

CENTRO UNIVERSITÁRIO UNIFACVEST  
CURSO DE CIÊNCIA DA COMPUTAÇÃO  
LUAN BARBOSA HUBNER

**PESC: PRONTUÁRIO ELETRÔNICO DA SAÚDE DO CIDADÃO**

Lages- SC  
2016

[guitarmak43@gmail.com](mailto:guitarmak43@gmail.com)

LUAN BARBOSA HUBNER

## **PESC: PRONTUÁRIO ELETRÔNICO DA SAÚDE DO CIDADÃO**

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

Orientador: Prof. Afonso Alberto Fernandes de Oliveira

Co-orientador: Prof. Marcio José Sembay

Co-orientador: Marlon Amarin

Lages- SC

2016

LUAN BARBOSA HUBNER

## **PESC: PRONTUÁRIO ELETRÔNICO DA SAÚDE DO CIDADÃO**

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

Orientador: Prof. Afonso Alberto Fernandes de Oliveira

Co-orientador: Prof. Marcio José Sembay

Co-orientador: Marlon Amorin

Lages, SC \_\_\_/\_\_\_/2016.

Nota \_\_\_\_\_

---

Prof. MSc. Marcio José Sembay  
Coordenador do curso de graduação

Lages- SC

2016

## **AGRADECIMENTOS**

Dedico esse trabalho a Deus, nosso Pai e criador, e agradeço a ele, pois somente por vontade dele é que as coisas acontecem, e se cheguei até aqui foi porque ele permitiu. Agradeço aos meus pais, Ananias e Zaira que lutaram muito, para que eu pudesse estudar, sempre me dando apoio independente se as vezes as condições não eram boas, mas continuavam firmes no propósito. A minha namorada Siéle que sempre esteve ao meu lado me apoiando em tudo que precisei, aos colegas de trabalho que me ajudaram com as entrevistas e ideias sobre o projeto, meu amigo e co-orientador Marlon Amorin que me auxiliou no desenvolvimento juntamente com o Professor Afonso meu orientador. Meus amigos de faculdade que em todos os momentos de dificuldades se fizeram presentes, não posso deixar de agradecer aos meus professores companheiros de caminhada, em especial ao Márcio Sembay, Afonso Alberto, Igor Muzeka e Ingrid, que nunca se negaram em ajudar em tudo que precisei.

Obrigado a todos, sem vocês nada disso teria acontecido.

Aluno<sup>1</sup>: Luan Barbosa Hubner

Orientador<sup>2</sup>: Afonso Afonso Alberto Fernandes de  
Oliveira

## RESUMO

O presente trabalho conclui a pesquisa sobre a gestão de prontuário eletrônico para unidade básica de saúde de Anita Garibaldi/SC. Buscou-se através de pesquisa exploratória entender e compreender os problemas que eram relatados na organização dos prontuários, no agendamento manual de consultas, nas quais áreas de abrangência, nas soluções de melhoria de atendimento e organização na unidade básica de saúde (UBS). Sendo o objetivo deste projeto, trazer organização, agilidade e proteção aos dados dos pacientes que possuem registros em prontuários de papeis, fazendo com que o profissional tenha um maior controle do que está sendo feito sem que se perca muito tempo com serviços burocráticos. Foram concluídos os objetivos em relação as entrevistas com profissionais que trabalham na UBS. Foi utilizada como norte da pesquisa, referenciais bibliográficos em artigos acadêmicos, livros e pesquisa online. Os resultados encontrados foram de grande valia junto a UBS, como segurança nos dados de cada paciente, organização dos trabalhos e agilidade no processo de atendimento, pois o software também conta com um sistema que gera e realiza a leitura do código de barras, o qual melhorou o processo de agendamentos de consultas nos quesitos confiabilidade e tempo de resposta.

**Palavras Chave:** *Prontuário eletrônico, UBS, Código de Barras.*

---

<sup>1</sup>Acadêmico do Curso de Ciência da Computação 8º fase, Disciplina de Trabalho de Conclusão de Curso II do Centro Universitário UNIFACVEST.

<sup>2</sup>Graduado em Ciência da Computação, Ciências Contábeis, especialização em Controle da Gestão Pública Municipal, especialização em Desen. De Sistemas Para Web - Sphere e Dot. NET e mestrado em Engenharia Química.

## ABSTRACT

This paper concludes the research on the management of electronic medical records for Basic Health Unit of Anita Garibaldi / SC. It was sought through exploratory research to understand the problems that were reported in the organization of medical records, in the manual scheduling of appointments, in which areas of coverage, in the solutions of improvement of care and organization in the Basic Health Unit (BHU). Being the goal of this project, to bring organization, agility and protection to the data of patients who have records in paper charts, making the professional have a greater control of what is being done without wasting much time with bureaucratic services. The objectives were concluded in relation to interviews with professionals working at BHU. It was used as the north of the research, bibliographic references in academic articles, books and online research. The results found were of great value with BHU, such as data security of each patient, work organization and agility in the service process, because the software also has a system that generates and performs bar code reading, which improved the scheduling process in terms of reliability and response time.

**Keywords:** *Electronic Medical Record, UBS, Barcode.*

## LISTA DE FIGURAS

<b>Figura 1:</b> Tela Delphi Embarcadero Seattle _____	21
<b>Figura 2:</b> Tela banco de dados SQLServer Management Studio _____	22
<b>Figura 3:</b> Leitor código de barras _____	24
<b>Figura 4:</b> Exemplo de código de barras _____	25
<b>Figura 5:</b> Diagrama de caso de uso _____	34
<b>Figura 6:</b> Diagrama de atividades atendimento _____	35
<b>Figura 7:</b> Diagrama de banco de dados do sistema _____	37
<b>Figura 8:</b> Tela login de acesso ao sistema _____	38
<b>Figura 9:</b> Tela login de acesso ao sistema _____	39
<b>Figura 10:</b> Tela de cadastro de cidadão _____	40
<b>Figura 11:</b> Tela de agendamento _____	41
<b>Figura 12:</b> Tela de cadastro de atendimento _____	42
<b>Figura 13:</b> Tela de backup _____	43

## LISTA DE SIGLAS

UBS	-	Unidade Básica de Saúde
ONU	-	Organização das Nações Unidas
CFM	-	Conselho Federal de Medicina
PEP	-	Prontuário Eletrônico do Paciente
SE	-	Sistemas Especialistas
LAN	-	<i>Local Area Network</i>
AC	-	Autoridade Certificadora
SUS	-	Sistema Único de Saúde
DATASUS	-	Departamento de Informática do SUS
IDE	-	<i>Integrated Development Environment</i>
SQL	-	<i>Structured Query Language</i>
IBM	-	<i>International Business Machines</i>
ACS	-	Agente Comunitário de Saúde
UML	-	<i>Unified Modeling Language</i>
SMS	-	<i>Short Message Service</i>
GB	-	Giga Byte
TB	-	Tera Byte
RAM	-	<i>Random-Access Memory</i>
HD	-	<i>Hard Disk</i>
CMS	-	Cartão municipal de saúde
CNS	-	Cartão nacional de saúde
DER	-	Diagrama de entidade relacionamento



## LISTA DE QUADROS

<b>Quadro 1.</b> Sistemas de informação na área da saúde .....	15
<b>Quadro 2.</b> Principais níveis que podem ser apresentados na implantação .....	16
<b>Quadro 3.</b> Vantagens e desvantagens no prontuário eletrônico versus o prontuário em papel	18
<b>Quadro 4.</b> Normas técnicas para prontuários eletrônicos.....	19
<b>Quadro 5.</b> Características Saúde Fly	44
<b>Quadro 6.</b> Cronograma do TCC I.....	45

## SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 Justificativa .....	12
1.2 Importância .....	13
1.3 Objetivos do Projeto .....	13
1.3.1 Objetivos Gerais .....	13
1.3.2 Objetivos Específicos.....	13
2 REVISÃO BIBLIOGRÁFICA .....	14
2.1 Prontuário Eletrônico.....	14
2.1.1 Prontuário Eletrônico do Paciente.....	14
2.1.2 Segurança e ética nas informações do prontuário eletrônico .....	16
2.1.3 Vantagens e desvantagens dos prontuários .....	17
2.1.4 Normas técnicas para prontuários eletrônicos.....	18
2.1.5 Prontuário eletrônico da atenção básica no Brasil .....	19
2.2 Ferramentas a serem usadas no projeto .....	20
2.2.1 Plataforma .....	20
2.3 Código de barras .....	24
2.3.1 Como funciona o código de barras.....	24
2.4 O município de Anita Garibaldi.....	25
2.4.1 Breve histórico sobre o município de Anita Garibaldi.....	25
2.4.2 A saúde no município de Anita Garibaldi.....	26
2.4.3 Secretaria municipal de saúde de Anita Garibaldi .....	26
2.4.4 Como funcionavam os prontuários na UBS de Anita Garibaldi .....	27
2.4.5 Hospital municipal Frei Rogério .....	28
2.4.6 Prontuários no hospital Frei Rogério .....	28
3 METODOLOGIA.....	29
3.1 Documentação .....	29
3.2 Natureza da pesquisa .....	29
3.3 Tipo da pesquisa .....	29
3.4 Técnicas de pesquisa.....	30
3.5 Coleta de dados .....	30
3.6 Limitações da pesquisa .....	31
4 PROJETO .....	32

4.1 Hardware.....	32
4.1 Requisitos do sistema.....	32
4.2 Ferramentas do projeto .....	33
4.3 Modelagem UML .....	33
4.3.1 Diagrama de caso de uso .....	33
4.3.2 Diagrama de atividades .....	35
4.3.3 Diagrama Entidade Relacionamento.....	36
4.4 Telas do sistema.....	37
4.4.1 Tela login de acesso ao sistema.....	37
4.4.2 Tela principal.....	38
4.4.3 Tela de cadastro de cidadão .....	39
4.4.4 Tela de agendamento.....	40
4.4.5 Tela de atendimento .....	41
4.4.6 Tela de Backup.....	43
5 TRABALHOS CORRELATOS .....	44
5.1 Saúde Fly .....	44
6 CRONOGRAMA .....	45
7 RESULTADOS .....	46
REFERÊNCIAS .....	47
APÊNDICE A – Configuração geral do sistema.....	50
APÊNDICE B – Agendamento de consulta .....	54
APÊNDICE C – <i>Login</i> no sistema .....	66
ANEXO A – Termo de aceite de implantação de trabalho acadêmico .....	69

## **1 INTRODUÇÃO**

Com o resultado dos grandes avanços da tecnologia de informação e a crescente mudança para sistemas de gestão, muitas áreas migraram para esse conceito. A área de saúde infelizmente ainda é uma das áreas mais crítica que temos em nosso país, muitas vezes por falta de recursos para contratar profissionais ou até mesmo para adquirir novos equipamentos, muitas unidades básicas de saúde não migram para esses sistemas e continuam a operar com prontuários e atendimentos todos em papel, acondicionados em pastas, envelopes e gavetas, sem nenhuma segurança e sigilo dos dados.

De acordo com o documento do Conselho Econômico e Social da ONU (Organização das Nações Unidas), no contexto da saúde, vêm contribuir com os cidadãos, uma vez que servem para melhorar os processos de tomada de decisões, contribuem para as trocas de experiências de domínios específicos e também aos cuidados pessoais, bem como a assistência qualificada e, igualmente, para reforçar a eficácia das instituições de saúde. Contribuem, ainda, para o desenvolvimento de sistemas de informações, das organizações de saúde, para a criação e implantação de redes públicas e particulares de saúde, para a construção de sistemas inteligentes, entre outros (ONU, 2001).

A grande discussão é, qual característica deve ter um sistema de informação para ser capaz de lidar com as diferentes facetas, que o próprio modelo de saúde apresenta como desafio e como esse sistema pode ser de tamanho aproveitamento?

Em resposta à pergunta acima, na conclusão deste trabalho foi possível notar o grande aproveitamento que se teve o PESC com a implantação na UBS, agilizou o processo de agendamento de consultas e foi de fácil entendimento dos seus usuários.

### **1.1 Justificativa**

Devido as dificuldades encontradas na hora do agendamento da consulta, surgiu a ideia de desenvolver um software que auxilie e reduza os problemas ali encontrados. Não cabe ao sistema de gestão de saúde ditar as regras na unidade, mas ele deve ser capaz de acompanhar e dar conta das decisões e estratégias tomadas pelos gestores da mesma. Por mais adequado que o sistema seja, ele não será perfeito no primeiro desenvolvimento, é natural que existam versões posteriores a serem incrementadas na busca de melhoramentos.

## **1.2 Importância**

Um dos maiores problemas da saúde pública, é a falta de agilidade no momento de marcar as consultas. Com esse sistema é possível agilizar e melhorar esse processo, porque facilita a identificação do paciente e a localização das informações de seu prontuário, de forma que não se perca tempo procurando, um-a-um, em meio a milhares de papéis.

## **1.3 Objetivos do Projeto**

### ***1.3.1 Objetivo Geral***

Desenvolver uma ferramenta que auxilie no processo de atendimento e agendamento de consulta, que ao ser utilizada seja de fácil entendimento, e ao paciente traga mais segurança para suas informações.

### ***1.3.2 Objetivos Específicos***

Conforme os objetivos gerais, foram traçados os seguintes objetivos específicos que o software desempenha:

- a) Organizar informações e assim minimizar os problemas encontrados na hora do agendamento do cidadão evitando o acúmulo de pastas sobre o balcão de atendimento.
- b) Garantir maior segurança dos dados contidos em cada prontuário, evitando perda e até mesmo vazamento de informações, de forma que só pessoas autorizadas tenham acesso.
- c) Monitorar o histórico de cada paciente, com maior facilidade.
- d) Garantir transparência tanto nos dados dos prontuários como também entre as funções internas da UBS.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Prontuário Eletrônico

Quando falamos de prontuário eletrônico percebemos que possui vários nomes, como: Prontuário do Paciente, Prontuário Familiar, Prontuário Médico, Prontuário Hospitalar, Registro do Paciente entre outros.

#### 2.1.1 Prontuário Eletrônico do Paciente

Independentemente de como é chamado o prontuário eletrônico, segundo Roger e Gaunt (1994, p.194), é “uma memória escrita das informações clínicas, biológicas, diagnósticos e terapêuticas de uma pessoa, às vezes individual e às vezes coletiva, constantemente atualizado”.

Por sua vez, o CFM (Conselho Federal de Medicina), no Artigo 1º da Resolução de nº 1.638/2002, define o Prontuário do Paciente como:

*Um conjunto de informações, sinais e imagens registradas, geradas a partir de fatos, acontecimentos e situações sobre a saúde do paciente e a assistência a ele prestada, de caráter legal, sigiloso e científico, utilizado para possibilitar a comunicação entre membros da equipe multiprofissional e a continuidade da assistência prestada ao indivíduo (PORTALDOMEDICO,2002).*

O PEP (Prontuário Eletrônico Do Paciente), foi desenvolvido com a ideia de melhorar o atendimento, e para que os profissionais da área de saúde, quando necessitassem saber algo ocorrido com algum cidadão, pudessem ter um acesso mais fácil a essas informações de atendimento, realizado anteriormente em cada paciente (MASSAD, 2003).

