



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

VINICIUS OLIVEIRA HOFFMANN DE CAMPOS SILVA

CONTROLE DE ATENDIMENTO AO PACIENTE: APLICAÇÃO WEB

**Assis/SP
2018**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

VINICIUS OLIVEIRA HOFFMANN DE CAMPOS SILVA

CONTROLE DE ATENDIMENTO AO PACIENTE: APLICAÇÃO WEB

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Vinicius O. H. de Campos Silva
Orientador(a): Prof. Dr. Osmar A. Machado

**Assis/SP
2018**

FICHA CATALOGRÁFICA

S586c SILVA, Vinicius Oliveira Hoffmann de Campos
Controle de atendimento ao paciente: aplicação web / Vinicius
Oliveira Hoffmann de Campos Silva. – Assis, 2018.

44p.

Trabalho de conclusão do curso (Análise e Desenvolvimento de
Sistemas). – Fundação Educacional do Município de Assis-FEMA

Orientador: Dr. Osmar Aparecido Machado

1.Web 2.ASP.NET 3.LINQ to SQL

CDD 005.4

CONTROLE DE ATENDIMENTO AO PACIENTE: APLICAÇÃO WEB

VINICIUS OLIVEIRA HOFFMANN DE CAMPOS SILVA

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

Orientador: _____
Osmar Aparecido Machado

Examinador: _____
Celio Desiro

**Assis/SP
2018**

DEDICATÓRIA

Dedico este trabalho à minha família, que me acompanharam dia após dia, que presenciaram meus humores, que motivaram e contribuíram para a conclusão desse projeto.

AGRADECIMENTOS

Primeiramente agradeço a Deus pelas suas obras realizadas e deixadas para humanidade como lição de vida e aprendizado. Através dessas escrituras me torno a cada dia mais uma pessoa melhor. Agradeço pelas pessoas envolvidas direta ou indiretamente, contribuindo de uma certa forma no resultado desse trabalho. Agradeço o orientador Osmar A. Machado por ter como qualidade o carisma. Todos esses elementos me ajudaram a continuar nesse trabalho de conclusão de curso.

“O sucesso normalmente vem para quem está ocupado demais para procurar por ele” – Henry David

RESUMO

O uso exclusivo de planilhas para controle e armazenamento de informações pode comprometer um modelo de gestão caso falte gerenciamento e controle. O objetivo desse trabalho é desenvolver uma aplicação web que irá substituir o uso de planilhas eletrônicas em uma unidade de saúde. Usando a técnica de levantamento de requisitos por entrevista foi possível elaborar uma aplicação que a substitua na coleta das informações dos pacientes. O trabalho será desenvolvido com a IDE Visual Studio 2017 e sua linguagem C# ASP.NET, e a sua persistência com o banco de dados SQL Server 2017 será utilizado o LINQ to SQL. Em conclusão, o aplicativo será capaz de cadastrar e realizar consultas de paciente, médico e agendamento, promovendo mudanças e melhorias essenciais para o bom desenvolvimento em uma unidade de saúde.

Palavras-chave: ASP.NET, Visual Studio, LINQ to SQL, Agendamentos de Pacientes.

ABSTRACT

The exclusive use of spreadsheets for control and storage of information can compromise an administration model in the absence of management and control. The purpose of this paper is to develop a web application that will replace the use of electronic spreadsheets in a health unit. Using the interviewing requirement survey technique, it was possible to develop an application to replace it in the collection of patients data. The work will be developed in IDE Visual Studio 2017 and C# ASP.NET language, and its persistence with the SQL Server 2017 database will be used the LINQ to SQL. In conclusion, the application will be able to register and query patients, doctors and appointments, promoting changes and essential improvements for the good development in a health unit.

Key Words: ASP.NET, Visual Studio, LINQ to SQL, Patients Scheduling.

LISTA DE ILUSTRAÇÕES

Figura 1: CASO DE USO VISÃO GERAL DO SISTEMA.....	27
Figura 2: CASO DE USO LOGAR	27
Figura 3: CASO DE USO MANTER PACIENTES	29
Figura 4: CASO DE USO MANTER MÉDICOS	30
Figura 5: CASO DE USO AGENDAR CONSULTA.....	32
Figura 6: MANTER AGENDAMENTOS	33
Figura 7: CASO DE USO GERAR RALATÓRIOS	35
Figura 8: DIAGRAMA DE CLASSE	37
Figura 9: DIAGRAMA DE SEQUÊNCIA AGENDAMENTO.....	38
Figura 10: DIAGRAMA DE ATIVIDADE AGENDAMENTO.....	39

LISTA DE TABELAS

Tabela 1: NARRATIVA LOGAR.....	28
Tabela 2: NARRATIVA MANTER PACIENTES	30
Tabela 3: NARRATIVA MANTER MÉDICOS.....	31
Tabela 4: NARRATIVA AGENDAR CONSULTA	33
Tabela 5: NARRATIVA MANTER AGENDAMENTOS.....	34
Tabela 6: NARRATIVA GERAR RELATÓRIOS.....	36

LISTAS DE ABREVIATURAS E SIGLAS

IBGE	Instituto Brasileiro de Geografia e Estatística
XLS	<i>Extensible Style Language</i>
SQL	<i>Structured Query Language</i>
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
UML	<i>Unified Modeling Language</i>
WWW	<i>World Wide Web</i>

SUMÁRIO

1. INTRODUÇÃO	13
2. FERRAMENTAS E TECNOLOGIAS UTILIZADAS NO PROJETO.....	15
2.1. ASTAH	15
2.1.1. Sql Server Management Studio	17
2.1.2. Microsoft Visual Studio Community.....	18
2.1.3. C# (C Sharp)	18
2.1.4. Linq to Sql.....	19
2.1.5. Levantamento de Requisitos	20
2.1.6. Técnicas de Levantamento de Requisitos	21
2.1.7. Levantamento Orientado a Pontos de Vista	21
2.1.8. Etnografia	22
2.1.9. Workshops.....	22
2.1.10. Prototipagem.....	23
2.1.11. Entrevistas	23
2.1.12. Questionários.....	23
2.1.13. Brainstorming	23
2.1.14. JAD.....	24
3. PROJETO: DESENVOLVIMENTO DO SISTEMA.....	24
2.2. PLANEJAMENTO DOS REQUISITOS.....	24
2.3. ELABORAÇÃO E MODELAGEM DO SISTEMA	25
2.3.1. Diagrama de Caso de Uso	26
2.3.2. Diagrama de Classe	36
2.3.3. Diagrama de Sequência.....	37
2.3.4. Diagrama de Atividade	38
3. CONCLUSÃO	40
4. REFERÊNCIAS.....	41

1. INTRODUÇÃO

Em 1960, a expectativa de vida do brasileiro era de 48 anos. Atualmente, em 2018, alcança os 75 anos de idade. De acordo com dados do Instituto Brasileiro de Geografia e Estatística (IBGE) a tecnologia é a chave para o avanço da medicina (2016), que resulta evidentemente nesse aumento da longevidade dos indivíduos.

A evolução da informática possibilitou o surgimento de softwares que visam, dentre diversas atividades, melhorar a eficiência e organização do armazenamento das informações na área da saúde, com a promessa de não só substituir as anotações em papel, mas também elevar a qualidade da assistência à saúde através de novos recursos e aplicações (VAN GINNEKEN E MOORMAN, 1997).

