Alterar" and "Limpar".

Figura 17 – Dados do Consultório

No Gerenciador de Senha Interna, observa-se uma interface gráfica simples, na qual o usuário define uma senha, tendo somente a possibilidade de alterá-la

posteriormente. É importante lembrar que essa já vem definida junto à aplicação com os dígitos “admin”.



Gerenciador De Senha Interna

Senha * Máximo de 8 caracteres

Confirmar Senha *

(*) Campos obrigatórios

Alterar Limpar

Figura 18 – Gerenciador de Senha Interna

No Gerenciador de Login, por exemplo, o administrador, define um nome para o usuário, uma senha e o nível de acesso dele no sistema (Administrador, Operador ou Técnico). O sistema de Inclusão, Alteração e Remoção é equivalente ao sistema de Agendamento, nele é escolhido o registro na tabela para que possa ser alterado ou excluído. Neste caso, o sistema já vem com um usuário pré-cadastrado, para que possa ser efetuado o primeiro acesso no sistema. Usuário este definido como (Usuário: admin; Senha: admin; Nível: ADMINISTRADOR) Vale ressaltar que é possível somente um usuário ser cadastrado com o mesmo nome. A figura 19 exemplifica o processo referido.

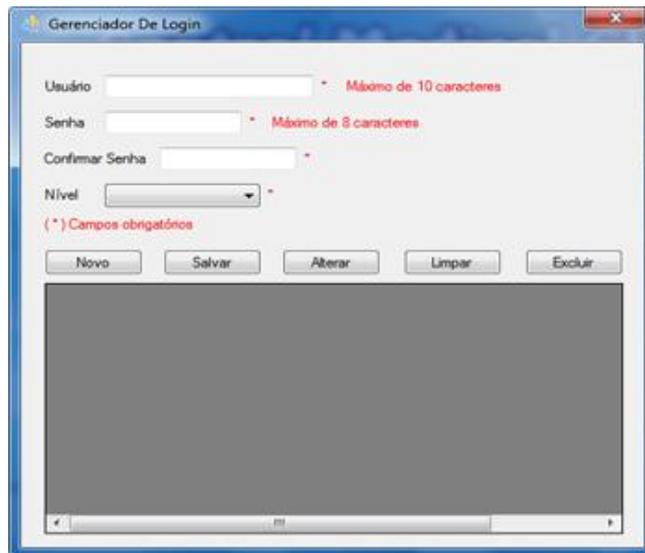


Figura 19 – Gerenciador De Login

O suporte do sistema é apenas uma janela na qual são informados os e-mails dos desenvolvedores, caso haja a necessidade de algum contato. Ela se encontra ilustrada na figura 20.



Figura 20 – Suporte

A janela Ajuda, também é importante dentro do sistema, pois facilitará e proporcionará ao novo usuário alguns links de cada módulo, e aparecerá no centro da tela, um texto explicativo que esclarecerá algumas dúvidas para que a interação com os módulos possa ser iniciada. Na figura 21, tem-se um exemplo deste processo.



Figura 21 – Ajuda

4. LIMITAÇÕES DO SISTEMA

Dentro das limitações do sistema, destacam-se:

- Não funciona em rede extranet, somente em Intranet;
- Não possui um módulo financeiro;
- Não possui um canal de comunicação entre os computadores em rede;
- Funciona apenas em Sistemas Operacionais Windows;
- Funciona somente com Base de dados SQL Server;

5. CONSIDERAÇÕES FINAIS

Nota-se que para o desenvolvimento de um sistema como este, é extremamente necessário e relevante, o envolvimento e utilização de várias tecnologias, todas essenciais e importantes para esta aplicação, como a arquitetura em camadas, que proporciona maior flexibilidade e facilidade na manutenção do código, além de deixá-lo menos robusto. Por ser um primeiro contato com algumas destas tecnologias, foi possível o desenvolvimento básico dos módulos deste sistema. Pretende-se futuramente, ao estudar-se com mais profundidade alguns destes recursos, atribuir uma variação de funcionalidades, usabilidades e manutenibilidades, utilizando novos métodos e tecnologias com um nível de complexidade ainda maior.

REFERÊNCIAS BIBLIOGRÁFICAS

PERON, Marluce. **“A História da Microsoft”**. TECMUNDO, 2009. Disponível em: <http://www.tecmundo.com.br/2068-a-historia-da-microsoft.htm>. Acessado em: 14/11/2011 às 20h10m.

AVILA, Steeven. **“História da Microsoft”**. INFOESCOLA, navegando e aprendendo, 2007. Disponível em: <http://www.infoescola.com/informatica/historia-da-microsoft>. Acessado em: 12/11/2011 às 21h.

PACIEVITCH, Yuri. **“SQL Server”**. INFOESCOLA, 2011. Disponível em: <http://www.infoescola.com/informatica/sql-server/>. Acessado em 11/11/2011 às 14h20m.

CARVALHO, Israel. **“A História do C#”**. Israel Carvalho, 2011. Disponível em: <http://www.israelcarvalho.com.br/?p=101>. Acessado em 08/11/11 as 08h32m.

CHIBA, Cláudio; NARDI, Alexandre. **“Desenvolvimento em Camadas”**. MSDN, 2007. Disponível em: http://www.microsoft.com/brasil/msdn/tecnologias/arquitetura/Layers_Developing.aspx. Acessado em: 09/11/2011 às 22h30m.

MACORATTI, José Carlos. **“Padrões de Projeto: O modelo MVC - ModelViewController”**. Macoratti.net, 2008. Disponível em: http://www.macoratti.net/vbn_mvc.htm. Acessado em: 01/11/2011 às 18h25m.