De acordo com o *Computer - Based Patient Record Institute* apud Murphy, Hanken e Waters (1999), “o PEP é um registro computadorizado do paciente, cuja informação é mantida eletronicamente sobre o status e cuidados de saúde de um indivíduo durante toda a sua vida”.

É uma ferramenta tanto de registro quando administrativa, onde está contido registro do paciente desde seu nascimento até o dia de sua morte, todas as suas informações, dados pessoais e histórico familiar, quais medicamentos ele fez uso, quais exames ele fez e quais os resultados obtidos pelo mesmo (COSTA apud MARTINS, 2001).

De acordo com Costa (2003), os fatores clínicos que impulsionaram a implementação de um PEP foram: possibilidade de compartilhar informações, melhoria da qualidade da assistência, aumento da eficiência de processos clínicos e redução de erros médicos.

Segundo Wechsler apud Floriani (2003), os sistemas da área da saúde ou como são mais conhecidos por Informática Médica, são divididos em três grupos, os quais podemos ver no quadro 1 a seguir:

**Quadro 1.** Sistemas de informação na área da saúde

Sistemas de Apoio à Decisão (SAD)	Via de regra, são softwares que apoiam o médico e sua decisão clínica sendo caracterizados tanto os aplicativos que utilizam dados e informações como também aqueles que utilizam o conhecimento, ou os chamados SE (Sistemas Especialistas);
Telemedicina	Podendo ser compreendida como a utilização dos recursos de telecomunicação e informática quando aplicados ao procedimento diagnóstico e terapêutico à distância, consultas, orientações, educação médica continuada à distância, com imagens transmitidas por meio eletrônico;
Prontuários Eletrônicos de Pacientes (PEPs)	Inicialmente desenvolvidos para documentar as informações do quadro clínico (estado de saúde) de um paciente.

Fonte: Wechsler apud Floriani (2003).

Não serve apenas para registrar o histórico do paciente, mais possui várias outras funções, tais como: facilitar o compartilhamento de informações entre os profissionais, é uma fonte de pesquisa e avaliação, uma fonte de informação, fornece uma enorme organização e gerência para os estabelecimentos de saúde (BENTES PINTO, 2006).

Quando aplicado em hospitais, recebe dados de algumas áreas diferentes comparado a UBS. Agrega dados de procedimentos cirúrgicos, da farmácia, dos médicos, da enfermagem, do pronto-socorro, do ambulatório, da internação e dos laboratórios. O benefício de se integrar as informações das diferentes áreas do hospital, é a otimização do fluxo de trabalho, que torna os processos menos burocráticos e mais eficientes, assim o profissional tem mais tempo para o cuidado com o paciente (POISSANT, 2005).

O processo de implementação do PEP passa por diversas etapas, qual se destacam pelo menos 5 níveis que as instituições podem apresentar durante a implementação quais podemos observar no quadro 2 a seguir.

**Quadro 2.** Principais níveis que podem ser apresentados na implantação

Nível 1	Em cada instituição (clínica, hospital, Operadora de plano de saúde, etc.), papel e registro eletrônico ainda coexistem para registro das informações do paciente.
Nível 2	O sistema já está mais informatizado. Um exemplo é a existência de agregação de imagens capturadas via “scanners”. Em geral, nesse nível existe pouca integração entre os departamentos da instituição.
Nível 3	Requer que o sistema esteja implementado em toda a instituição e contenha elementos como integração com sistema de gerenciamento da prática médica, alertas clínicos e programas de educação ao paciente. Neste nível os requisitos de confidencialidade, segurança e proteção dos dados são atendidos.
Nível 4	O escopo da informação é muito mais amplo. As informações são compartilhadas com outros integrantes do sistema de saúde. Assim, este nível requer que a identificação do paciente seja única e feita em nível nacional.
Nível 5	Nesse nível o registro denomina-se Registro Eletrônico de Saúde: inclui uma rede de prestadores de serviço médico, tendo o paciente como centro. A informação não é baseada somente nas necessidades do serviço de saúde; é baseada na saúde e doença do indivíduo e da comunidade.

Fonte: Massad et al. (2003) apud Waagemann (1996).

### ***2.1.2 Segurança e ética nas informações do prontuário eletrônico***

Segundo Durand (1999), ética pode ser definida como, “uma reflexão sobre as questões fundamentais do agir humano (fim e sentido da vida humana, fundamento da obrigação e do dever, natureza do bem e do mal, valor da consciência moral, etc.)”. Quando o assunto é ética e privacidade de informações na área da saúde tudo se torna peculiar e delicado, pois o principal enfoque nesta área é a vida das pessoas.

O profissional que recebe as informações, registra e manipula os dados de cada paciente, tem a responsabilidade de guardar a integridade, e preservar com total sigilo as informações que ali constam (MASSAD, 2003).

As informações que estão contidas no prontuário eletrônico do paciente são de propriedade do paciente, tanto os profissionais da área da saúde quanto as instituições que tem acesso a essas informações devem ser fiéis depositários do prontuário. As informações são obtidas pelo paciente, durante o atendimento, através do resultado de exames e outros procedimentos realizados. Todo profissional da saúde não só o médico, mas todos os demais profissionais, só tem autorização para o contato com as informações de cada paciente, pela necessidade profissional, então não cabe só ao médico o dever de guardar segredo, mas sim a todos que tenham acesso a estes dados (MOTTA, 2003).



Para que estes dados tenham um maior sigilo, a segurança no software também não pode falhar. Segundo Martins (2004), a segurança da informação está suportada pelos seguintes princípios básicos:

- a) Integridade: Processo que tem objetivo de assegurar que os dados não sejam alterados por pessoas não autorizadas;
- b) Confidencialidade: Proteger as informações para que não sejam vistas de forma indiscriminadamente;
- c) Disponibilidade: Disponibilizar o acesso ao sistema apenas para usuários autorizados;
- d) Autenticação: Processo de verificação da identidade de uma pessoa;
- e) Autorização: Limitações de acesso a áreas dentro do sistema;
- f) Auditoria: Assegurar que toda atividade do usuário possa ser visualizada posteriormente;
- g) Controle de acesso por login e senha: Acesso ao sistema só por pessoas cadastradas;
- h) Biometria: Acesso só ao usuário estando pessoalmente ao sistema;
- i) Firewall: Limitar e controlar o acesso de terceiros a rede local interna (LAN).

Como vimos, a segurança no sistema depende de uma variedade de condições, e deve ser feita com muita atenção.

### ***2.1.3 Vantagens e desvantagens dos prontuários***

Sabemos que os sistemas de informação são muito eficientes, e na área da saúde são inúmeras suas vantagens, mas nem tudo tem só o lado bom. O quadro 3 a seguir nos mostra algumas vantagens e desvantagens no prontuário eletrônico versus o prontuário em papel segundo Bentes Pinto (2006):

**Quadro 3.** Vantagens e desvantagens no prontuário eletrônico versus o prontuário em papel

<b>Prontuário</b>	<b>Vantagens</b>	<b>Desvantagens</b>
Prontuário em Papel	<ul style="list-style-type: none"> <li>a) Maior liberdade na maneira de escrever;</li> <li>b) Facilidade no manuseio;</li> <li>c) Não requer treinamento para o seu manuseio;</li> <li>d) Nunca fica “fora do ar”.</li> </ul>	<ul style="list-style-type: none"> <li>a) Ilegibilidade em decorrência dos “ hieróglifos” da equipe de saúde;</li> <li>b) Espaço único;</li> <li>c) Ambiguidade,</li> <li>d) Perda frequente de informações;</li> <li>e) Multiplicidade de pastas;</li> <li>f) Dificuldade de acesso e de pesquisa coletiva;</li> <li>g) Falta de padronização; e fragilidade do papel.</li> </ul>
Prontuário Eletrônico	<ul style="list-style-type: none"> <li>a) Redução no tempo de atendimento e custos, eliminação da redundância na demanda de exames;</li> <li>b) Desterritorialização;</li> <li>c) Possibilidades de reconstrução histórica e completa dos casos acerca dos pacientes, registros médicos, tratamentos, laudos;</li> <li>d) Contribuição para a pesquisa;</li> <li>e) Fim do problema de compreensão dos hieróglifos da equipe de saúde;</li> <li>f) Facilidade na organização e no acesso às informações;</li> </ul>	<ul style="list-style-type: none"> <li>a) Manutenção dos prontuários em papel para fins jurídicos, em virtude da indefinição legal dos documentos eletrônicos;</li> <li>b) Necessidade de grande investimento em hardware, software e treinamento;</li> <li>c) Resistência a mudanças;</li> <li>d) Demora na sua implantação,</li> <li>e) Falhas na tecnologia,</li> <li>f) Falhas no sistema de fornecimento de energia elétrica.</li> </ul>

Fonte: Bentes Pinto (2006, p. 39).

Ao analisar as vantagens do prontuário eletrônico do paciente, vemos que os benefícios são muitos, mas mesmo assim alguns doutrinadores destacam que pelas desvantagens do prontuário eletrônico deve-se manter por um determinado tempo os prontuários em papel para fins legais.

#### ***2.1.4 Normas técnicas para prontuários eletrônicos***

Quando se trabalha com prontuários eletrônicos deve-se ficar atento a algumas técnicas para o seu desenvolvimento, seu uso, guarda e manuseio. Segundo Portaldomedico (2002), há algumas regras que devem ser seguidas para evitar futuras complicações. O quadro 4 abaixo nos mostra essas regras:

**Quadro 4.** Normas técnicas para prontuários eletrônicos

Integridade da Informação e Qualidade do Serviço	O sistema de informações deverá manter a integridade da informação através do controle de vulnerabilidades, de métodos fortes de autenticação, do controle de acesso e métodos de processamento dos sistemas operacionais conforme a norma ISO/IEC 15408, para segurança dos processos de sistema.
Cópia de Segurança	Deverá ser feita cópia de segurança dos dados do prontuário pelo menos a cada 24 horas. Recomenda-se que o sistema de informação utilizado possua a funcionalidade de forçar a realização do processo de cópia de segurança diariamente. O procedimento de back-up deve seguir as recomendações da norma ISO/IEC 17799.
Bancos de Dados	Os dados do prontuário deverão ser armazenados em sistema que assegure, pelo menos, as seguintes características: Compartilhamento dos dados; Independência entre dados e programas; Mecanismos para garantir a integridade, controle de conformidade e validação dos dados; Controle da estrutura física e lógica; Linguagem para a definição e manipulação de dados (SQL – Standard Query Language); Funções de auditoria e recuperação dos dados.
Privacidade e Confidencialidade	Com o objetivo de garantir a privacidade, confidencialidade dos dados do paciente e o sigilo profissional, faz-se necessário que o sistema de informações possua mecanismos de acesso restrito e limitado a cada perfil de usuário, de acordo com a sua função no processo assistencial.
Autenticação	O sistema de informação deverá ser capaz de identificar cada usuário através de algum método de autenticação.
Auditoria	Auditoria – O sistema de informações deverá possuir registro (log) de eventos, conforme prevê a norma ISO/IEC 17799. Registro das exceções e de outros eventos de segurança relevantes devem ser mantidos por um período de tempo não inferior a 10 (dez) anos, para auxiliar em investigações futuras e na monitoração do controle de acesso.
Transmissão de Dados	Para a transmissão remota de dados identificados do prontuário, os sistemas deverão possuir um certificado digital de aplicação única emitido por uma AC (Autoridade Certificadora) credenciada pelo ITI responsável pela AC Raiz da estrutura do ICP-Brasil, a fim de garantir a identidade do sistema.
Certificação do software	A verificação do atendimento destas normas poderá ser feita através de processo de certificação do software junto ao CFM, conforme especificado a seguir.
Digitalização de prontuários	Os arquivos digitais oriundos da digitalização do prontuário médico deverão ser controlados por módulo do sistema especializado que possua as seguintes características. Mecanismo próprio de captura de imagem em preto e branco e colorida independente do equipamento scanner

Fonte: [http://www.portalmedico.org.br/resolucoes/cfm/2002/1639\\_2002.htm](http://www.portalmedico.org.br/resolucoes/cfm/2002/1639_2002.htm)

### ***2.1.5 Prontuário eletrônico da atenção básica no Brasil***

No Brasil a implantação de prontuário eletrônico ainda está muito atrás de outros países. O CFM incentiva o uso dos prontuários eletrônicos, pois considera mais seguro do que o prontuário em papel, e também considera que seu uso pode propiciar o compartilhamento das informações do paciente, que possibilita a continuidade do seu tratamento (SBIS, 2012).

Atualmente o maior prontuário eletrônico no Brasil é o do SUS (Sistema Único de Saúde), que atende uma população de 15 milhões de pacientes, o ministério da saúde investiu em 2015 R\$ 91.2 milhões na aquisição de computadores, e entregou mais de 25 mil computadores para as unidades básicas de saúde espalhadas por todos os estados, com a missão de armazenar as informações do histórico do paciente (BRASIL, 2015).

O software de prontuário eletrônico do SUS (e-SUS AB), segundo ministro da saúde Arthur Chioro, representa agilidade e organização tanto para o usuário dos serviços públicos de saúde quanto para os profissionais de saúde (BRASIL, 2015).

Todos os estados estão conectados em rede pelo sistema QualiSUS-Rede, todas suas informações são encaminhadas diariamente para o servidor do ministério da saúde. Com a nova versão do prontuário eletrônico do SUS, podem ser feitos lançamentos direto no sistema, que antes eram separados ou eram feitos manualmente como: acompanhamento de pré-natal com informações importantes sobre a mulher e seu bebê. Na saúde bucal, os dentistas podem cadastrar as informações, e acessar pelo sistema próprio dos dentistas, que é odontograma eletrônico, e os gestores municipais agora podem acompanhar o trabalho de seus funcionários e gestores das unidades e imprimir relatórios individuais, pois a produção de cada um está registrado separadamente (BRASIL, 2015).

O que mais se aproxima da implementação de um sistema eletrônico de registro nacional público para pacientes no Brasil é o Cartão Nacional de Saúde, regulamentado pelo ministério da saúde em 2011. Esse cartão tem como objetivo a vinculação do que está ocorrendo ao paciente dentro do SUS, ao profissional que realizou e a unidade de saúde onde foi realizado (DATASUS, 2016).

O Datasus (Departamento De Informática do SUS), qual é vinculado ao Ministério da Saúde, disponibiliza diversos sistemas para gestão de serviços do SUS. Existem alguns sistemas que tem acesso ao Ministério da Saúde e dão suporte aos serviços de saúde do SUS, que são sistemas particulares contratados por alguns municípios (DATASUS, 2016).

## **2.2 Ferramentas a serem usadas no projeto**

### ***2.2.1 Plataforma***

O sistema foi desenvolvido para sistema operacional Windows 32 bits, pois assim é compatível com qualquer de suas versões e compatível para ser usado como cliente e servidor.

### ***2.2.1.1 Ferramenta de desenvolvimento***

Para o desenvolvimento do sistema foi usada a linguagem *Object Pascal* (Pascal Orientado a Objeto), essa linguagem é muito utilizada para desenvolvimento de softwares em todo mundo e por grandes empresas. Veremos a seguir um breve apanhado sobre essa linguagem.