Com o passar dos anos a tecnologia foi evoluindo tornando-se rápida e otimizada na prestação de serviços, passou a ser não apenas um diferencial e sim uma necessidade buscada e aprimorada com o tempo. Com a ajuda da ciência junto com a tecnologia obteve vários experimentos, sendo eles: Raio X, Ultrassom, Tomografia computadorizada, ressonância magnética, impressoras 3D, cirurgia robótico, Órgãos artificiais e nanotecnologia que contribuíram significativamente na área da saúde, potencializando os diagnósticos de patologias, na melhoria da qualidade de vida, e na possibilidade de as pessoas acessarem bens e serviços de saúde disponibilizados na internet.

O uso da tecnologia promoveu agilidade no atendimento à área da saúde. Por meio de aplicativos é possível realizar agendamentos, enviar exames e/ou informações sobre um atendimento, por exemplo. As vantagens da utilização do software promovem maior agilidade, organização e segurança da informação para a área da saúde. Pacientes que chegam com o quadro de saúde delicado, que precisam de um rápido atendimento, são encaminhados para salas de exames e são monitoradas por esses softwares.

Atualmente, as transformações tecnológicas na área da saúde se encontram cada vez mais rápidas e a cada momento surgem novas técnicas diferentes e aparatos mais modernos no mercado (COUTO Daniela, REGIANA Eliane, JESUS Josiane, LUIZ Gelson, LORENZINI Alacoque, 2006)

Partindo do que é realizado na unidade de saúde de Maracaí e levantando informações sobre o controle de agendamento dos pacientes, observou-se que a atividade é realizada pelo uso de planilha. Essas planilhas são compartilhadas na rede e quando são acessadas por mais de uma pessoa elas são abertas apenas no modo de visualização, ou seja, não podem ser alteradas.

Outro problema observado foram os computadores com o sistema operacional Windows terem sido formatados para Linux. Isso ocorreu um erro de compatibilidade de planilha, devido de não ter nenhuma compatibilidade com a ferramenta de outro sistema. A partir do momento que a planilha é aberta com outro aplicativo possivelmente ocorre um erro de formatação dos dados, perdas de informação ou simplesmente não abrir.

Alinhado à esta realidade, o presente trabalho tem como objetivo a desenvolver uma aplicação web que seja executada através do browser de um computador, ajudando o profissional na área da saúde, em uma unidade básica de saúde, na coleta das informações, como Cadastrar Paciente, Cadastrar Médico, Realizar Agendamento, Consultar Paciente, Consultar Médico, Consultar Agendamento, Gerar Relatório.

A primeira justificativa é descartar o uso de aplicações para desktop, ou seja, desenvolver uma aplicação WEB, podendo acessar de qualquer computador através do browser utilizando o caminho da URL.

A vantagem de utilizar recursos online é simplesmente de acessar a informação de um modo fácil e rápido em qualquer dispositivo que tenha acesso a internet. A segunda justificativa é descartar o uso de planilhas como armazenamento de dados e controle.

2. FERRAMENTAS E TECNOLOGIAS UTILIZADAS NO PROJETO

Este capítulo tratará da metodologia utilizada no desenvolvimento desse trabalho descrevendo cada ferramenta que será utilizada.

Hoje em dia é necessário a utilização de uma metodologia de trabalho. O dicionário (AURÉLIO, 2002) define metodologia como um conjunto de métodos, regras e um consenso utilizado em determinada disciplina e sua aplicação.

O método de desenvolvimento será com softwares da Microsoft licenciado para estudantes.

- ✓ Astah professional 7.2.0
Ferramenta UML utilizada na modelagem de requisitos.
- ✓ SQL Server Management Studio 17.4
Sistema gerenciador de Banco de dados relacional (SGBD).
- ✓ Microsoft Visual Studio Community 2017 15.5.3
Ambiente de desenvolvimento integrado da Microsoft para desenvolvimento de software especialmente dedicado ao .NET.
- ✓ C# (C Sharp)
Linguagem de Programação desenvolvida pela Microsoft como parte da plataforma .NET.
- ✓ LINQ to SQL
Componente do .NET Framework 3.5 que fornece uma infraestrutura de tempo de execução para gerenciar dados relacionais como objetos.

2.1. ASTAH

Astah é um software para modelagem UML (Linguagem de Modelagem Unificada). Desenvolvida por Change Vision, anteriormente conhecido como JUDE um acrônimo de Java and UML Developers Environment (Ambiente para Desenvolvimento UML e Java).

Astah disponibiliza uma lista de diagramas para o desenvolvedor:

- ✓ Diagrama de Sequência

- ✓ Diagrama de Comunicação
- ✓ Diagrama de Visão Geral de Interação
- ✓ Diagrama de Tempo
- ✓ Diagrama de Interação
- ✓ Diagrama de Máquina de Estados
- ✓ Diagrama de Caso de Uso
- ✓ Diagrama de Atividades
- ✓ Diagrama de Comportamento
- ✓ Diagrama de Pacotes
- ✓ Diagrama de Implantação
- ✓ Diagrama de Estruturas Compostas
- ✓ Diagrama de Perfil
- ✓ Diagrama de Objetos
- ✓ Diagrama de Componentes
- ✓ Diagrama de Classes
- ✓ Diagrama de Estruturas

O software foi desenvolvido no Japão na plataforma Java, o que garante sua portabilidade para qualquer plataforma que possui JVM (Máquina Virtual Java). Anteriormente conhecido como JUDE obteve o prêmio “Produto de Software do ano 2006”, pela Agencia de Promoção de Informação Tecnológica no Japão. Algumas empresas que já usaram Astah são Amazon.com, Google e Oracle.

Martin Fowler (2009) nos diz que, UML são notações gráficas, ou seja, desenhos de referência de um projeto deixando-o de fácil entendimento. Um diagrama, quando elaborado pode transmitir qual é a ideia central, ajudando-o a entendê-lo. O principal meio de comunicação entre os membros de uma equipe de desenvolvedores são os conjuntos consideráveis de diversos diagramas que o ajudam a definir uma aplicação.

Criada por Grady Booch, Ivar Jacobson & James Rumbaugh, a UML é hoje o método mais comum para o paradigma orientada a objetos. Os objetos da UML são: especificação, documentação, estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo do sistema de informação. Carneiro (2015).

2.1.1. **Sql Server Management Studio**

SQL é uma sigla que significa (Structured Query Language) traduzida no português (Linguagem de Consulta Estruturada), padrão para acesso a banco de dados usados em inúmeros sistemas, tais como:

- ✓ MySQL
- ✓ Oracle
- ✓ Sybase
- ✓ Access
- ✓ DB2
- ✓ PostgreSQL
- ✓ SQL Server

Nesse projeto foi utilizado o banco SQL Server da Microsoft. Segundo Kalen Delaney (2009) diz, em 1988 a Microsoft lançou sua primeira versão do SQL Server, foi desenvolvida para a plataforma OS/2 juntamente com a Sybase. Em 1994 a Microsoft e a Sybase formalmente encerrarão sua parceria. De 1995 até a atualidade a Microsoft vem lançando melhorias significativas em suas novas versões do SQL Server. Sua última versão foi em 2017.

Miranda (2016) descreve que, Banco de dados são conjuntos de arquivos organizados para criar algum sentido e facilitar buscas e pesquisas. Com o avanço da tecnologia, os bancos de dados passaram a ser armazenados em softwares. Para as empresas, um sistema de gerenciamento de banco de dados eficaz é vital para o bom andamento das operações. Com ele, é possível manter organizados os registros de empregados, folhas de pagamentos, contabilidade, gestão de projetos, inventários e muitas outras informações. Um banco de dados capaz de relacionar e extrair informações de várias tabelas ao mesmo tempo é conhecido como “banco de dados relacional”. Os mais simples, que trabalham somente com uma tabela de cada vez, são chamados de “banco de dados de arquivo simples”. Contudo, a grande maioria dos programas de banco de dados modernos são relacionais. Concluimos que banco de dados é a alma da empresa,

seja ela uma microempresa ou uma empresa gigantesca como a Google, sem os dados as pessoas não têm informações para a tomada de decisão.

2.1.2. Microsoft Visual Studio Community

O Microsoft Visual Studio .NET é uma ferramenta de desenvolvimento que surgiu no final de 2001, que acabou revolucionando a maneira de se programar, facilitando no desenvolvimento de aplicações web e dispositivos móveis o que antes era complexo de fazer. A IDE da Microsoft é especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e J# (J Sharp). Além de ser uma IDE de desenvolvimento na área web, usando a plataforma ASP.NET como websites, aplicativos webs, serviços webs e aplicativos móveis. Com o avanço da tecnologia a Microsoft vem lançando novas funcionalidades, novos recursos para sua IDE Visual Studio, deixando sempre o desenvolvedor com os recursos nas palmas das mãos. Troelsen (2009, pag. 45) comenta, esta ferramenta é, de longe, a IDE mais rico em recursos.

Segundo Ribeiro (2017) em, 7 de novembro de 2005, Microsoft lançou o Visual Studio 2005, que incluiu Visual Basic 8. NET Framework versão 2. Em 27 de fevereiro de 2008, a Microsoft lançou o Visual Studio 2008, incluindo Visual Basic 9. NET Framework versão 3.5. Em 12 de abril de 2010, a Microsoft lançou o Visual Studio 2010 e NET Framework 4.

2.1.3. C# (C Sharp)

O C# é uma linguagem de programação cuja sintaxe principal é bastante similar à sintaxe Java. Tanto C# quanto Java fazem parte da família C de linguagens de programação (C, Objective C, C++, etc.) e, portanto, compartilham uma sintaxe similar. Assim como Java, de muitas maneiras, é uma versão mais limpa do C++, o C# pode ser visto como uma versão mais limpa do Java. A verdade é que grande parte das estruturas sintáticas de C# é modelada de acordo com diversos aspectos do Visual Basic e C++. Troelsen (2009, pag. 7).

MSDN (2012) ressalta que, o processo de compilação do C# é mais simples e mais flexível comparado ao C++ ou Java, pois elimina-se a necessidade de arquivos de

cabeçalhos separados e não há a necessidade da declaração de métodos e tipos em uma ordem específica. Um arquivo contendo códigos de programação podem ser definidos quaisquer números de classes, estruturas, interfaces e eventos que forem necessários.

De acordo com OFICINA DA NET (2007), comparada ao C e C++, a linguagem C# é restrita e melhorada em vários aspectos, tais como:

- Unsafe mode: ponteiros e fórmulas aritméticas utilizados sem verificação de integridade só podem ser utilizados na modalidade de modo inseguro;
- Overflow: acessos a objetos são realizados através de referências seguras, dessa forma não podem ser invalidadas e normalmente as operações aritméticas são checadas contra sobrecarga;
- Garbage collector: os objetos não são liberados da memória explicitamente, e sim através de um processo que elimina objetos quando não há referências;
- Disposable: Destrutores não existem, mas utilizada junto com a construção using block, torna possível que recursos alocados por um objeto sejam liberados.

Segundo Araújo (2013), C# é mais seguro com tipos: as conversões implícitas por default são conversões seguras, tais como ampliação de inteiros e conversões de um tipo derivado para um tipo base; não existem conversões implícitas entre inteiros e variáveis lógicas ou enumerações; não existem ponteiros nulos; todas as conversões implícitas definida pelo usuário deve ser marcada explicitamente.

Ainda de acordo com OFICINA DA NET (2007), as diferenças do C# com o Java são: diferentemente de Java, no C# são implementados propriedades e sobrecargas de operadores, implementa modo inseguro que permite a manipulação de ponteiros e aritméticos sem checagem, exceções não são checadas, implementa goto como estrutura de controle, oferece suporte a indexadores e delegates.

2.1.4. Linq to Sql

LINQ to SQL é uma API que permite expressões de consulta LINQ em dados guardando dentro de bancos de dados relacionais. Propicia um número de tipos (dentro do Assembly System.Data.Linq.dll) que facilitam a comunicação entre sua base de código e o motor do banco de dados físico. O objetivo principal é proporcionar consistência entre banco de

dados relacionais e a lógica de programação utilizada para interagir com eles. Troelsen (2009, pag. 778).

2.1.5. Levantamento de Requisitos

O início para todas as atividades de desenvolvimento de software é o levantamento de requisitos. Esta atividade é repetida em todas as etapas da engenharia de requisitos.

Sommerville (2003) propõe um processo genérico de levantamento e análise que contém as seguintes atividades:

- **Compreensão do domínio:** Os analistas devem desenvolver sua compreensão do domínio da aplicação;
- **Coleta de requisitos:** É o processo de interagir com os stakeholders do sistema para descobrir seus requisitos. A compreensão do domínio se desenvolve mais durante essa atividade;
- **Classificação:** Essa atividade considera o conjunto não estruturado dos requisitos e os organiza em grupos coerentes;
- **Resolução de conflitos:** Quando múltiplos stakeholders estão envolvidos, os requisitos apresentarão conflitos. Essa atividade tem por objetivo solucionar esses conflitos;
- **Definição das prioridades:** Em qualquer conjunto de requisitos, alguns serão mais importantes do que outros. Esse estágio envolve interação com os stakeholders para a definição dos requisitos mais importantes;
- **Verificação de requisitos:** Os requisitos são verificados para descobrir se estão completos e consistentes e se estão em concordância com o que os stakeholders desejam do sistema.

Identifica-se um levantamento de requisitos conveniente através da boa definição do projeto, da efetividade do projeto, de informações necessárias a um perfeito diagnóstico e de soluções inteligentes. Quanto ao levantamento de requisitos impróprio, o resultado é um diagnóstico pobre com conclusões prejudicadas, não identificar das causas dos

problemas, custos elevados, prazos vencidos ou comprometedores, omissão de processos fundamentais e perda.

2.1.6. Técnicas de Levantamento de Requisitos

Às técnicas de levantamento de requisitos tem por objetivo superar as dificuldades relativas a esta fase. Todas as técnicas possuem um conceito próprio e suas respectivas vantagens e desvantagens, que podem ser utilizadas em conjunto pelo analista. Serão apresentadas de forma resumida nessa etapa algumas técnicas de levantamento de requisitos.