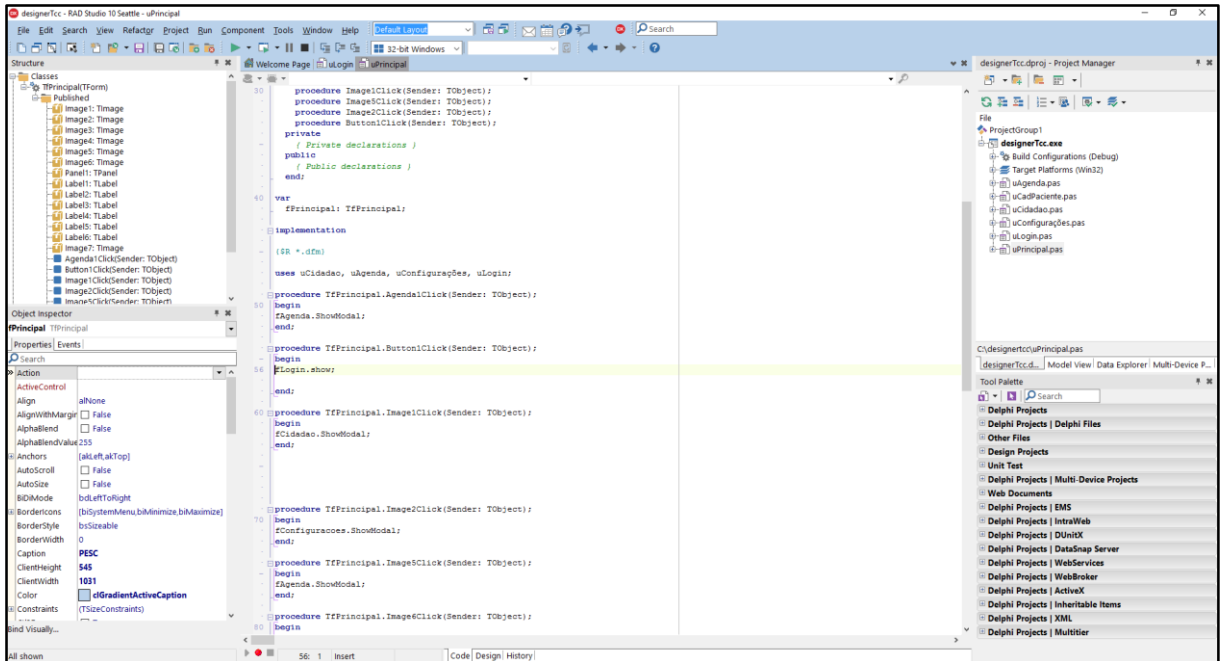
### ***2.2.1.2 Object Pascal***

*Object Pascal* é um seguimento de linguagens derivadas do pascal, que podem ser implementadas com orientação a objetos. A ferramenta mais usada para o desenvolvimento de softwares em *Object Pascal* é o IDE Embarcadero Delphi.

Seus programas são constituídos por módulos de códigos fontes chamados de *units*, as *units* ficam armazenadas em seus próprios arquivos com extensão .dcu e são compiladas separadamente. O código fonte incluído o programa principal fica todo armazenado em arquivos com extensão .pas (MANGUAL, 2014).

### ***2.2.1.3 IDE Delphi***

Para criação do sistema foi usado como ferramenta de desenvolvimento o IDE Delphi Embarcadero Versão 10 Seattle, a qual é muito usada no mundo todo para desenvolvimento em linguagem de programação Object Pascal, a figura 1 nos mostra a tela do Delphi Embarcadero versão gratuita, usado no desenvolvimento do projeto.



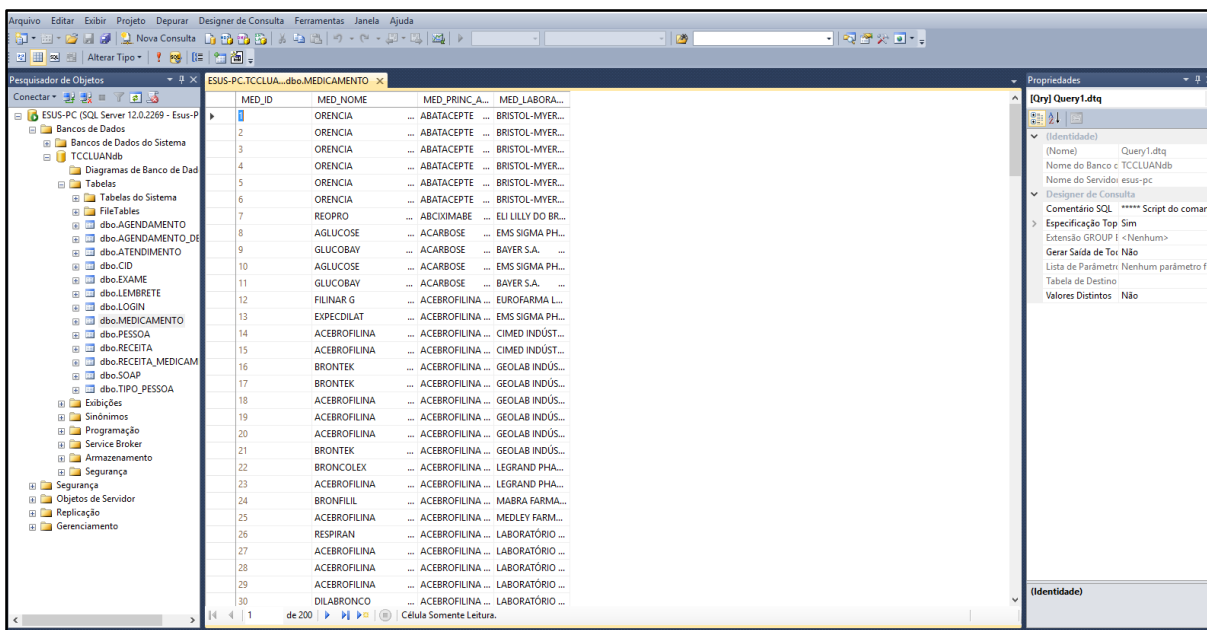
Fonte: Adaptado de Delphi Embarcader Seattle.

#### 2.2.1.4 Banco de dados

SQL (*Structured Query Language*) ou linguagem de consulta estruturada, foi desenvolvido com o propósito de ter um sistema padronizado para acessar e manipular dados em um banco de dados relacional. Mesmo a IBM (*International Business Machines*) sendo a pioneira no desenvolvimento da teoria de um sistema nesse sentido, a *Oracle Comporation* foi quem primeiro comercializou essa tecnologia. Com o passar do tempo outras linguagens permitiram o acesso e manipulação de dados relacionais, mas pelo fato de não serem uma linguagem de fácil entendimento não foram tão aceitos universalmente como SQL (KEVIN; KLINE, 2010).

Para o armazenamento e manipulação de dados no sistema, foi utilizado o banco de dados SQLServer Management Studio da Microsoft (Figura 2), que é uma ferramenta gratuita e de ótima qualidade. O SQLServer Management Studio permite acessar e gerenciar o banco de dados através de interface gráfica e também de instruções via código.

**Figura 2:** Tela banco de dados SQLServer Management Studio



Fonte: Adaptado de SQLServer Management Studio 2014.

### 2.2.1.5 Leitor de código de barras USB

Para leitura de informações será usado o leitor de código de barras BR-400 da empresa bematech (Figura 3), é o leitor mais preciso da categoria, fácil de ser utilizado e possui uma leitura rápida devida a ótima resolução e profundidade de leitura. O leitor BR-400 é compatível com sistemas operacionais como Windows, Linux e Android e não necessita de nenhuma mídia de instalação, para utiliza-lo basta conectar a USB e irá reconhecer automaticamente (BEMATECH, 2016).

Figura 3: Leitor código de barras



Fonte: <http://www.bematech.com.br/equipamentos/produto/br-400#>

## 2.3 Código de barras

Codificar um sinal elétrico em informações vem de tempos atrás, onde tínhamos o código Morse, onde sons longos e curtos, eram associados a um tipo de símbolo com informações.

### 2.3.1 Como funciona o código de barras

O código de barras é uma forma do computador ler uma informação numérica, que pode estar presente em produtos, envelopes e documentos. Seu funcionamento consiste em converter a informação que foi impressa, em sinais elétricos, que traduz a informação ali contida em forma de letras ou números, baseado em um laser que emite a luz, e um sistema de detecção que conta com ajuda de um componente eletrônico chamado de fotodetector, que vai decodificar o que ali está contido em cada código (GS1, 2016).

Cada código é único e inalterado, depois que se é criado nunca mais esse código poderá ser alterado, isso para nunca dar conflito na hora da leitura (GS1, 2016).



Um exemplo de composição do código de barras podemos observar na Figura 4, que nos mostra como funciona o código de barras utilizado em produtos que consumimos no dia-a-dia.

**Figura 4:** Exemplo de código de barras



Fonte: [http://www.gb-print.com/blog/wp-content/uploads/2012/06/codigobarrasgs1\\_ean13.png](http://www.gb-print.com/blog/wp-content/uploads/2012/06/codigobarrasgs1_ean13.png)

O criador do código de barras foi Norman Joseph Woodland, nascido em 1921 Atlantic City, Nova Jersey, Joseph estudou engenharia mecânica na Universidade de Drexel da Filadélfia. Criado em 1952 o código de barras só teve um padrão definido na década de 70, o qual era um padrão numérico para identificar produtos, primeiramente utilizado nos Estados Unidos e Canadá (INFOESCOLA, 2016).

Em 14 de Dezembro de 2012 aos 91 anos morreu Joseph, com diagnóstico de doença de Alzheimer, onde morava em um lar para idosos em Nova Jersey (EXAME, 2012).

## 2.4 O município de Anita Garibaldi

### 2.4.1 Breve histórico sobre o município de Anita Garibaldi

Antiga colônia Hercílio Luz, hoje chamada de Anita Garibaldi, que na época era um distrito de Lages, ganhou este nome em 1930 para homenagear à catarinense ilustre, e uma das maiores heroínas brasileiras a qual teve destaque mundialmente, e que em uma de suas viagens teve passagem pela cidade. Anita Garibaldi foi conhecida mundialmente como a Heroína dos

Dois Mundos, por sua bravura tanto no Brasil como na Itália. Em 1961 após trinta anos de luta do povo, o município de Anita Garibaldi conseguiu sua emancipação política, ficando assim independente do município de Lages. Situa-se geograficamente no Planalto Serrano, com uma área total de 637 Km<sup>2</sup>, possui atualmente uma população de 7881 habitantes, seu principal setor é o primário com 75% da renda gerada, seguido do setor secundário com 15% e 10% o setor terciário. A educação do município constitui-se dos cursos de pré-escolar, ensino fundamental e ensino médio (MARTELLO, 1996).

Hoje Anita Garibaldi é conhecida como cidade dos lagos, por ser uma cidade rodeada por 5 usinas hidrelétricas, que ajudaram e ainda contribuem muito para o desenvolvimento do município, através de projetos sociais, culturais e apoio as instituições.

#### ***2.4.2 A saúde no município de Anita Garibaldi***

A cidade de Anita Garibaldi conta com 3 unidades básicas de saúde e um hospital municipal quais veremos a seguir um pouco de como é o atendimento, como funcionam os prontuários e quais condições se encontram.

#### ***2.4.3 Secretaria municipal de saúde de Anita Garibaldi***

Segundo Secretário da Saúde Antônio Figueiró, a secretaria Municipal de Saúde de Anita Garibaldi SC, órgão da administração direta da Prefeitura Municipal, tem por missão dar uma boa estrutura ao atendimento de saúde no município, sempre buscando cumprir os princípios do SUS, quais princípios que objetivam atender a todos os habitantes do município de Anita Garibaldi. Segundo Antônio, o prontuário eletrônico será de muito bom aproveitamento nas unidades, pois permitirá tanto ao médico quanto ao enfermeiro acessarem as informações relativas aos pacientes de um modo fácil e rápido sem que perca tempo e atrapalhe o atendimento, e também será de grande apoio na hora dos agendamentos diários.

Nas UBS o atendimento é feito diariamente de segunda a sexta-feira, as fichas são distribuídas a partir das 8 horas da manhã pela ordem de chegada, mas para o atendimento médico é preciso passar por uma triagem, aonde é feito a medição da pressão cardíaca do paciente, medido sua altura e calcular o seu peso, para que esses dados sejam encaminhados junto ao prontuário para o atendimento médico.

Segundo Rodrigo Gehrke diretor de encaminhamentos externos da UBS, diariamente são encaminhadas 7 consultas de nosso município para o consórcio, o consórcio foi criado na

região da amures para que os municípios que não tem estrutura ou médicos especialistas para realizar algumas consultas e exames, possam encaminhar seus pacientes, exceto pacientes com caso de emergência, esses são encaminhados diretamente para as emergências dos hospitais.

As consultas de urgência que devem ser direcionadas ao consórcio são casos que o clínico geral não teve êxito no tratamento ou que ele mesmo julga que não tem qualificação para tratar aquele caso e precisa ser encaminhado a um especialista. As quantidades de encaminhamentos são feitas conforme a disponibilidade de investimento do município.

A secretaria municipal de saúde de Anita Garibaldi conta com médicos, dentistas, enfermeiros, auxiliares de enfermagem, psicólogos, assistente social e fisioterapeutas, todos atendendo pelo SUS, onde o atendimento não tem custo algum para o paciente. Duas unidades estão localizadas na cidade e uma no interior, as comunidades do interior também recebem uma vez por mês o atendimento do médico e dentista na unidade móvel, que é feito em um micro-ônibus equipado com dois consultórios para as consultas.

#### ***2.4.4 Como funcionavam os prontuários na UBS de Anita Garibaldi***

Antes da criação do software na UBS, os prontuários eram todos armazenados em pastas em um armário, na hora da distribuição das fichas para consulta, é pedido para que cada paciente ao buscar atendimento na unidade, tenha em mãos cartão do sus e o número do prontuário para que seja localizado seu histórico, mas nem sempre isso acontece, a maioria da população não apresenta o número de seu prontuário, ou as vezes informa, mas na hora de procurá-lo não corresponde ao que está na gaveta.

A numeração dos prontuários é dividido por região, cada região tem sua ACS (Agente Comunitário De Saúde), qual é responsável por organizar esses prontuários conforme os moradores daquela região. Quando um morador muda de região, a agente responsável vai até a unidade, retira aquele paciente e deixa em aberto aquela numeração, quando um novo morador se muda para a área é feito um novo cadastro, mas muitas vezes usa-se o número de prontuário que está em aberto. Essa numeração por não ser continua, causa muitos conflitos de números de prontuários, pois cada região começa a numeração do 1(um), em vez de ter uma sequência para o município todo. Muitos prontuários já foram perdidos também, devido a alagamentos, mudanças de local, reformas e outros fatores.

#### ***2.4.5 Hospital municipal Frei Rogério***

Segundo Ricardo Bertuol Dacoregio responsável técnico e gerente da enfermagem, hoje o Hospital Frei Rogério é administrado por uma associação beneficente, conta com uma boa estrutura e boas condições de atendimento, tem vinte e nove leitos disponíveis para internamentos, um centro cirúrgico com obstétrico, sala cirúrgica, setor de raio X e sala de parto.

O atendimento é feito vinte e quatro horas por dia, tendo atendimento de emergência e urgência, atende pelo SUS, particular e com convênios. Os médicos que são três, todos fazem plantão, trabalham com sobreaviso, e são cedidos pela prefeitura municipal, também cada médico tem uma equipe de enfermagem que auxilia nos procedimentos necessários.

#### ***2.4.6 Prontuários no hospital Frei Rogério***

Os prontuários no Hospital Frei Rogério são todos feitos manualmente e separados individualmente em folhas e armazenadas em um setor separado, ainda não possui um sistema que possa armazenar e monitorar esses dados. Segundo Ricardo, o risco de perder informações dos pacientes é grande, por mais cuidado que possa ter ao trabalhar com esses papéis mesmo assim fica vulnerável a outros fatores.

### **3 METODOLOGIA**

Neste item veremos a metodologia de pesquisa empregada, para o desenvolvimento do projeto.

#### **3.1 Documentação**

Em pesquisa bibliográfica, a documentação é a coleção de textos com objetivo de explicar, ou para demonstrar o problema escolhido pelo pesquisador. Para iniciar este projeto, foi elaborado uma pesquisa prévia, na busca de materiais que pudessem contribuir para a resolução do problema em questão, de forma criteriosa e seletiva (RUIZ, 2002).

#### **3.2 Natureza da pesquisa**

O presente trabalho apresenta suas informações por meio de pesquisa de caráter exploratório e descritivo.

Uma pesquisa exploratória consiste em familiarizar-se com um determinado assunto, o qual ainda não se tenha muito conhecimento e seja pouco explorado. Ao término de uma pesquisa exploratória, deve-se estar a par do assunto estudado, e apto a construir hipóteses. Por ser de modo exploratório o resultado da pesquisa depende muito da intuição do explorador (GIL, 2008).

A pesquisa descritiva tem o objetivo de descrever características de um certo assunto já conhecido, e proporcionar novas visões sobre a realidade nela exposta. Ao final da pesquisa descritiva, o pesquisador terá reunido e analisado muitas informações sobre o assunto (GIL, 2008).

#### **3.3 Tipo da pesquisa**

A pesquisa aqui se define como bibliográfica e de campo, por ser utilizada na busca de justificar os objetivos e inovações descritos no projeto.

A pesquisa bibliográfica abrange toda bibliografia disponível que tenha relação ao tema estudado, estando contida em diversos meios, desde publicações, boletins, revistas, livros, jornais, monografias, pesquisas, teses etc., até meios de comunicações orais como, radio, filme,

televisão, gravação etc. Seu objetivo é que o pesquisador tenha contato direto com todo material que tenha sobre um determinado assunto (MARCONI; LAKATOS, 2007).

Pesquisa de campo tem como objetivo de seu uso, conseguir informações e/ou conhecimentos sobre um determinado problema, o qual se procura uma solução, ou de uma hipótese que se deseja comprovar, ou ainda descobrir novos fenômenos. Consiste em observar os fatos e fenômenos que ocorrem espontaneamente. (MARCONI; LAKATOS, 2003).

Segundo (TRUJILLO, 1982, pág. 229) “A pesquisa de campo não deve ser confundida com a simples coleta de dados, exige contar com controles adequados e com objetivos preestabelecidos que discriminam suficientemente o que deve ser coletado”.

### **3.4 Técnicas de pesquisa**

Para realização de uma pesquisa é necessário fazer uma mistura entre os dados, as evidências, as informações e o conhecimento teórico adquirido sobre um determinado assunto, deve-se construir uma porção do saber. O conhecimento acumulado não é só resultado das atividades investigativas do pesquisador, mas é também uma continuação do que já foi elaborado e sistematizado por quem já trabalhou o assunto anteriormente (LUDKE; ANDRÉ, 1986).

### **3.5 Coleta de dados**

Para a coleta das informações sobre a viabilidade e tópicos importantes a serem abordados nesse projeto, foi aplicado na pesquisa a técnica de entrevista, a qual foi feita com profissionais da saúde, das UBS e do hospital municipal de Anita Garibaldi.

Em entrevista com estes profissionais, tive a oportunidade de definir quais as melhores práticas a serem tomadas para a realização do projeto, além de receber opiniões para o desenvolvimento. Também foram realizadas reuniões para discutir os planos sobre a implantação do sistema, nos quais, informações obtidas através das entrevistas foram transcritas no item 2.5.3 deste projeto.

Uma entrevista é uma conversa profissional entre duas pessoas, com objetivo de uma delas obter informações sobre um determinado assunto. É um procedimento utilizado na investigação social, para levantamento de dados ou para diagnosticar e tratar algum problema social. (MARCONI; LAKATOS, 2007).

Para (GOODE; HATT, 1969, pág. 237), a entrevista “Consiste no desenvolvimento de precisão, focalização, fidedignidade e validade de certo ato social como a conversação”.

### **3.6 Limitações da pesquisa**

No que se refere a pesquisa exploratória, assim como também a descritiva, pelo fato do tema não possuir um grande acervo de material para estudo, então a pesquisa limitou-se um pouco em artigos, sites e alguns livros. Para reunir informações e criar maior conhecimento sobre o assunto, foi aplicado uma coleta de dados, que foi a principal fonte para se familiarizar com o tema. A coleta ocorreu através de entrevistas feitas com profissionais da área da saúde, que também teve algumas limitações pois a entrevista não poderia ser qualquer hora, devido ao horário de atendimento dos entrevistados.

Referente ao projeto, ele se limita em alguns pontos para sua primeira versão. O sistema não poderá ser acessado em rede por cliente e servidor, devido as condições da rede local, que ainda passa por problemas internos. O *software* será instalado em apenas um computador na UBS, portanto só um usuário de cada vez, poderá utilizar o sistema nesta primeira versão.

## 4 PROJETO

### 4.1 Hardware

O desenvolvimento do *software* deste projeto foi em duas etapas, e em *hardwares* diferentes.

Na primeira etapa foi feito o desenvolvimento do *software*, em uma máquina pessoal com sistema operacional Windows 7 Ultimate 64 bits, e que possui os seguintes *hardwares*:

- a) Processador Intel Core I5;
- b) Memória RAM de 8GB;
- c) Placa de Vídeo dedicada 2GB;
- d) HD Armazenamento de 1TB;
- e) Leitor código de barras USB.

Considerando que nesta etapa o sistema foi desenvolvido e utilizado apenas para testes, a sua execução ocupou apenas uma parcela mínima do *hardware* utilizado.

Na segunda etapa, após terem sido feitos os testes necessário no sistema e o mesmo estar apto a ser usado, foi feita a implantação do sistema na UBS. O sistema operacional que foi utilizado no local é o Windows 10 32 bits, com os seguintes Hardwares:

- a) Processador Intel Core I3;
- b) Memória RAM de 6GB;
- c) HD Armazenamento de 1TB;
- d) Leitor código de barras USB.

### 4.1 Requisitos do sistema

Todo sistema independente do seu tamanho não funciona sozinho, depende de outros sistemas e hardwares que trabalhem em conjunto com ele. Para que isso ocorra sem problemas é necessário ter alguns requisitos tanto em software quanto em hardware.

Para instalação e utilização do PESC os requisitos mínimos recomendados são:

- a) Sistema operacional Windows 7 32/64 bits
- b) SQL server 2014
- c) Memória RAM 1 GB



- d) Processador 2,0 GHz
- e) 8 GB livre no disco

## **4.2 Ferramentas do projeto**

Para o desenvolvimento do projeto foi utilizado a plataforma de desenvolvimento Object Pascal, descrita no item 2.3.1.1 deste trabalho. O sistema foi desenvolvido com a IDE Delphi Embarcadero 10 Seattle, e suas informações armazenadas em um banco de dados com a linguagem SQL, qual foi escolhido o banco de dados SQLServer Management Studio e para leitura de código de barras foi utilizado o leitor USB BR-400 da empresa bematech.

## **4.3 Modelagem UML**

UML (Unified Modeling Language) é uma linguagem que padroniza como se especifica, constrói, visualiza e documenta sistemas orientados a objeto. Facilita o desenvolvimento de um software, permite integrar a organização e os elementos tecnológicos, que estarão presentes na construção do sistema, ajudando a dominar a complexidade das necessidades encontradas e definir os processos e fluxos informativos (NUNES; O'NEILL, 2016).

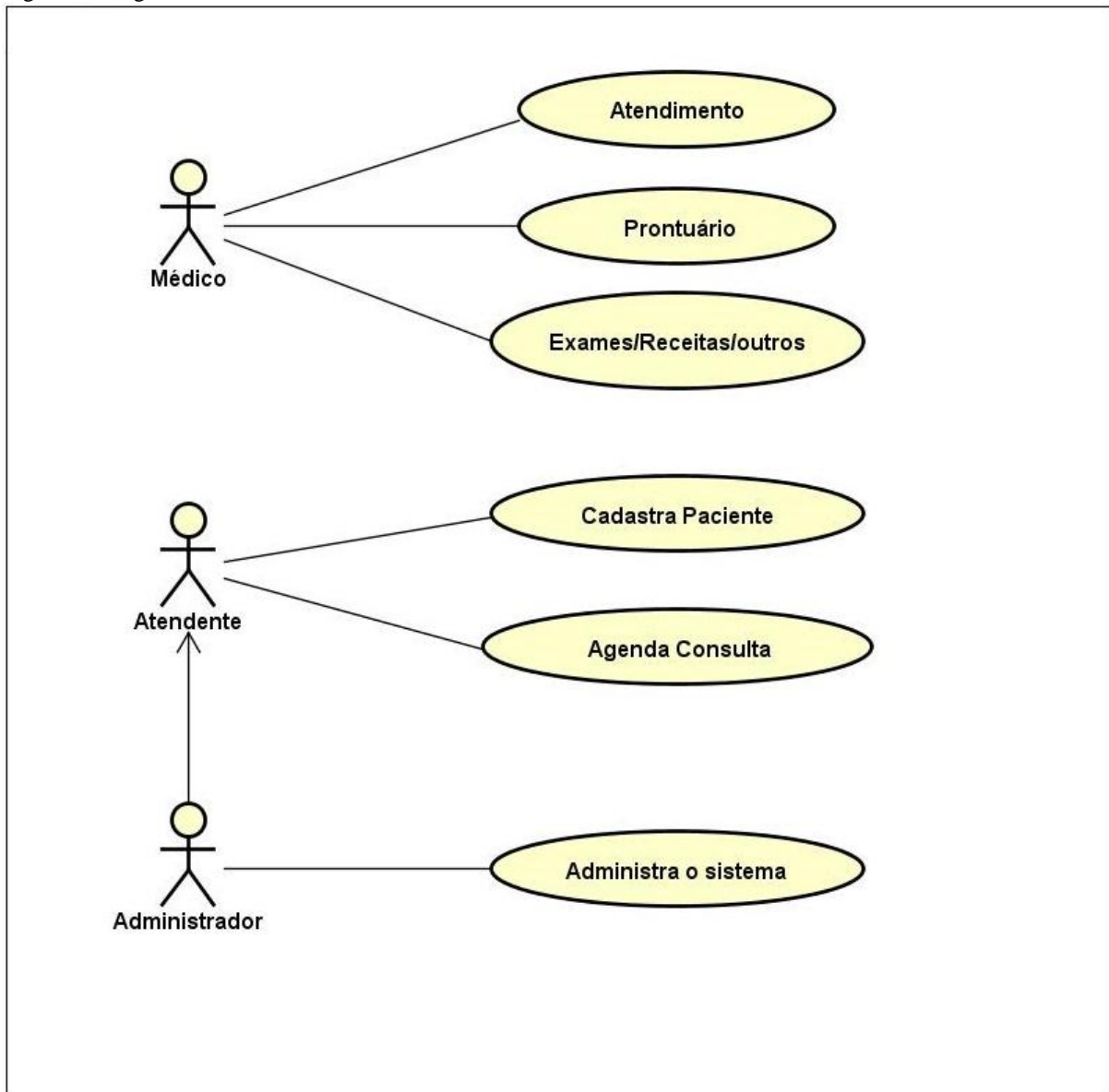
Em UML um modelo é constituído por um conjunto de diagramas, esses representam os aspectos complementares de um sistema. Cada diagrama utiliza símbolos que representam elementos que serão modelados e também linhas que ligam esses elementos (NUNES; O'NEILL, 2016).

### ***4.3.1 Diagrama de caso de uso***

É usado para identificar os limites do sistema e descrever os serviços que serão disponibilizados para cada um dos diversos usuários do *software*. Se o levantamento de requisitos for bem trabalhado e que se consiga identificar bem o que o sistema exige, o resultado final será muito bom, e o sistema será muito útil pois estará de acordo com as necessidades do utilizador (NUNES; O'NEILL, 2016).

A seguir (Figura 5) é mostrado o diagrama de caso de uso que descreve as funções que cada usuário é autorizado a fazer dentro do sistema.

Figura 5: Diagrama de caso de uso



Arquivo do Autor

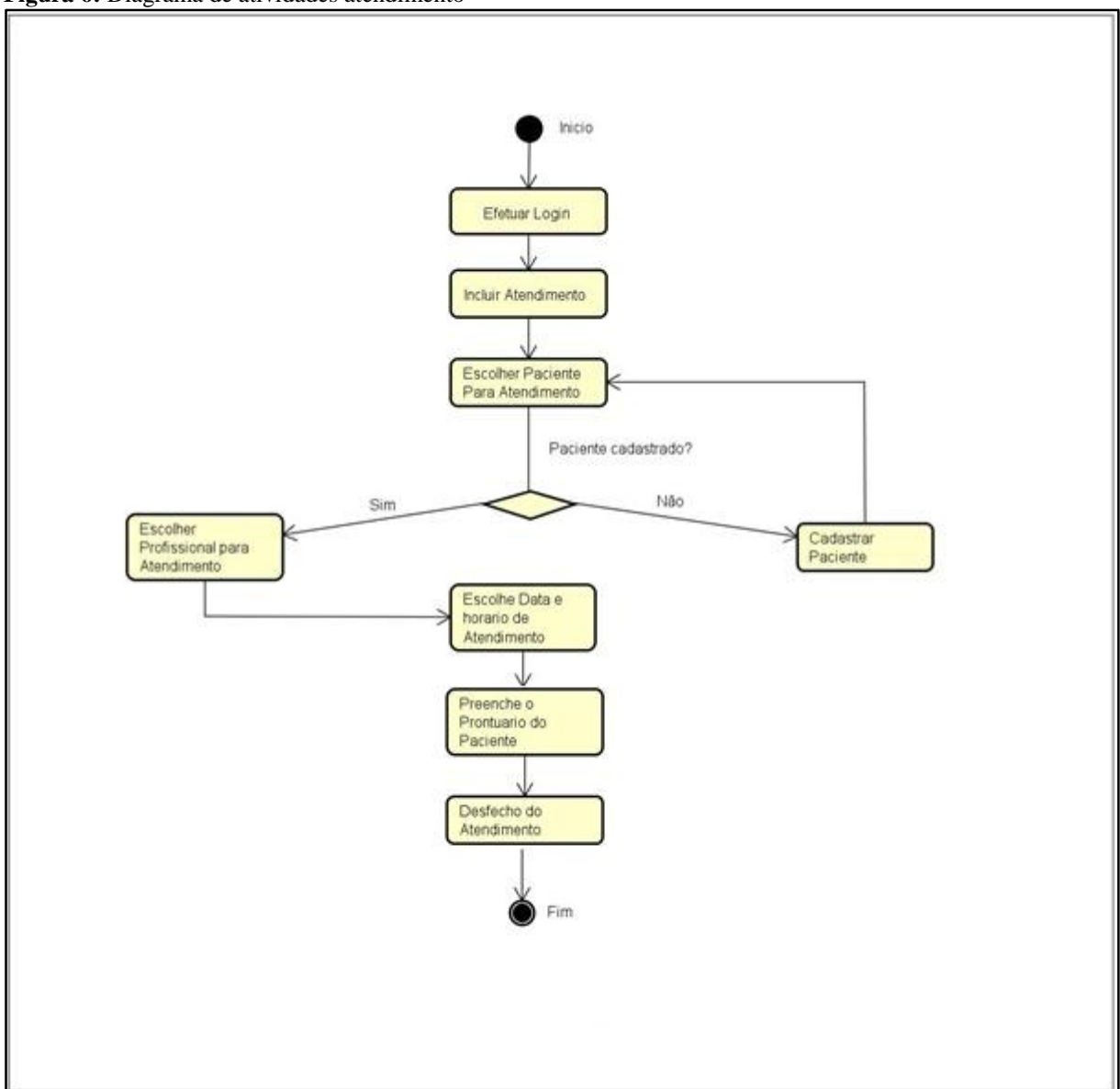
No diagrama anterior (Figura 5), o cenário é constituído por 3 (três) atores. O atendente que tem acesso as funções de cadastrar paciente e agendar atendimento, o médico tem acesso as funções de atendimento, preencher prontuário, solicitar exame, receitas e agenda e algumas configurações do sistema, o administrador esse tem acesso a quase todas as áreas do sistema, menos nos prontuários.