2.1.7. Levantamento Orientado a Pontos de Vista

Para qualquer sistema, de tamanho médio ou grande, normalmente há diferentes tipos de usuário final. Muitos stakeholders têm algum tipo de interesse nos requisitos do sistema. Por esse motivo, mesmo para um sistema relativamente simples, existem muitos pontos de vista diferentes que devem ser considerados. Os diferentes pontos de vista a respeito de um problema 'veem' o problema de modos diferentes. Contudo, suas perspectivas não são inteiramente independentes, mas em geral apresentam alguma duplicidade, de modo que apresentam requisitos comuns.

As abordagens orientadas a ponto de vista, na engenharia de requisitos, reconhecem esses diferentes pontos de vista e utiliza-os para estruturar e organizar o processo de levantamento e os próprios requisitos. Uma importante capacidade da análise orientada a pontos de vista é que ela reconhece a existência de várias perspectivas e oferece um framework para descobrir conflitos nos requisitos propostos por diferentes stakeholders. Sommerville (2003).

2.1.8. **Etnografia**

Nesta técnica, o analista se insere no ambiente de trabalho em que o sistema será utilizado. O trabalho diário é observado e são anotadas as tarefas reais em que o sistema será utilizado. O principal objetivo da etnografia é que ela ajuda a descobrir requisitos de sistema implícitos, que refletem os processos reais, em vez de os processos formais, onde as pessoas estão envolvidas.

Etnografia é particularmente eficaz na descoberta de dois tipos de requisitos:

Os requisitos derivados da maneira como as pessoas realmente trabalham, em vez da maneira pelas quais as definições de processo dizem como elas deveriam trabalhar;

Os requisitos derivados da cooperação e conscientização das atividades de outras pessoas.

A análise de observação tem algumas desvantagens como, consumir bastante tempo e o analista ser induzido a erros em suas observações. Mas em geral a técnica de observação é muito útil e frequentemente usada para complementar descobertas obtidas por outras técnicas. Sommerville (2003).

2.1.9. **Workshops**

Trata-se de uma técnica de elicitação em grupo usada em uma reunião estruturada. Devem fazer parte do grupo uma equipe de analistas e uma seleção dos *stakeholders* que melhor representam a organização e o contexto em que o sistema será usado, obtendo assim um conjunto de requisitos bem definidos.

Ao contrário das reuniões, onde existe pouca interação entre todos os elementos presentes, o *workshop* tem o objetivo de acionar o trabalho em equipe. Há um facilitador neutro cujo papel é conduzir a *workshop* e promover a discussão entre os vários mediadores. As tomadas de decisão são baseadas em processos bem definidos e com o objetivo de obter um processo de negociação, mediado pelo facilitador. Sommerville (2003).

2.1.10. **Prototipagem**

Protótipo tem por objetivo explorar aspectos críticos dos requisitos de um produto, implementando de forma rápida um pequeno subconjunto de funcionalidades deste produto. O protótipo é indicado para estudar as alternativas de interface do usuário; problemas de comunicação com outros produtos; e a viabilidade de atendimento dos requisitos de desempenho. As técnicas utilizadas na elaboração do protótipo são várias: interface de usuário, relatórios textuais, relatórios gráficos, entre outras. Sommerville (2003).

2.1.11. **Entrevistas**

A entrevista é uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados. Convém que o entrevistador dê margem ao entrevistado para expor as suas ideias. É necessário ter um plano de entrevista para que não haja dispersão do assunto principal e a entrevista fique longa, deixando o entrevistado cansado e não produzindo bons resultados. Sommerville (2003).

2.1.12. **Questionários**

O uso de questionário é indicado, por exemplo, quando há diversos grupos de usuários que podem estar em diversos locais diferentes do país. Neste caso, elaboram-se pesquisas específicas de acompanhamento com usuários selecionados, que a contribuição em potencial pareça mais importante, pois não seria prático entrevistar todas as pessoas em todos os locais. Sommerville (2003).

2.1.13. **Brainstorming**

Brainstorming é uma técnica para geração de ideias. Ela consiste em uma ou várias reuniões que permitem que as pessoas sugiram e explorem ideias. Sommerville (2003).

2.1.14. JAD

O JAD facilita a criação de uma visão compartilhada do que o produto de software deve ser. Através da sua utilização os desenvolvedores ajudam os usuários a formular problemas e explorar soluções. Dessa forma, os usuários ganham um sentimento de envolvimento, posse e responsabilidade com o sucesso do produto.

Nesse projeto utilizaremos a técnica de levantamento de requisitos por entrevista, logo a frente será descrita as etapas. Sommerville (2003).

3. PROJETO: DESENVOLVIMENTO DO SISTEMA

2.2. Planejamento dos requisitos

É a primeira etapa em que se preocupa com o início do projeto de um software. Nela, foi conversado com o cliente e levantado os requisitos para entender o objetivo do sistema, e como ele irá atender as necessidades do cliente.

Foram identificados os seguintes perfis de usuários no sistema proposto, o administrador e o usuário.

Administrador é um usuário com privilégios para gerenciar outros usuários do sistema.

- ✓ Gerenciador de usuários: controlar o cadastro, alterações e exclusão de usuários.
 - Criar usuário
 - Excluir usuário
 - Alterar senha
 - Alterar privilégios

O perfil identificado como Usuário nas documentações do sistema refere-se ao atendente na unidade de saúde, que é a pessoa responsável por atender os pacientes do posto de saúde e realizar agendamentos no sistema.

- ✓ Manter pacientes: controlar o cadastro, alterações e exclusão de pacientes.

- Cadastrar paciente
 - Pesquisar paciente
 - Excluir paciente
 - Alterar paciente
 - Agendar paciente
- ✓ Manter médico
- Cadastra médico
 - Pesquisar médico
 - Excluir médico
 - Alterar médico
- ✓ Manter agendamentos
- Pesquisar agendamento
 - Alterar agendamento
 - Excluir agendamento
- ✓ Manter relatórios:
- Gerar relatório de paciente
 - Gerar relatório de médico
 - Gerar relatório de consultas em aberto
 - Gerar relatório de consultas encerradas
 - Gerar relatório de consultas em retorno
 - Gerar relatório de consultas canceladas
 - Gerar relatório de Retornos
 - Gerar relatório de usuários do sistema (Somente administrador pode realizar).

2.3. Elaboração e Modelagem do sistema

É a segunda etapa em que as informações obtidas através da concepção, ou seja, com a conversa com o cliente anteriormente na primeira etapa, nas quais são expandidas e

refinadas. A elaboração é realizada através de criação e refinamento de cenários de usuários que descrevem como o usuário final irá interagir com o sistema. Nas etapas seguintes serão mostrados:

- ✓ Diagrama de Caso de Uso
- ✓ Diagrama de ER
- ✓ Diagrama de Classe
- ✓ Diagrama de Atividade

A modelagem é uma das principais atividades que levam à implementação de um bom software.