### 4.3.2 Diagrama de atividades

O diagrama de atividade é constituído com elementos de modelação simples, mas, no entanto, é eficaz para descrever fluxos de trabalho e detalhar operações em uma classe, incluindo processos paralelos. Um diagrama de atividades, também pode ser usado para descrever um fluxo mais amplo, que envolve diversos *use cases* (NUNES; O'NEILL, 2016).

A seguir veremos a figura 6, que contém o diagrama de atividades e apresenta o fluxo para o atendimento de um paciente.

**Figura 6:** Diagrama de atividades atendimento



Arquivo do Autor

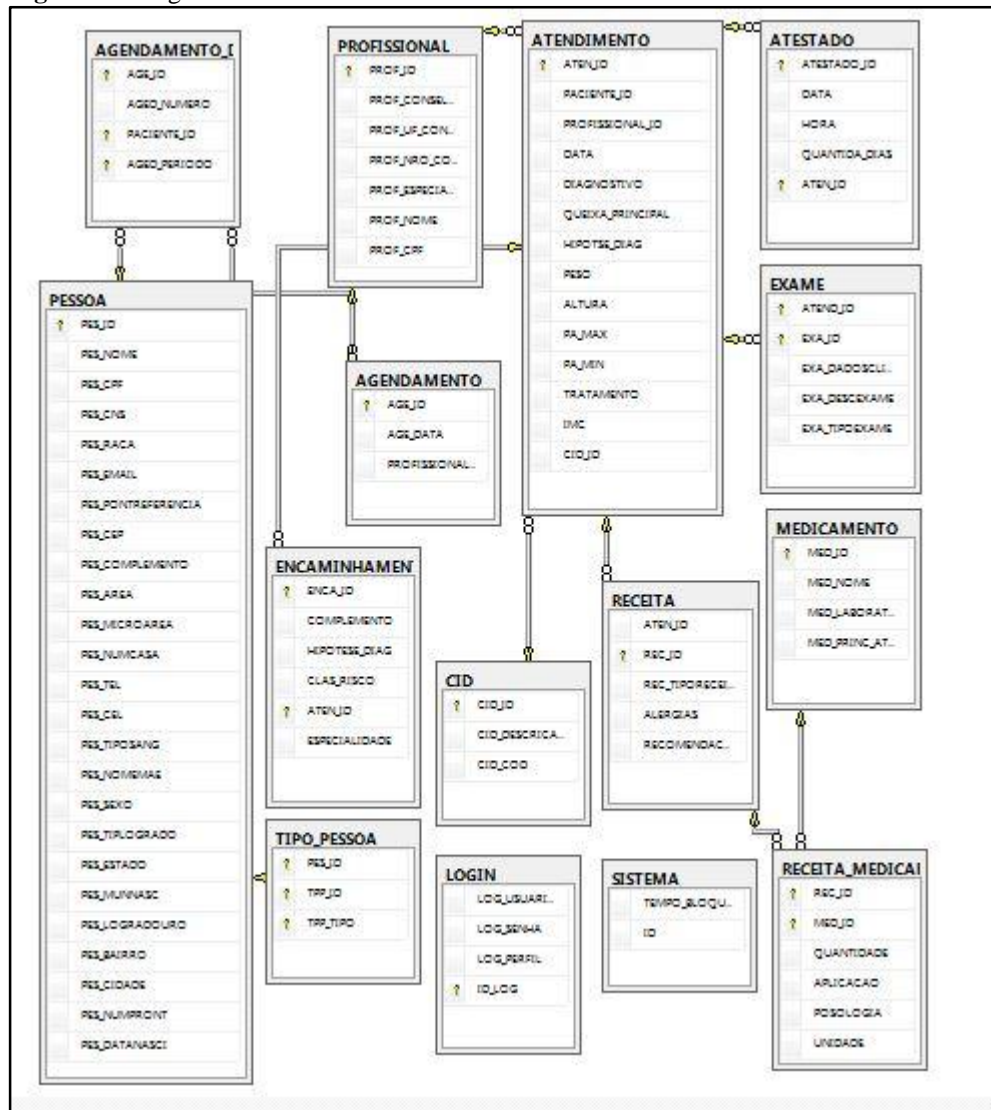
O paciente chega na UBS para pegar ficha de consulta, o atendente do balcão faz o *login* no sistema, vai na agenda do médico localiza o paciente, se já tem cadastro inclui na agenda se não tem cadastra e inclui. Após isso o paciente será chamado pelo médico que irá preencher seu prontuário, fazer o desfecho de emitir receitas, exames ou o que for necessário para aquele atendimento.

### ***4.3.3 Diagrama Entidade Relacionamento***

O diagrama DER (diagrama de entidade relacionamento) é a representação gráfica do modelo de entidade e relacionamento. Descreve a estrutura de um banco de dados, define os dados mantidos pelo sistema e o fluxo desses dados. É um modelo abstrato que tem o objetivo de descrever a maneira conceitual dos dados a serem utilizados, e é representado por entidades, relacionamentos e atributos (Londeix, 1995).

A diante pode-se observar o diagrama de estrutura de dados do sistema PESC (Figura 7), a maioria de suas tabelas são ligadas uma com as outras, ficando assim um banco de dados bem relacionado.

**Figura 7:** Diagrama de banco de dados do sistema



Arquivo do Autor

## 4.4 Telas do sistema

A seguir veremos a apresentação das principais telas que fazem parte do sistema.

### 4.4.1 Tela *login* de acesso ao sistema

A tela de *login* a seguir (Figura 8), é a primeira tela apresentada ao usuário depois do sistema instalado e configurado.

Figura 8: Tela login de acesso ao sistema

A imagem mostra a interface de login do sistema PESCS. No topo, há um banner com o logotipo PESCS (uma casa azul com uma cruz branca) e o nome 'PESCS' em letras azuis. Abaixo do banner, o título 'ACESSO AO SISTEMA' está centralizado em um fundo cinza claro. O formulário de login contém dois campos de entrada: 'Usuário' e 'Senha', ambos com caixas de texto brancas e bordas cinzas. Abaixo dos campos, há dois botões de ação: 'Cancelar' e 'Entrar', ambos em um fundo laranja com texto preto.

Arquivo do Autor

Nessa tela o usuário informará seus *login* de acesso e o sistema faz a pesquisa no banco de dados para verificar se os dados informados são compatíveis.

#### ***4.4.2 Tela principal***

A tela principal do sistema (Figura 9) é a aonde o usuário escolhe qual parte irá acessar no sistema, sendo que cada usuário tem um tipo de visualização por nível de acesso.

Figura 9: Tela login de acesso ao sistema



Arquivo do Autor

#### ***4.4.3 Tela de cadastro de cidadão***

Próxima tela (Figura 10) é a de cadastro do cidadão, nesta tela é feito o cadastro dos cidadãos que utilizam serviços na UBS, aonde posteriormente suas informações serão acessadas pelo profissional da saúde na hora do atendimento.

Figura 10: Tela de cadastro de cidadão

Identificação

Nome Completo  \* Data Nascimento 01/11/2016  \*

CPF  \* CNS  \* Nº Prontuário  \*

Nome Mãe  Tipo Sanguíneo

Sexo  \* Raça/Cor  \* Município Nascimento  \*

Localização

Área  \* Micro Área  \*

Tipo Logradouro  Logradouro  Bairro

Complemento  Ponto Referência  Nº Casa

Estado  Cidade  CEP

Telefone ( ) -  Celular ( ) -  E-mail

Arquivo do Autor

Na tela de cadastro do cidadão também é gerado e impresso o CMS (cartão municipal de saúde), que recebe um código de barras contendo o número do CNS.

#### ***4.4.4 Tela de agendamento***

Na tela de agendamento (Figura 11) é feito o lançamento dos pacientes na agenda do médico, e também é possível consultar agendamentos passados e agendar futuros. Não é possível agendar consultas para finais de semana ou data passada o sistema já traz bloqueado essa opção.



**Figura 11:** Tela de agendamento

Profissional: zaira

Período: Manhã

Imprimir Agenda

Seleção a Data: novembro de 2016

Hoje: 01/11/2016

Nº Ficha	Paciente	nº Prontuário	Área	Micro Área
1	MARCUS	0	0	0
2	jardel	0	0	0
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Remover Paciente

Cancelar

Arquivo do Autor

Para agilizar a localização do paciente e inclui-lo na agenda do médico, é usado o leitor de código de barras no CMS gerado no próprio sistema, nesta parte do sistema é possível determinar qual período será o atendimento, qual médico irá atender, a data de agendamento e imprimir a agenda do médico.

#### 4.4.5 Tela de atendimento

Tela de atendimento (Figura 12), será a parte do sistema aonde o médico irá lançar os registros dos atendimentos feitos para cada paciente.

Figura 12: Tela de cadastro de atendimento

The screenshot shows a web application window titled "PESC - [Atendimento]". The main header is "PESC - Prontuário Eletrônico da Saúde do Cidadão". Below this, the patient's name is displayed as "Paciente: JOÃO PEDRO DA SILVA MACHADO ALVAREZ". A navigation bar contains tabs for "PEP", "Receita", "Exames", "Atestado", "Encaminhamento", and "Histórico de atendimentos". The main content area is divided into four sections: "QP - Queixa Principal" (a large empty text box), "HDX/DX - Hipótese diagnóstico - Diagnóstico" (a large empty text box), "DX-Diagnóstico" (a large empty text box), and "TR- Tratamento" (a large empty text box). To the right of the "HDX/DX" section, there are input fields for "Peso" (Kg), "Altura" (Cm), "IMC", and "Pressão Arterial" (mmHg). Below the "DX-Diagnóstico" section is a "CID" dropdown menu. At the bottom right, there are two buttons: "Limpar" (with a red X icon) and "Confirmar" (with a document icon). The bottom left corner shows "Usuário Logado: zaira".

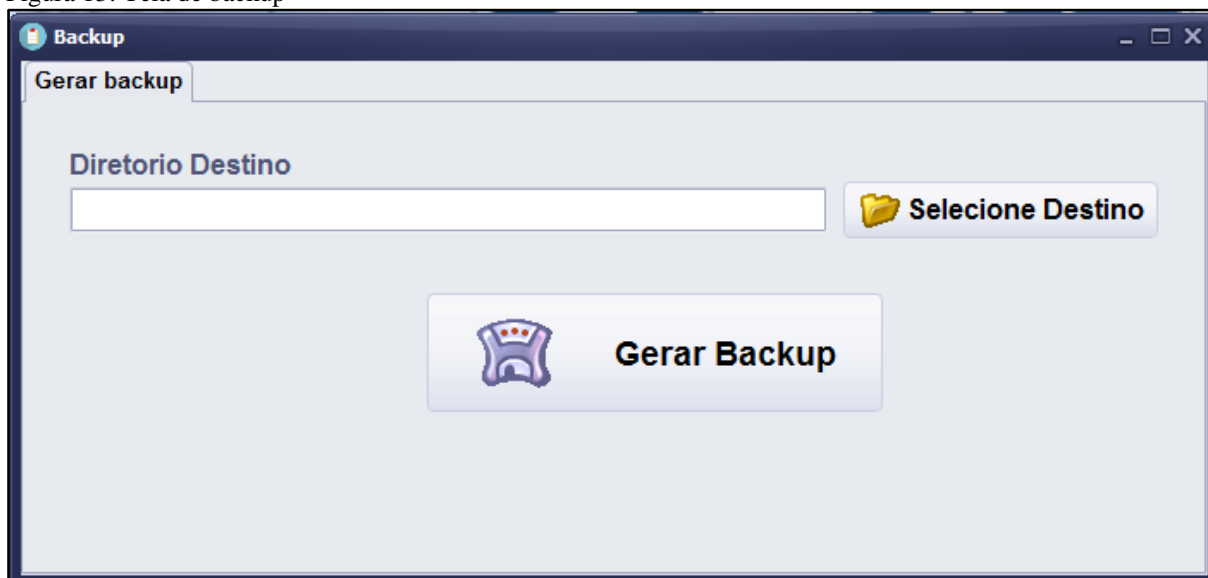
Arquivo do Autor

Nessa primeira versão do sistema, só será possível o lançamento de atendimento médico. O médico pode manipular o prontuário do paciente, dar receita, pedido de exame, encaminhamento, atestado e visualizar histórico de outros atendimentos já realizados.

#### 4.4.6 Tela de Backup

O backup do banco de dados do sistema é gerado nesta tela como podemos observar na figura 13 a seguir.

Figura 13: Tela de backup



Arquivo do Autor

O backup do sistema só pode ser acessado pelo administrador, e por medidas de segurança a restauração de backups e gerenciamento de banco de dados ficará por responsabilidade do desenvolvedor dar suporte.

## 5 TRABALHOS CORRELATOS

Nessa seção será apresentado um sistema que também faz parte da área de gestão de prontuários eletrônicos. Foi escolhido como exemplo, Saúde Fly da Betha Sistemas, existem vários outros *softwares* que também fazem parte desta área, porém, o escolhido é o que possui maior semelhança com o presente projeto, e é o mais usado em nossa região.

### 5.1 Saúde Fly

Desenvolvido pela empresa Betha Sistemas, o Saúde fly conta com seu funcionamento direto na nuvem, não necessita de um servidor local como banco de dados, tudo nele é feito online, o cliente só precisa de uma boa conexão com a internet e um computador para o acesso. Pelo fato de ser 100% online, o Saúde Fly permite que o acesso do usuário possa ser feito de qualquer dispositivo conectado à internet. Para reduzir o número de faltantes nas consultas, foi criado uma funcionalidade que o sistema envia mensagens de celular (SMS), automáticas, para lembrar os pacientes de suas consultas (BETHA,2016).