2.3.1. Diagrama de Caso de Uso

Os Casos de Uso foram popularizados originalmente por Ivar Jacobson e embora já existissem há algum tempo, havia pouca padronização ao seu uso. Em relação e levantamentos de requisitos os Casos de Uso são ferramentas valiosas, tornando de fácil entendimento as funcionalidades do sistema. Vale ressaltar que os Casos de Uso são uma visão externa do sistema, em outras palavras, não tem quaisquer correlações entre as classes. Martin Fowler (2004) enfatiza cuidado ao utilizar os Casos de Uso, pois podem apresentar alguns problemas como ser pequeno demais gerando dúvidas e perguntas ou ser grande demais dificultando a leitura e sua compreensão.

2.3.1.1. Caso de Uso Visão Geral do Sistema

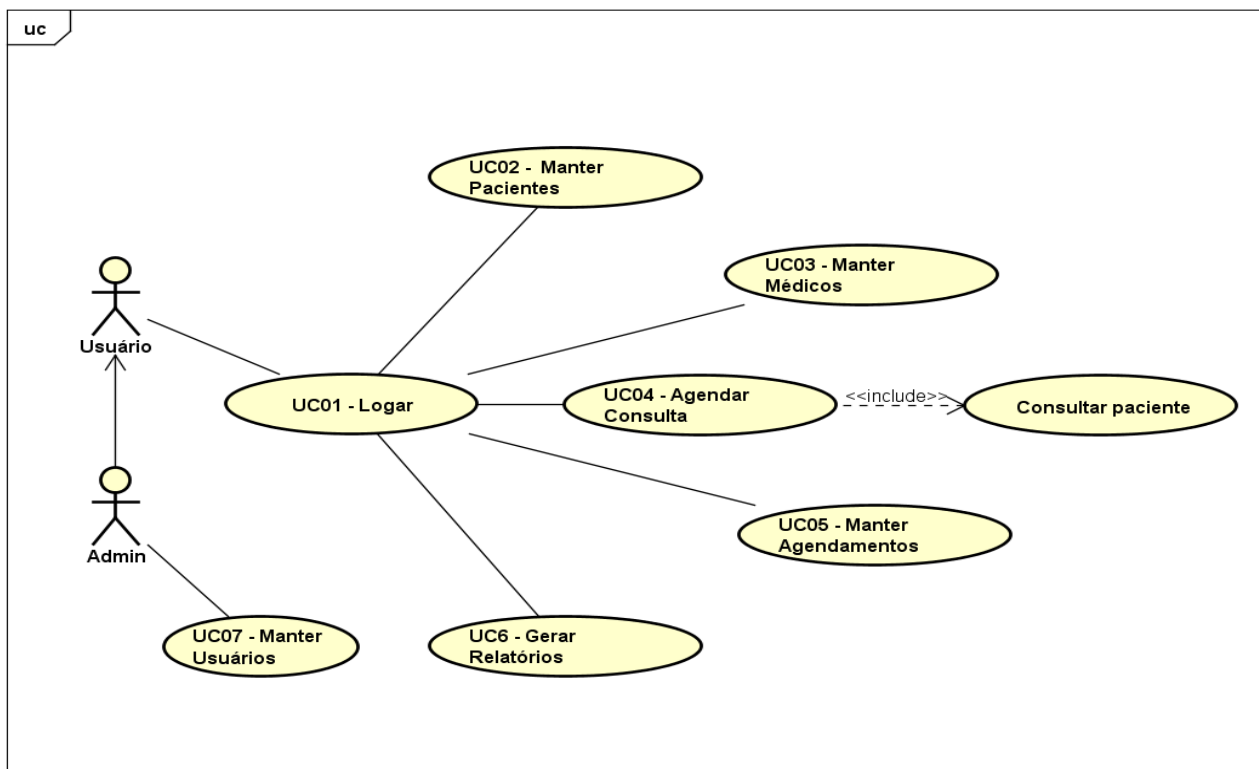


Figura 1: CASO DE USO VISÃO GERAL DO SISTEMA

2.3.1.2. Case de Uso Logar

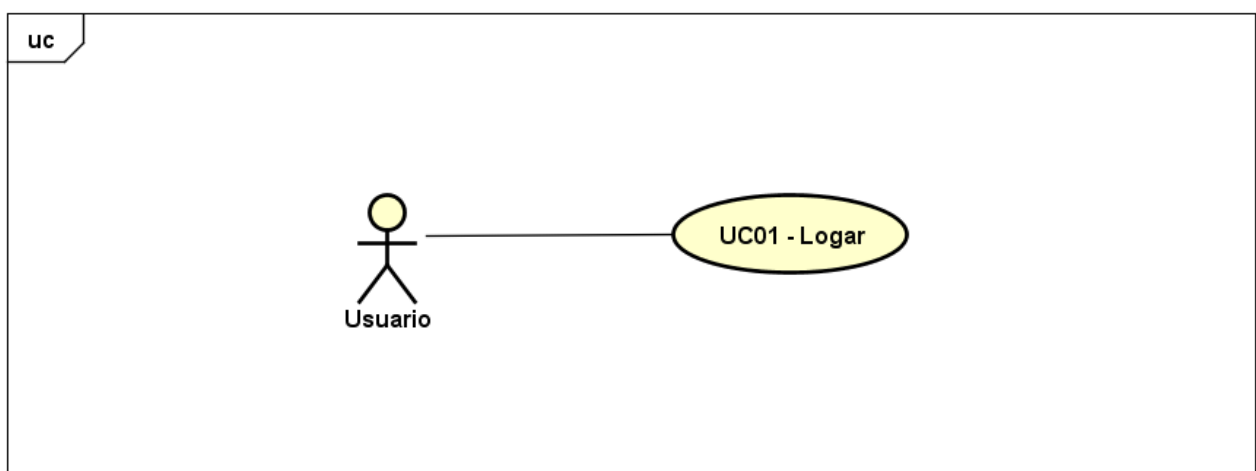


Figura 2: CASO DE USO LOGAR

Identificação: UC01	
Caso de Uso: Logar	
Ator: Usuário	
Tipo: Primário	
Pré – condição: Usuário cadastrado	
Pós – condição: Usuário logado	
SEQUÊNCIA TÍPICA DE EVENTOS	
ATOR	SISTEMA
1. Informa usuário e senha	2. Valida o acesso
SEQUÊNCIA ALTERNATIVA	
1.a Acesso cancelado pelo usuário	
2.a Usuário ou senha incorretos	
2.b Usuário não cadastrado na base de dados.	