No quadro 5, veremos algumas características do sistema Saúde Fly:

Quadro 5. Características Saúde Fly

Integrado ao SUS	Possui integração com todos os sistemas informatizados do próprio SUS.
Agilidade no atendimento aos pacientes	Facilita a entrada e o andamento de serviços prestados a pacientes atendidos pelo SUS.
Visão informatizada da situação clínica dos pacientes	Agilidade na localização de informações, além de diagnósticos mais completos e pacientes melhor atendidos.
Software aberto	Permite que a implantação de funcionalidades seja feita com prontidão, e os ajustes implementados tornam-se automaticamente disponíveis a todos que usam o sistema.
Personalização de acordo com cada região	Permite a inclusão de funcionalidades que acompanham peculiaridades regionais.
Facilidade no acesso	Permite o registro de atendimentos também por conexão 3G.
Reduz o uso de papel	E ainda desburocratiza o acompanhamento de solicitações.
Acompanhamento dos medicamentos	Controle eficiente da retirada de remédios.
Relatórios e gráficos de pacientes	Facilita o total conhecimento da saúde da população.

Fonte: <http://www.betha.com.br/produto/saude-e-assistencia-social/saude-fly>

## 6 CRONOGRAMA

Para a realização do projeto proposto neste trabalho, seguiu-se o seguinte cronograma.

Quadro 6. Cronograma do TCC I

Atividades Realizadas	Mar.	Abr.	Mai.	Jun.	Jul.	Ago.	Set.	Out.	Nov.	Dez.
Idealização do projeto e levantamento de requisitos										
Pesquisa bibliográfica										
Estudo de técnicas e metodologia										
Especificação do projeto										
Pré-apresentação TCC										
Entrega Final TCC I										
Defesa e apresentação TCC I a banca avaliadora										
Adequação da proposta para o projeto										
Definição das funcionalidades do sistema										
Execução do projeto										
Testes										
Entrega Final TCC II										
Defesa e apresentação TCC II a banca avaliadora										

## **7 RESULTADOS**

Essa primeira versão de software, implantado gratuitamente na UBS Dr. Clóvis Cechin, auxiliou na gestão de prontuários eletrônicos e nos agendamentos da UBS, a qual não possuía condições para implantação de um software particular por motivos financeiros e estruturais.

Com a funcionalidade de código de barras, o agendamento de consultas se tornou mais prático e mais rápido, evitou filas, acúmulo de papeis em cima do balcão do atendimento, dificuldades na leitura das informações e transtornos causados quando o paciente não apresenta o número de seu prontuário.

Os profissionais que trabalham no local implantado, já estão utilizando o sistema e sem dificuldades, como era o esperado pelo projeto.

## REFERÊNCIAS

- BEMATECH. **Hardwares e Leitores: BR 400.** Disponível em: <<http://www.bematech.com.br/equipamentos/produto/br-400>>. Acesso em: 15 mar.2016.
- BETHA, S. **Saúde Fly.** Disponível em: <<http://www.betha.com.br/produto/saude-e-assistencia-social/saude-fly>>. Acesso em: 05 jun.2016.
- BRASIL, SUS: **Prontuário Eletrônico Beneficiará 15 Milhões De Pacientes.** Disponível em: < <http://www.brasil.gov.br/saude/2015/09/sus-prontuario-eletronico-beneficiara-15-milhoes-de-pacientes> >. Acesso em: 29 mar.2016.
- COSTA, C. G. A; Marques, A. **Implementação de um Prontuário Eletrônico do Paciente na Maternidade Escola Januário Cicco: Um Primeiro Passo.** Rio de Janeiro, PEP'99, 1999.
- COSTA, C. G. A. **Prontuário Eletrônico do Paciente: Legislação, Auditoria e Conectividade,** - 8º Congresso Latino Americano de Serviços de Saúde, 2003.
- COSTA, C. G. A. **Desenvolvimento e Avaliação Tecnológica de um Sistema de Prontuário Eletrônico do Paciente, baseado nos Paradigmas da World Wide Web e da Engenharia de Software.** Dissertação de Mestrado. Universidade Estadual de Campinas, 2001.
- DATASUS. **Histórico.** Disponível em: <<http://datasus.saude.gov.br/datasus>>. Acesso em: 21 abr.2016.
- DURAND, G. **Introdução Geral à Bioética – História, Conceitos e Instrumentos.** Editora Loyola, São Paulo, 1999.
- EXAME, **Inventor Do Código De Barras Morre Nos Eua,** Disponível em: < <http://exame.abril.com.br/tecnologia/noticias/inventor-do-codigo-de-barras-morre-nos-eua> >. Acesso em 20 abr.2016.
- GB, N.P. **Código De Barras EAN13.** Disponível em:<[http://www.gb-print.com/blog/wp-content/uploads/2012/06/codigobarrasgs1\\_ean13.png](http://www.gb-print.com/blog/wp-content/uploads/2012/06/codigobarrasgs1_ean13.png)>. Acesso em: 25 mar.2015
- GIL, A.C. **Como elaborar projetos de pesquisa.** 5. ed. São Paulo: Atlas, 2008.
- GOODE, W.J.; HATT, P.K. **Métodos em pesquisa social.** 2 ed. São Paulo. Ed. Nacional, 1969.
- GS1, **Entenda Como Funciona O Código De Barras.** Disponível em: < <https://www.gs1br.org/codigos-e-padres/entenda-como-funciona-o-codigo-de-barras>>. Acesso em: 02 abr.2016.
- INFOESCOLA, **Código De Barras.** Disponível em: <<http://www.infoescola.com/curiosidades/codigo-de-barras>>. Acesso em 20 abr.2016.
- KLINE, D; KLINE, K.E. **Sql: O Guia Essencial - Manual De Referencia Profissional.** 1.ed, 2010.

LONDEIX, A. **Modelo de Entidade e Relacionamento**, 2003.

LÜDKE, M; ANDRÉ, MARLI, E.D.A. **Pesquisa em educação: abordagens qualitativas**. São Paulo: EPU, 1986.

MANGUAL, D. **Object Pascal e Delphi Programas e Units**. 2014. Disponível em: <<http://slideplayer.com.br/slide/354018>>. Acesso em: 25 setembro. 2016.

MARCONI, M.A; LAKATOS, E.M. **Fundamentos de metodologia científica**. 6. ed. São Paulo: Atlas, 2007.

MARCONI, M. A.; LAKATOS, E.M. **Fundamentos de metodologia científica**. 5. ed. São Paulo: Atlas, 2003.

MARTELLO, **História De Anita Garibaldi**.1.ed.Anita Garibaldi: SC, 1996.

MARTINS, A.S. E; ZANARDO, J. S. **Sistema de Controle de Acesso para os Requisitos da Saúde**, Anais do IX Congresso Brasileiro de Informática em Saúde, novembro de 2004.

MASSAD, E; MARIN, H. F; AZEVEDO, R. S. **O prontuário eletrônico do paciente na assistência, informação e conhecimento médico**. São Paulo: H. de F. Marin, 2003.

MOTTA, G. H. M. B. **Um Modelo de Autorização Contextual para o Controle de Acesso ao Prontuário Eletrônico do Paciente em Ambientes Abertos e Distribuídos**. Tese de Doutorado, Escola Politécnica da USP, 2003. Disponível em: <[http://www.teses.usp.br/teses/disponiveis/3/3142/tde-05042004-152226/publico/tese\\_Gustavo\\_Motta](http://www.teses.usp.br/teses/disponiveis/3/3142/tde-05042004-152226/publico/tese_Gustavo_Motta)>. Acesso em 05 abr.2016.

MURPHY, G.F; HANKEN, M.A; WATERS, K.A. **Electronic Health Records: Changing the Vision**. Philadelphia: W.B. Saunders Company, 1999.

NUNES, M; O'NEILL; Henrique. **Fundamental de Uml**.2ed.Sao Paulo: edi.FCA,2016.

ONU- Conselho Econômico e Social. Technologies de l'information et de la communication pour le secteur de la sante. E/ECA/DISD/CODI.2/24. In: DEUXIÈME RÉUNION DU COMITÉ DE L'INFORMATION POUR LE DÉVELOPPEMENT, Addis-Abeba, Ethiopie 4-7 septembre 2001.

PINTO, V.B. Prontuário eletrônico do paciente: documento técnico de informação e comunicação do domínio da saúde. Encontros Bibli. **Revista Eletrônica de Biblioteconomia e Ciência da Informação**, Florianópolis, v. 11, n. 21, 2006.

POISSANT, L. et al. The impact of electronic health records on time efficiency of physicians and nurses: a systematic review. **Journal of the American Medical Informatics Association**, vol. 12 n° 5, 2005.



PORTALMEDICO. **Resolução Cfm.** Disponível em: <[http://www.portalmedico.org.br/resolucoes/cfm/2002/1639\\_2002.htm](http://www.portalmedico.org.br/resolucoes/cfm/2002/1639_2002.htm)>. Acesso em: 20 abr.2016

POSTGRESQL, C.B, **sobre o PostgreSQL.** Disponível em: <<https://www.postgresql.org.br/sobre>>. Acesso em 28 mar.2016.

RITCHEY, T. **Programando Java & JavaScript para Netscape 2.0.** São Paulo, p.18-27, Ed. Quark. 1996.

ROGER, F. F. H, GAUNT, P. N. **The need for security - a clinical view.** Int J Biomed Comput, v. 35, Suppl 1, p.194, 1994.

RUIZ, J.A. **Metodologia Científica:** Guia para eficiência nos estudos. 5. ed. São Paulo: Atlas, 2002.

SABATINI, R. M. E. Informatizando o consultório médico. **Revista Informática Médica**, v. 1, n. 4. Ago. 1999.

SBIS, **Cartilha sobre Prontuário Eletrônico: A certificação de sistemas de registro eletrônico de saúde do Conselho Federal de Medicina.** Disponível em: <[http://www.sbis.org.br/certificacao/Cartilha\\_SBIS\\_CFM\\_Prontuario\\_Eletronico\\_fev\\_2012.pdf](http://www.sbis.org.br/certificacao/Cartilha_SBIS_CFM_Prontuario_Eletronico_fev_2012.pdf)>. Acesso em: 16 abr.2016.

SILVEIRA, P.G.; LOPES, S.G.M. **Introdução À Arquitetura E Design De Software: Uma visão sobre a Plataforma Java.** ed. Rio de Janeiro: RJ, 2003.

TRUJILLO, F. A. **Metodologia da ciência.** 3. Ed. Rio de Janeiro: Kennedy, 1974.  
V, B. J. H; MUSEN, M.A. **Handbook of medical informatics.** Netherlands: Springer-Verlag. 1997.

WECHSLER, R et al. **A informática no consultório médico.** Jornal de Pediatria, Rio de Janeiro, p. 3-12. Maio 2003. Disponível em:< [http://www.scielo.br/scielo.php?pid=S0021-755720033000700002&script=sci\\_abstract&tlng=pt](http://www.scielo.br/scielo.php?pid=S0021-755720033000700002&script=sci_abstract&tlng=pt)>. Acesso em: 11 mar. 2016.

## APÊNDICE A – Configuração geral do sistema

```

unit ConfiguracaoGeral;
interface
uses
FireDAC.Comp.Client, FireDAC.Comp.UI, FireDAC.UI.Intf, uDefinicoes,
FireDAC.VCLUI.Login;
const
LICENCE_KEY = 'M36RB-R4AWH-KXEN2-29ENT-WBXQA';
type
TConfiguracaoGeral = class
strict private
class var FInstance : TConfiguracaoGeral;
private
FDGUIxLoginDialog1: TFDGUIxLoginDialog;
FDGUIxWaitCursor : TFDGUIxWaitCursor;
FUsuarioLogado_Codigo: Double;
FConnection: TFDCConnection;
FTipoUsuario: TTipoUsuario;
function RestaurarDatabase: Boolean;
procedure ConfigurarAcessoBancoDados;
class procedure ReleaseInstance();
public
class function GetInstance(): TConfiguracaoGeral;
property UsuarioLogado_Codigo: Double read FUsuarioLogado_Codigo write
FUsuarioLogado_Codigo;
property Connection: TFDCConnection read FConnection write FConnection;
property TipoUsuario: TTipoUsuario read FTipoUsuario write FTipoUsuario;
constructor Create;
destructor Destroy; override;
end;

implementation

uses
System.SysUtils, vcl.Dialogs;

{ TConfiguracaoGeral }

constructor TConfiguracaoGeral.Create;
begin
ConfigurarAcessoBancoDados;
end;

destructor TConfiguracaoGeral.Destroy;
begin
if FDGUIxWaitCursor <> nil then
FDGUIxWaitCursor.Free;

```

```

if FConnection <> nil then
  FConnection.Free;

inherited;
end;

procedure TConfiguracaoGeral.ConfigurarAcessoBancoDados;
var
  xFileName: String;
  xLICENCE_KEY: String;
begin
  FDGUIxWaitCursor := TFDGUIxWaitCursor.Create(nil);
  FDGUIxWaitCursor.ScreenCursor := gcrAppWait;

  FConnection := TFDConnection.Create(nil);
  FConnection.ConnectionName := 'MSSQLConn';
  FConnection.DriverName := 'MSSQL';

  xFileName := ExtractFilePath(ParamStr(0)) + 'DBconfig.ini';

  if FileExists(xFileName) then
  begin
    FConnection.Params.LoadFromFile(xFileName);
    FConnection.Params.Values['DriverID'] := 'MSSQL';

    if FConnection.Params.Values['Database'] = '' then
      FConnection.Params.Values['Database'] := 'PEScdb';

    FConnection.Open;
  end
  else
  begin
    xLICENCE_KEY := EmptyStr;
    repeat
      if not InputQuery('Chave de Instalação', 'Informe a Chave', xLICENCE_KEY) then
        Halt

    until LICENCE_KEY = Trim(xLICENCE_KEY);

    FDGUIxLoginDialog1 := TFDGUIxLoginDialog.Create(nil);
    try
      FDGUIxLoginDialog1.Provider := 'Forms';
      FDGUIxLoginDialog1.Caption := 'Configurando o primeiro acesso';
      FDGUIxLoginDialog1.VisibleItems.Add('Server');
      FDGUIxLoginDialog1.HistoryEnabled := True;
      FDGUIxLoginDialog1.HistoryWithPassword := True;
      FDGUIxLoginDialog1.HistoryStorage := hsFile;
      FDGUIxLoginDialog1.HistoryKey := xFileName;

```

```

FConnection.Params.Clear;
FConnection.Params.Values['Database'] := 'master';
FConnection.DriverName := 'MSSQL';

FConnection.LoginPrompt := True;
FConnection.LoginDialog := FDGUIxLoginDialog1;
try
FConnection.Connected := True;

if FConnection.Connected then
begin

if RestaurarDatabase then
begin
FConnection.Connected := False;
FConnection.Params.LoadFromFile(xFileName);
FConnection.Params.Values['DriverID'] := 'MSSQL';
FConnection.Params.Values['Database'] := 'PESCdb';
FConnection.LoginPrompt := False;
FConnection.Connected := True;
end;
end;

except
on E: Exception do
begin
ShowMessage('Não foi possível acessar o servidor de Banco de Dados. ');
Halt;
end;
end;
finally
FDGUIxLoginDialog1.Free;
end;
end;
end;

class function TConfiguracaoGeral.GetInstance: TConfiguracaoGeral;
begin
if not Assigned(Self.FInstance) then
self.FInstance := TConfiguracaoGeral.Create;

Result := Self.FInstance;
end;

class procedure TConfiguracaoGeral.ReleaseInstance;
begin
if Assigned(Self.FInstance) then
Self.FInstance.Free;
end;

```

```

function TConfiguracaoGeral.RestaurarDatabase: Boolean;
var
  qryRestore: TFDQuery;
  xPath: String;
begin
  Result := False;

  qryRestore := TFDQuery.Create(nil);
  try
    qryRestore.Connection := FConnection;
    try
      with qryRestore do
        begin
          SQL.Clear;
          Open('select filename from sysfiles where name = "master"');
          xPath := ExtractFilePath(qryRestore.FieldName('filename')).AsString;

          Close;
          SQL.Clear;
          SQL.Add('RESTORE DATABASE PESCDdb from disk ="' + ExtractFilePath(ParamStr(0)) +
            'DBZero.bak"');
          SQL.Add('WITH FILE = 1,');
          SQL.Add('MOVE N"TCCLUANdb" TO N"' + xPath + 'PESCDdb.MDF',');
          SQL.Add('MOVE N"TCCLUANdb_LOG" TO N"' + xPath + 'PESCDdb_LOG.LDF',');
          SQL.Add('NOUNLOAD,');
          SQL.Add('REPLACE,');
          SQL.Add('STATS = 10');

          ExecSQL;
        end;

        Result := True;
      except on E: Exception do
        ShowMessage('Erro ao restaurar o banco de dados' + sLineBreak + sLineBreak + E.Message);
      end;

    finally
      qryRestore.Free;
    end;

  end;

initialization

finalization
TConfiguracaoGeral.ReleaseInstance();

end.