Tabela 1: NARRATIVA LOGAR

2.3.1.3. Caso de Uso Manter Pacientes

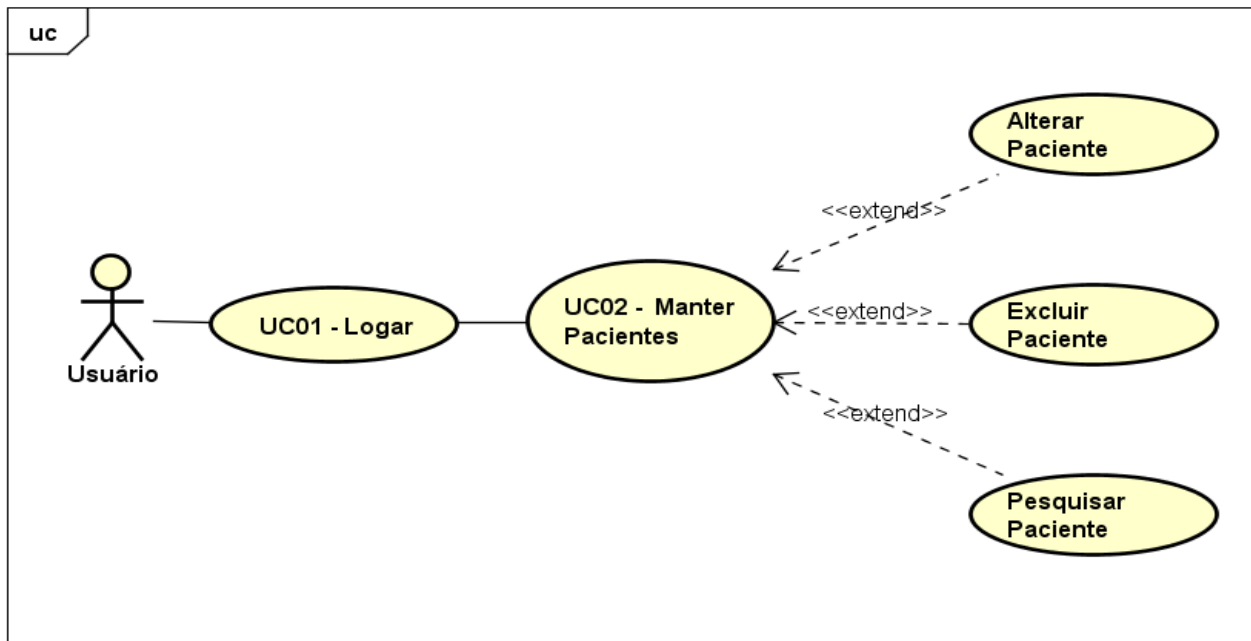


Figura 3: CASO DE USO MANTER PACIENTES

Identificação: UC02	
Caso de Uso: Manter Pacientes	
Ator: Usuário	
Tipo: Primário	
Pré – condição: Usuário logado	
Pós – condição: Paciente Cadastrado	
SEQUÊNCIA TÍPICA DE EVENTOS	
ATOR	SISTEMA
1. Clica em Manter Pacientes	2. Abre a Interface
3. Preenche os campos necessários	4. Confere os campos com seus requisitos

5. Clica em Salvar	6. Valida as informações e salva no banco
SEGUÊNCIA ALTERNATIVA	
1.a Usuário pode fechar a interface	
1.b Usuário pode escolher em pesquisar paciente	
1.c Usuário pode escolher em alterar paciente	
1.d Usuário pode escolher em excluir paciente	
4.a Sistema pode retornar um erro por não atender os requisitos	
6.a Sistema não salva devido a existência do erro	

Tabela 2: NARRATIVA MANTER PACIENTES

2.3.1.4. Caso de Uso Manter Médicos

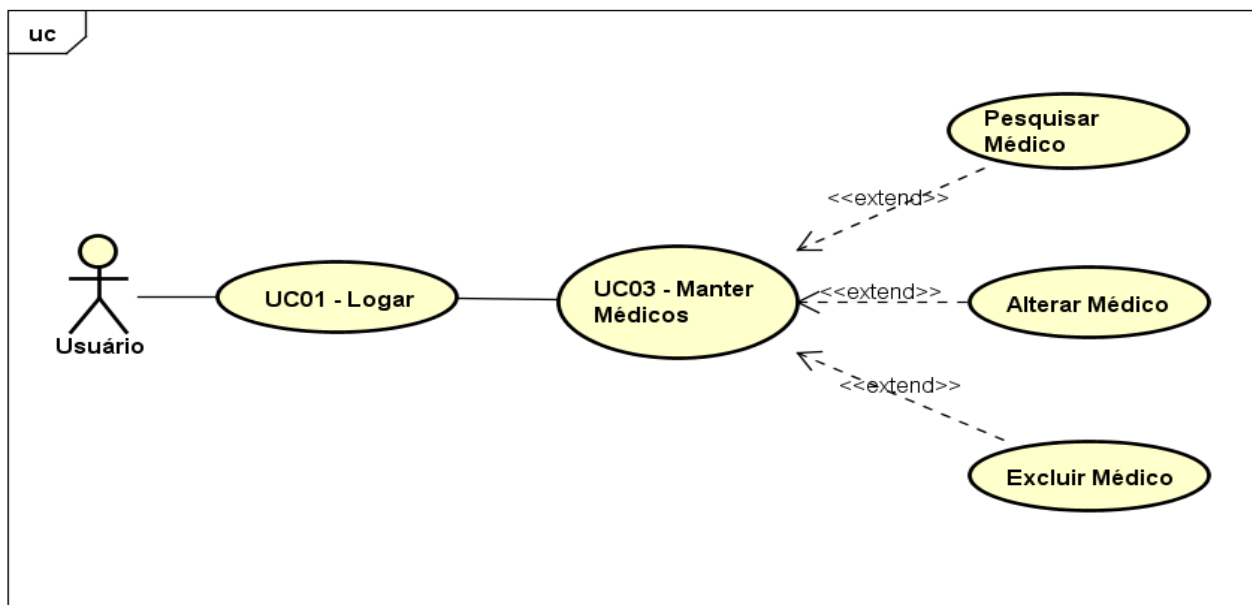


Figura 4: CASO DE USO MANTER MÉDICOS

Identificação: UC03	
Caso de Uso: Manter Médicos	
Ator: Usuário	
Tipo: Primário	
Pré – condição: Usuário logado	
Pós – condição: Médico cadastrado	
SEQUÊNCIA TÍPICA DE EVENTOS	
ATOR	SISTEMA
1. Clica em Manter Médicos	2. Abre a interface
3. Preenche os campos necessários	4. Confere os campos com seus requisitos
5. Clica em Salvar	6. Valida as informações e salva no banco
SEQUÊNCIA ALTERNATIVA	
1.a Usuário pode fechar a interface	
1.b Usuário pode escolher em pesquisar médico	
1.c Usuário pode escolher em alterar médico	
1.d Usuário pode escolher em excluir médico	
4.a Sistema pode retornar um erro por não atender os requisitos	
6.a Sistema não salva devido a existência do erro	