```

## APÊNDICE B – Agendamento de consulta

```

unit uAgendamento;
interface
uses
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, FireDAC.Stan.Intf, FireDAC.Stan.Option,
FireDAC.Stan.Error, FireDAC.UI.Intf, FireDAC.Phys.Intf, FireDAC.Stan.Def,
FireDAC.Stan.Pool, FireDAC.Stan.Async, FireDAC.Phys, FireDAC.VCLUI.Wait,
Data.DB, FireDAC.Comp.Client, FireDAC.Stan.Param, FireDAC.DatS,
FireDAC.DApt.Intf, FireDAC.DApt, Vcl.DBGrids, Vcl.Grids, Vcl.Samples.Calendar,
Vcl.ComCtrls, FireDAC.Comp.DataSet, FireDAC.Phys.MSSQL, FireDAC.Phys.MSSQLDef,
FireDAC.Stan.StorageBin, Vcl.Menus, Vcl.ExtCtrls, Vcl.StdCtrls, Vcl.Mask,
Vcl.DBCtrls, frxClass, frxDBSet, Vcl.Buttons, uRelatoriocidadoes,
Vcl.Imaging.jpeg, frxExportPDF, frxReportHelper, uDefinicoes, ConfiguracaoGeral,
System.Actions, Vcl.ActnList, uCadPaciente;

const
NUMERO_DE_FICHAS = 15;
type
TfAgendamento = class(TForm, TInterfacePaciente)
qryAGENDAMENTO_DETALHE: TFDQuery;
dseAGENDAMENTO_DETALHE: TDataSource;
memAGENDAMENTO_DETALHE: TFDMemTable;
pnlCalendario: TPanel;
MonthCalendar1: TMonthCalendar;
qryProfissional: TFDQuery;
dseProfissional: TDataSource;
memAGENDAMENTO_DETALHEAGED_NUMERO: TIntegerField;
memAGENDAMENTO_DETALHEPES_AREA: TWideStringField;
memAGENDAMENTO_DETALHEPES_MICROAREA: TWideStringField;
memAGENDAMENTO_DETALHEPES_NOME: TStringField;
memAGENDAMENTO_DETALHEPES_NUMPRONT: TIntegerField;
qryAux: TFDQuery;
memAGENDAMENTO_DETALHEAGE_ID: TLargeintField;
Label2: TLabel;
btnExcluirAgendamento: TSpeedButton;
memAGENDAMENTO_DETALHEPACIENTE_ID: TLargeintField;
frxReport1: TfrxReport;
frxdseAGENDAMENTO_DETALHE: TfrxDBDataset;
Label4: TLabel;
qryAGENDAMENTO_DETALHEPACIENTE_ID: TLargeintField;
qryAGENDAMENTO_DETALHEAGE_ID: TLargeintField;
qryAGENDAMENTO_DETALHEAGED_NUMERO: TIntegerField;
qryAGENDAMENTO_DETALHEPES_NOME: TStringField;
qryAGENDAMENTO_DETALHEPES_AREA: TIntegerField;
qryAGENDAMENTO_DETALHEPES_NUMPRONT: TIntegerField;

```

```

qryAGENDAMENTO_DETALHEPES_MICROAREA: TIntegerField;
Panel1: TPanel;
Image1: TImage;
Panel2: TPanel;
DBGrid1: TDBGrid;
Label1: TLabel;
boxProfissional: TComboBoxEx;
boxPeriodo: TComboBox;
Label3: TLabel;
Shape1: TShape;
SpeedButton1: TSpeedButton;
frxPDFExport1: TfrxPDFExport;
SpeedButton2: TSpeedButton;
ActionList1: TActionList;
actRemoverPaciente: TAction;
actAtenderPaciente: TAction;
timerAgendamento: TTimer;
procedure MonthCalendar1Click(Sender: TObject);
procedure boxProfissionalSelect(Sender: TObject);
procedure DBGrid1DbClick(Sender: TObject);
procedure Excluir1Click(Sender: TObject);
procedure DBGrid1DrawColumnCell(Sender: TObject; const [Ref] Rect: TRect;
DataCol: Integer; Column: TColumn; State: TGridDrawState);
procedure FormActivate(Sender: TObject);
procedure memAGENDAMENTO_DETALHEAfterScroll(DataSet: TDataSet);
procedure memAGENDAMENTO_DETALHEBeforeDelete(DataSet: TDataSet);
procedure boxPeriodoSelect(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure SpeedButton1Click(Sender: TObject);
procedure btnImprimirClick(Sender: TObject);
procedure actRemoverPacienteExecute(Sender: TObject);
procedure actAtenderPacienteExecute(Sender: TObject);
procedure FormCreate(Sender: TObject);
strict private
FProfissionalID: Double;
private
FConnection: TFDCConnection;
procedure Pesquisa;
procedure SetConnection(const Value: TFDCConnection);
procedure BuscarAgendamentos(pData: TDate; pProfID: Double; pPeriodo: Integer);
function BuscarIndiceProfissionalSelecioneado: integer;
function AdicionarAgendamento(pAGE_DATA: TDate; pAGED_NUMERO: Integer;
pPES_ID: Double; pPROFISSIONAL_ID: Double): Boolean;
procedure PopularBoxProfissional;

procedure BuscarCodigoAgendamento(pAGE_DATA: TDate; pPES_ID: Double; var
xAGE_ID: Double);
procedure InserirAgendamentoDetalhe(pPES_ID: Double; pAGE_ID: Double;
pAGED_NUMERO: Integer);
procedure InserirAgendamento(pAGE_DATA: TDate; pPES_ID: Double);

```

```

function BuscarPaciente: Double;
function PermiteAgendar: Boolean;
function ExcluirAgendamentoDetalhe: Boolean;
procedure HabilitarBotoes;
public
{ Public declarations }
property Connection: TFDCConnection read FConnection write SetConnection;
procedure PrepararTela(ProfissionalID: Double = 0);
end;

var
fAgendamento: TfAgendamento;

implementation

uses
DateUtils, uPesquisaPaciente;

{$R *.dfm}

{ TForm1 }

procedure TfAgendamento.FormActivate(Sender: TObject);
begin
WindowState := wsMaximized;
MonthCalendar1.Date:= Now;
end;

procedure TfAgendamento.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Action := caFree;
fAgendamento := nil;
end;

procedure TfAgendamento.FormCreate(Sender: TObject);
begin
qryAux.Connection := TConfiguracaoGeral.GetInstance.Connection;
qryProfissional.Connection := TConfiguracaoGeral.GetInstance.Connection;
qryAGENDAMENTO_DETALHE.Connection :=
TConfiguracaoGeral.GetInstance.Connection;
end;

procedure TfAgendamento.MonthCalendar1Click(Sender: TObject);
begin
BuscarAgedamentos(MonthCalendar1.Date,          BuscarIndiceProfissionalSelecioneado,
BoxPeriodo.ItemIndex);
end;

procedure TfAgendamento.boxPeriodoSelect(Sender: TObject);
begin

```



```

BuscarAgedamentos(MonthCalendar1.Date,          BuscarIndiceProfissionalSelecionado,
BoxPeriodo.ItemIndex);
end;

```

```

procedure TfAgendamento.boxProfissionalSelect(Sender: TObject);
begin
BuscarAgedamentos(MonthCalendar1.Date,          BuscarIndiceProfissionalSelecionado,
BoxPeriodo.ItemIndex);
end;

```

```

procedure TfAgendamento.btnImprimirClick(Sender: TObject);
begin
frxReport1.SetarDescricaoCabecalhoRelatorio('lblProfissional', boxProfissional.Text);
frxReport1.SetarDescricaoCabecalhoRelatorio('lblData', FormatDateTime('DD/MM/YYYY',
MonthCalendar1.Date));
frxReport1.SetarDescricaoCabecalhoRelatorio('lblPeriodo', boxPeriodo.Text);

```

```

frxReport1.ShowReport;
end;

```

```

procedure TfAgendamento.BuscarAgedamentos(pData: TDate; pProfID: Double; pPeriodo:
Integer);
var
I: Integer;
begin
memAGENDAMENTO_DETALHE.DisableControls;
memAGENDAMENTO_DETALHE.Close;
memAGENDAMENTO_DETALHE.Open;
try
if (pData>0) and (pProfID>0) then
begin

with qryAGENDAMENTO_DETALHE do
begin
Close;
SQL.Clear;
SQL.Add('SELECT          AGENDAMENTO_DETALHE.PACIENTE_ID,
AGENDAMENTO_DETALHE.AGE_ID,');
SQL.Add('AGENDAMENTO_DETALHE.AGED_NUMERO,');
SQL.Add('PACIENTE.PES_NOME,          PACIENTE.PES_AREA,
PACIENTE.PES_NUMPRONT,');
SQL.Add('PACIENTE.PES_MICROAREA FROM AGENDAMENTO_DETALHE');
SQL.Add('INNER JOIN AGENDAMENTO ON');
SQL.Add('      AGENDAMENTO.AGE_ID = AGENDAMENTO_DETALHE.AGE_ID');
SQL.Add('INNER JOIN PESSOA PACIENTE ON');
SQL.Add('      PACIENTE.PES_ID = AGENDAMENTO_DETALHE.PACIENTE_ID');
SQL.Add('WHERE          AGENDAMENTO.AGE_DATA          =          '+
QuotedStr(FormatDateTime('YYYY-MM-DD', pData)));
SQL.Add(' AND AGENDAMENTO.PROFISSIONAL_ID = :PROFISSIONAL_ID');
SQL.Add(' AND AGENDAMENTO_DETALHE.AGED_PERIODO = :AGED_PERIODO');

```

```

end;

qryAGENDAMENTO_DETALHE.ParamByName('AGED_PERIODO').AsInteger      :=
pPeriodo;
qryAGENDAMENTO_DETALHE.ParamByName('PROFISSIONAL_ID').AsFloat     :=
pProfID;
qryAGENDAMENTO_DETALHE.Open;

if not qryAGENDAMENTO_DETALHE.IsEmpty then // verifica se nao é vazio
begin
for I := 1 to NUMERO_DE_FICHAS do //percorre o grid
begin
memAGENDAMENTO_DETALHE.Append;
memAGENDAMENTO_DETALHEAGED_NUMERO.AsInteger := I;

if qryAGENDAMENTO_DETALHE.Locate('AGED_NUMERO', I , []) then //verifica se
localizou o id se sim
begin
memAGENDAMENTO_DETALHEAGE_ID.AsFloat      :=
qryAGENDAMENTO_DETALHE.FieldByName('AGE_ID').AsFloat;
memAGENDAMENTO_DETALHEPACIENTE_ID.AsFloat :=
qryAGENDAMENTO_DETALHE.FieldByName('PACIENTE_ID').AsFloat;
memAGENDAMENTO_DETALHEPES_NOME.AsString   :=
qryAGENDAMENTO_DETALHE.FieldByName('PES_NOME').AsString; // adiciona o
campo referente ao id no grid
memAGENDAMENTO_DETALHEPES_AREA.AsString   :=
qryAGENDAMENTO_DETALHE.FieldByName('PES_AREA').AsString ;
memAGENDAMENTO_DETALHEPES_MICROAREA.AsString :=
qryAGENDAMENTO_DETALHE.FieldByName('PES_MICROAREA').AsString ;
memAGENDAMENTO_DETALHEPES_NUMPRONT.AsString :=
qryAGENDAMENTO_DETALHE.FieldByName('PES_AREA').AsString

end
else
memAGENDAMENTO_DETALHEPES_NOME.AsString := "";
memAGENDAMENTO_DETALHE.Post;
end;
end
else
begin
for I := 1 to NUMERO_DE_FICHAS do // gera numero de fichas no grid
begin
memAGENDAMENTO_DETALHE.Append;
memAGENDAMENTO_DETALHEAGED_NUMERO.AsInteger := I;
memAGENDAMENTO_DETALHEPES_NOME.AsString := "";
memAGENDAMENTO_DETALHE.Post;
end;
end;
end;

```

```

finally
memAGENDAMENTO_DETALHE.First;
memAGENDAMENTO_DETALHE.EnableControls;
end;
end;

////////// função ver profissional selecionado //////////

function TfAgendamento.BuscarIndiceProfissionalSelecionado: Integer;
begin
if boxProfissional.ItemIndex > -1 then
Result := Integer(boxProfissional.ItemsEx[boxProfissional.ItemIndex].Data)
else
Result := 0;
end;

//////////

procedure TfAgendamento.DBGrid1DbClick(Sender: TObject);
var
xCodPaciente: Double;
// xteste: TDateTime;
begin
if TConfiguracaoGeral.GetInstance.TipoUsuario = tpuMedico then
begin
//xteste:= DateOf(Now);
if memAGENDAMENTO_DETALHEPES_NOME.AsString <> " " then
//if MonthCalendar1.Date = date then
actAtenderPaciente.Execute;
end else
if PermiteAgendar then
begin
xCodPaciente := BuscarPaciente;
if xCodPaciente > 0 then
begin
AdicionarAgendamento(
MonthCalendar1.Date,
memAGENDAMENTO_DETALHEAGED_NUMERO.AsInteger,
xCodPaciente,
BuscarIndiceProfissionalSelecionado);
end;
end;

end;

////////// colorir linhas grid
procedure TfAgendamento.DBGrid1DrawColumnCell(Sender: TObject;
const [Ref] Rect: TRect; DataCol: Integer; Column: TColumn;
State: TGridDrawState);