Tabela 3: NARRATIVA MANTER MÉDICOS

2.3.1.5. Caso de Uso Agendar Consulta

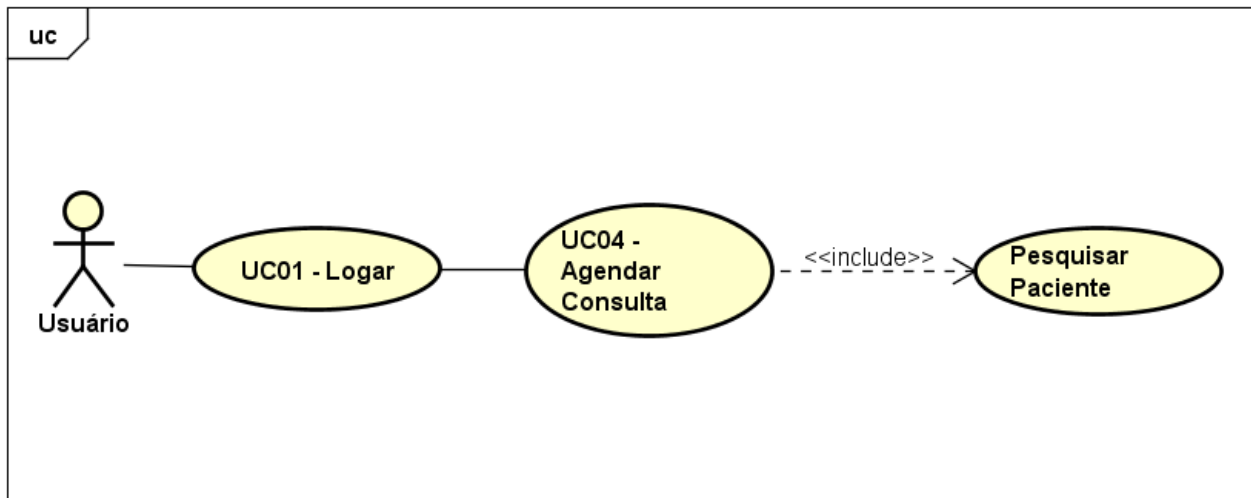


Figura 5: CASO DE USO AGENDAR CONSULTA

Identificação: UC04	
Caso de Uso: Agendar Consulta	
Ator: Usuário	
Tipo: Primário	
Pré – condição: Usuário logado	
Pós – condição: Consulta agendada	
SEQUÊNCIA TÍPICA DE EVENTOS	
ATOR	SISTEMA
1. Clica em Agendar Consulta	2. Abre a interface
3. Pesquisa o paciente	4. Retorna à informação do paciente
5. Preenche os campos requeridos	6. Confere os campos e seus requisitos

7. Clica em Salvar	8. Valida as informações e salva no banco
SEQUÊNCIA ALTERNATIVA	
1.a Usuário pode fechar a interface	
4.a Sistema pode não retornar o paciente "Paciente inexistente"	
5.a Usuário pode cancelar o agendamento	
6.a Sistema aponta os campos vazios ou erro de requisitos	
8.a Sistema não salva devido a existência do erro.	

Tabela 4: NARRATIVA AGENDAR CONSULTA

2.3.1.6. Manter Agendamentos

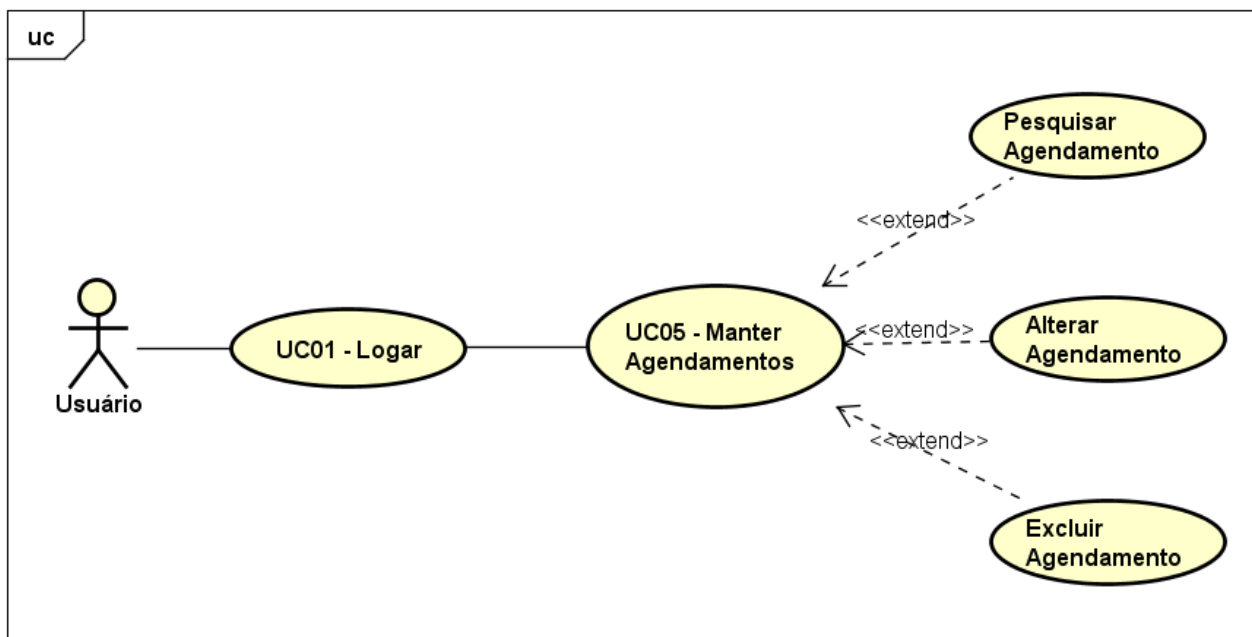


Figura 6: MANTER AGENDAMENTOS

Identificação: UC05	
Caso de Uso: Manter Agendamentos	
Ator: Usuário	
Tipo: Primário	
Pré – condição: Consulta realizada	
Pós – condição: Agendamento em andamento	
SEQUÊNCIA TÍPICA DE EVENTOS	
ATOR	SISTEMA
1. Clica em Manter Agendamentos	2. Abre a interface
3. Pesquisa a consulta	4. Retorna à consulta
5. Clica em visualizar	6. Abre a interface com os dados contidos na consulta
SEQUÊNCIA ALTERNATIVA	
1.a Usuário pode fechar a interface	
4.a Sistema pode não retornar à consulta “Consulta inexistente”	
5.a Usuário pode cancelar a visualização	