```

```

var
I: Integer;

begin

begin
if ((DayOfWeek(MonthCalendar1.Date) in [1, 7])) or
(MonthCalendar1.Date < DateOf(Now)) then
DBGrid1.Canvas.Brush.Color := clGray
else
begin
if (memAGENDAMENTO_DETALHEPES_NOME.AsString = "") then
DBGrid1.Canvas.Brush.Color := clMoneyGreen
else
DBGrid1.Canvas.Brush.Color := clteal;
end;

DBGrid1.Canvas.FillRect(Rect);
DBGrid1.DefaultDrawDataCell(Rect, Column.Field, State);

end;

end;

function TfAgendamento.PermiteAgendar: Boolean;
begin
Result := (memAGENDAMENTO_DETALHEPES_NOME.AsString = "") and
(not (DayOfWeek(MonthCalendar1.Date) in [1, 7])) and
(MonthCalendar1.Date >= DateOf(Now)) and
(TConfiguracaoGeral.GetInstance.TipoUsuario = tpuAtendente)or
(TConfiguracaoGeral.GetInstance.TipoUsuario = tpuAdministrador);
end;

procedure TfAgendamento.Pesquisa;
begin
BuscarAgedamentos(MonthCalendar1.Date,          BuscarIndiceProfissionalSelecionado,
BoxPeriodo.ItemIndex);
end;

procedure TfAgendamento.Excluir1Click(Sender: TObject);
begin
qryAGENDAMENTO_DETALHE.Delete;
end;

////////////////////////////////////

function TfAgendamento.BuscarPaciente: Double;
begin
Result := 0;

```

```

frmPesquisaPaciente := TfrmPesquisaPaciente.Create(Self);
try
frmPesquisaPaciente.Connection := TConfiguracaoGeral.GetInstance.Connection;
if frmPesquisaPaciente.ShowModal = mrOK then
begin
Result := frmPesquisaPaciente.qryPesquisa.FieldByName('PES_ID').AsFloat;
end;
finally
frmPesquisaPaciente.Free;
end;

end;

////////////////////////////////////

procedure TfAgendamento.actAtenderPacienteExecute(Sender: TObject);
begin
if fCadPessoa = nil then
fCadPessoa := TfCadPessoa.Create(Self);
fCadPessoa.Visualizar(memAGENDAMENTO_DETALHEPACIENTE_ID.AsFloat);
fCadPessoa.Setup(Self);
fCadPessoa.Show
end;

procedure TfAgendamento.actRemoverPacienteExecute(Sender: TObject);
begin
memAGENDAMENTO_DETALHE.Delete;
BuscarAgedamentos(MonthCalendar1.Date,          BuscarIndiceProfissionalSelecionado,
BoxPeriodo.ItemIndex);
end;

function          TfAgendamento.AdicionarAgendamento(pAGE_DATA:          TDate;
pAGED_NUMERO: Integer; pPES_ID: Double; pPROFISSIONAL_ID: Double): Boolean;
var
xAGE_ID: Double;
begin
xAGE_ID := 0;

BuscarCodigoAgendamento(pAGE_DATA, pPROFISSIONAL_ID, xAGE_ID);

if (xAGE_ID > 0) then
InserirAgendamentoDetalhe(pPES_ID, xAGE_ID, pAGED_NUMERO)
else
begin
InserirAgendamento(pAGE_DATA, pPROFISSIONAL_ID);

BuscarCodigoAgendamento(pAGE_DATA, pPROFISSIONAL_ID, xAGE_ID);

if (xAGE_ID > 0) then
InserirAgendamentoDetalhe(pPES_ID, xAGE_ID, pAGED_NUMERO);

```

end;

BuscarAgedamentos(pAGE\_DATA, pPROFISSIONAL\_ID, BoxPeriodo.ItemIndex);  
end;

//

```

procedure TfAgendamento.InserirAgendamentoDetalhe(pPES_ID: Double; pAGE_ID:
Double; pAGED_NUMERO: Integer);
begin
with qryAux do
begin
Close;
SQL.Clear;
SQL.Add('INSERT AGENDAMENTO_DETALHE (AGE_ID, AGED_NUMERO,
PACIENTE_ID, AGED_PERIODO)');
SQL.Add('VALUES');
SQL.Add('(:AGE_ID, :AGED_NUMERO, :PACIENTE_ID, :AGED_PERIODO)');
end;
qryAux.ParamByName('AGE_ID').AsFloat := pAGE_ID;
qryAux.ParamByName('AGED_NUMERO').AsFloat := pAGED_NUMERO;
qryAux.ParamByName('PACIENTE_ID').AsFloat := pPES_ID;
qryAux.ParamByName('AGED_PERIODO').AsInteger := boxPeriodo.ItemIndex;

try
qryAux.ExecSQL;
except
on E: Exception do
Application.MessageBox(
PChar('Paciente já possui agendamento para esta data'),
PChar('Atenção!'));

end;

end;

```

```

procedure TfAgendamento.memAGENDAMENTO_DETALHEAfterScroll(DataSet:
TDataSet);
begin
HabilitarBotoes;
end;

```

```

procedure TfAgendamento.memAGENDAMENTO_DETALHEBeforeDelete(DataSet:
TDataSet);
begin
if Application.MessageBox(PChar('Deseja Excluir este agendamento?'), Pchar(Caption),
MB_YESNO) = ID_NO then
Abort
else
if ExcluirAgendamentoDetalhe then

```

```

BuscarAgedamentos(MonthCalendar1.Date,
BoxPeriodo.ItemIndex);

```

```

BuscarIndiceProfissionalSelecionado,

```

```

end;

```

```

function TfAgendamento.ExcluirAgendamentoDetalhe: Boolean;

```

```

begin

```

```

with qryAux do

```

```

begin

```

```

Close;

```

```

SQL.Clear;

```

```

SQL.Add('DELETE FROM AGENDAMENTO_DETALHE');

```

```

SQL.Add('WHERE AGE_ID = :AGE_ID');

```

```

SQL.Add(' AND PACIENTE_ID = :PACIENTE_ID');

```

```

end;

```

```

qryAux.ParamByName('AGE_ID').AsFloat :=

```

```

memAGENDAMENTO_DETALHEAGE_ID.AsFloat;

```

```

qryAux.ParamByName('PACIENTE_ID').AsFloat :=

```

```

memAGENDAMENTO_DETALHEPACIENTE_ID.AsFloat;

```

```

qryAux.ExecSQL;

```

```

Result := qryAux.RowsAffected > 0;

```

```

end;

```

```

////////////////////////////////////////////////////////////////

```

```

procedure TfAgendamento.InserirAgendamento(pAGE_DATA: TDate; pPES_ID: Double);

```

```

begin

```

```

with qryAux do

```

```

begin

```

```

Close;

```

```

SQL.Clear;

```

```

SQL.Add('INSERT INTO AGENDAMENTO');

```

```

SQL.Add('(AGE_DATA, PROFISSIONAL_ID)');

```

```

SQL.Add('VALUES');

```

```

SQL.Add('(' + QuotedStr(FormatDateTime('YYYY-MM-DD', pAGE_DATA)) + ',

```

```

:PROFISSIONAL_ID)');

```

```

end;

```

```

qryAux.ParamByName('PROFISSIONAL_ID').AsFloat := pPES_ID;

```

```

qryAux.ExecSQL;

```

```

end;

```

```

////////////////////////////////////////////////////////////////

```

```

procedure TfAgendamento.BuscarCodigoAgendamento(pAGE_DATA: TDate; pPES_ID:
Double; var xAGE_ID: Double);

```

```

begin

```

```

with qryAux do

```

```

begin

```

```

Close;

```

```

SQL.Clear;

```

```

SQL.Add('SELECT AGE_ID FROM AGENDAMENTO');
SQL.Add('WHERE AGE_DATA= ' + QuotedStr(FormatDateTime('YYYY-MM-DD',
pAGE_DATA)));
SQL.Add(' AND PROFISSIONAL_ID = :PROFISSIONAL_ID');
end;
qryAux.ParamByName('PROFISSIONAL_ID').AsFloat := pPES_ID;
qryAux.Open;
xAGE_ID := qryAux.FieldByName('AGE_ID').AsFloat;
qryAux.Close;
end;

```

```

////////////////////////////////////////

```

```

procedure TfAgendamento.PopularBoxProfissional;
begin
boxProfissional.Clear;
qryProfissional.Open;
qryProfissional.First;

if (FProfissionalID > 0) then
begin
if (qryProfissional.Locate('PROF_NRO_CONSELHO', FProfissionalID, [])) then
begin
boxProfissional.AddItem(
qryProfissional.FieldByName('PROF_NOME').AsString,
TObject(qryProfissional.FieldByName('PROF_ID').AsInteger));

boxProfissional.ItemIndex := 0;

boxProfissional.Enabled := False;
end
else
ShowMessage('Profissional não encontrado');
end
else
begin
while not qryProfissional.Eof do
begin
boxProfissional.AddItem(
qryProfissional.FieldByName('PROF_NOME').AsString,
TObject(qryProfissional.FieldByName('PROF_ID').AsInteger));

qryProfissional.Next;
end;
end;

end;

procedure TfAgendamento.PrepararTela(ProfissionalID: Double);
begin

```



FProfessionalID := ProfessionalID;

PopularBoxProfessional;

BuscarAgedamentos(MonthCalendar1.Date,  
BoxPeriodo.ItemIndex);

BuscarIndiceProfissionalSelecionado,

HabilitarBotoes;  
end;

```
procedure TfAgendamento.HabilitarBotoes;
begin
  case TConfiguracaoGeral.GetInstance.TipoUsuario of
    tpuAdministrador ;;
    tpuAtendente : begin
      btnExcluirAgendamento.Action := actRemoverPaciente;
      btnExcluirAgendamento.Font.Color:= clRed;
    end;
    tpuMedico : begin
      btnExcluirAgendamento.Action := actAtenderPaciente;
      btnExcluirAgendamento.Font.Color:= clGreen;
    end;
  end;
end;
```

```
btnExcluirAgendamento.Enabled :=
(memAGENDAMENTO_DETALHEPACIENTE_ID.AsInteger > 0) and
(not (DayOfWeek(MonthCalendar1.Date) in [1, 7])) and
(MonthCalendar1.Date >= DateOf(Now));
end;
```

```
procedure TfAgendamento.SetConnection(const Value: TFDCConnection);
begin
  FConnection := Value;
```

```
qryAGENDAMENTO_DETALHE.Connection := Value;
qryProfissional.Connection := Value;
end;
```

```
procedure TfAgendamento.SpeedButton1Click(Sender: TObject);
begin
  Close;
end;
```

end.

## APÊNDICE C – *Login no sistema*

```
unit uLogin;
```

```
interface
```

```
uses
```

```
Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.jpeg, Vcl.ExtCtrls,
Vcl.StdCtrls, FireDAC.Stan.Intf, FireDAC.Stan.Option, FireDAC.Stan.Error,
FireDAC.UI.Intf, FireDAC.Phys.Intf, FireDAC.Stan.Def, FireDAC.Stan.Pool,
FireDAC.Stan.Async, FireDAC.Phys, FireDAC.Phys.MSSQL, FireDAC.Phys.MSSQLDef,
FireDAC.VCLUI.Wait, FireDAC.Stan.Param, FireDAC.DatS, FireDAC.DApt.Intf,
FireDAC.DApt, Data.DB, FireDAC.Comp.DataSet, FireDAC.Comp.Client, uDefinicoes,
ConfiguracaoGeral;
```

```
type
```

```
TfLogin = class(TForm)
  edtUsuario: TEdit;
  edtSenha: TEdit;
  btnAcesso: TButton;
  Image1: TImage;
  Panel1: TPanel;
  qryLogin: TFDQuery;
  dtsLogin: TDataSource;
  qryLoginLOG_USUARIO: TStringField;
  qryLoginLOG_SENHA: TStringField;
  qryLoginLOG_PERFIL: TStringField;
  Label1: TLabel;
  Label2: TLabel;
  btnSair: TButton;
  qryLoginID_LOG: TIntegerField;
  procedure btnSairClick(Sender: TObject);
  procedure btnAcessoClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
```

```
fLogin: TfLogin;
```

```
implementation
```

```

{$R *.dfm}

uses uPrincipal, uPesquisaPaciente;

procedure TfLogin.btnAcessoClick(Sender: TObject);
begin
  qryLogin.Open;

  if qryLogin.Locate('LOG_USUARIO', edtUsuario.Text, [loCaseInsensitive ]) and
    qryLogin.Locate('LOG_SENHA', edtSenha.Text, [loCaseInsensitive ]) then
  begin
    with qryLogin do
    begin
      Close;
      SQL.Clear;
      SQL.Add('SELECT * FROM LOGIN');
      SQL.Add('WHERE LOG_USUARIO = :USUARIO');
      SQL.Add(' AND LOG_SENHA = :SENHA');
    end;

    qryLogin.ParamByName('USUARIO').AsString := edtUsuario.Text;
    qryLogin.ParamByName('SENHA').AsString := edtSenha.Text;
    qryLogin.Open;

    TConfiguracaoGeral.GetInstance().UsuarioLogado_Codigo :=
    qryLogin.FieldByName('ID_LOG').AsFloat;

    if      qryLoginLOG_PERFIL.AsString      =      'Administrador'      then
    TConfiguracaoGeral.GetInstance().TipoUsuario := tpuAdministrador;
    if      qryLoginLOG_PERFIL.AsString      =      'Atendente'          then
    TConfiguracaoGeral.GetInstance().TipoUsuario := tpuAtendente;
    if      qryLoginLOG_PERFIL.AsString      =      'Médico'              then
    TConfiguracaoGeral.GetInstance().TipoUsuario := tpuMedico;

    qryLogin.Close();

    ModalResult := mrOK;
  end
  else
  begin
    if (edtUsuario.Text = "") and (edtSenha.Text= "") then
    begin
      // mostra mensagem
      ShowMessage('Preencha os campos login e senha!');
      edtUsuario.SetFocus;
    end
    else // campos nao conferem
    begin
      // campos nao conferem
      ShowMessage('Usuario ou senha inválidos!');
    end
  end
end

```

```
edtUsuario.Clear;  
edtSenha.Clear;  
end;
```

```
Abort  
end;  
end;
```

```
procedure TfLogin.btnSairClick(Sender: TObject);  
begin  
close;  
end;
```

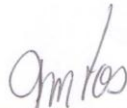
```
procedure TfLogin.FormCreate(Sender: TObject);  
begin  
qryLogin.Connection := TConfiguracaoGeral.GetInstance.Connection;  
end;
```

```
end.
```

**ANEXO A – Termo de aceite de implantação de trabalho acadêmico****TERMO DE ACEITE DE IMPLANTAÇÃO DE TRABALHO ACADÊMICO**

Autorizo o aluno: Luan Barbosa Hubner, da 8ª Fase do Curso de Ciência da Computação do Centro Universitário Unifacvest, para testes e implantação do sistema PESC (prontuário eletrônico da saúde do cidadão) sem custos para a UBS (unidade básica de saúde) Dr. Clóvis Cechin em Anita Garibaldi, apenas como confirmação de aplicação do (TCC) trabalho de conclusão de curso módulo 02.

Anita Garibaldi –SC, 19 de agosto de 2016



Gabriela Matos Varela  
Secretária Municipal de Saúde

---

Secretária da Saúde  
Gabriela Matos Varela

14.016.436/0001-83  
Fundo Mun. de Saúde  
Anita Garibaldi

Carimbo da Empresa