Tabela 5: NARRATIVA MANTER AGENDAMENTOS

2.3.1.7. Caso de Uso Gerar Relatórios

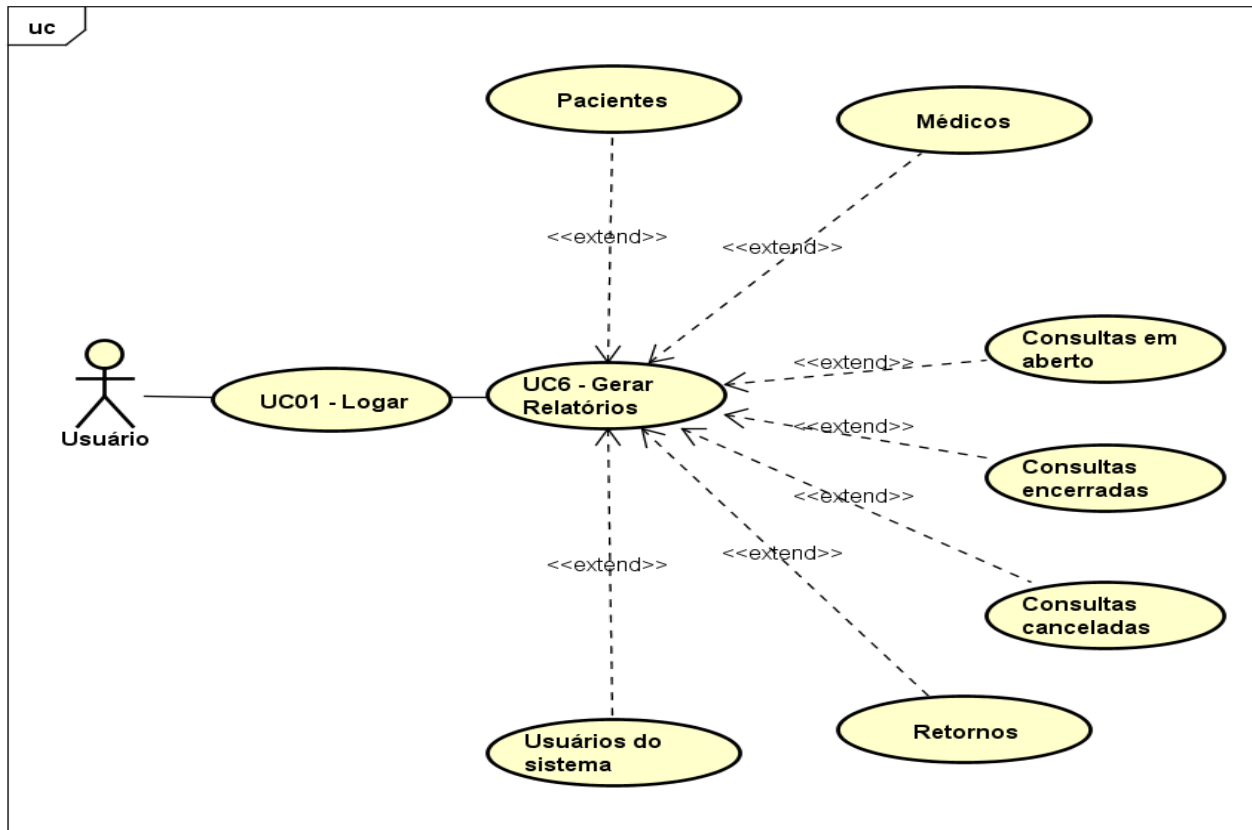


Figura 7: CASO DE USO GERAR RALATÓRIOS

Identificação: UC06
Caso de Uso: Gerar Relatórios
Ator: Usuário
Tipo: Primário
Pré – condição: Usuário logado
Pós – condição: Imprimindo relatório
SEQUÊNCIA TÍPICA DE EVENTOS

ATOR	SISTEMA
1. Clica em Gerar Relatórios	2. Abre a interface
3. Filtra por categoria	4. Retorna os dados
5. Clica em visualizar impressão	6. Exibe impressão
7. Clica em imprimir	8. Imprimi
SEQUÊNCIA ALTERNATIVA	
1.a Usuário fecha a interface	
3.a Usuário não escolhe nenhuma opção de categoria	
4.a Sistema retorna em branco “Não a informações”	
5.a Usuário não clica em visualizar	
7.a Usuário não imprimi	

Tabela 6: NARRATIVA GERAR RELATÓRIOS

2.3.2. Diagrama de Classe

Os Diagramas de Classe são espinhas dorsais da UML e referem-se aos tipos de objetos que são encontrados no sistema e o relacionamento existente entre eles. Mostram as propriedades e operações e suas restrições à maneira como os objetos estão conectados. As caixas do diagrama de classe são divididas em três compartimentos: nome da classe, seus atributos e suas operações. FOWLER (2005)

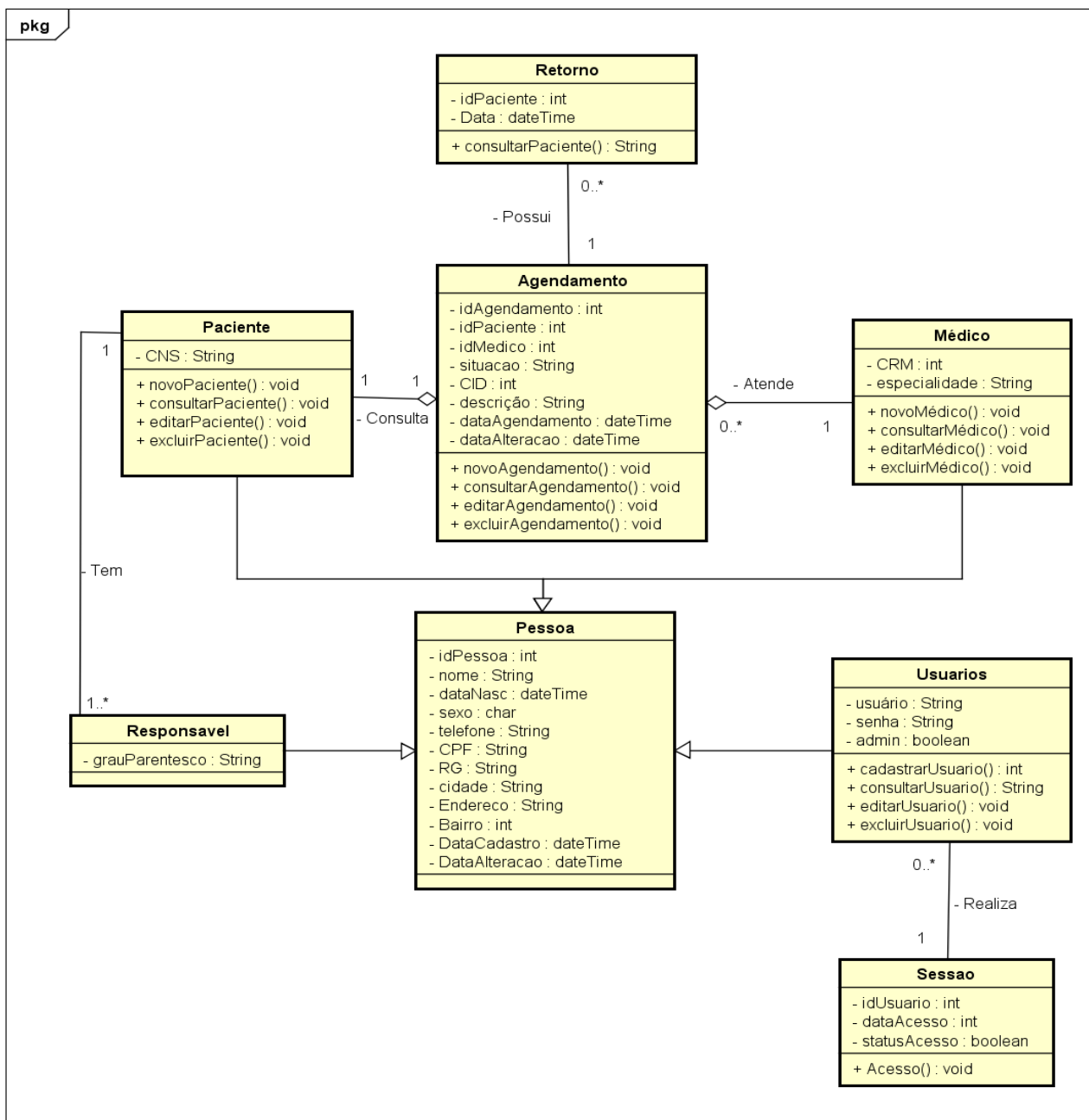


Figura 8: DIAGRAMA DE CLASSE

2.3.3. Diagrama de Sequência

O Diagrama de Sequência é um diagrama comportamental que se preocupa com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. GUEDES (2018)

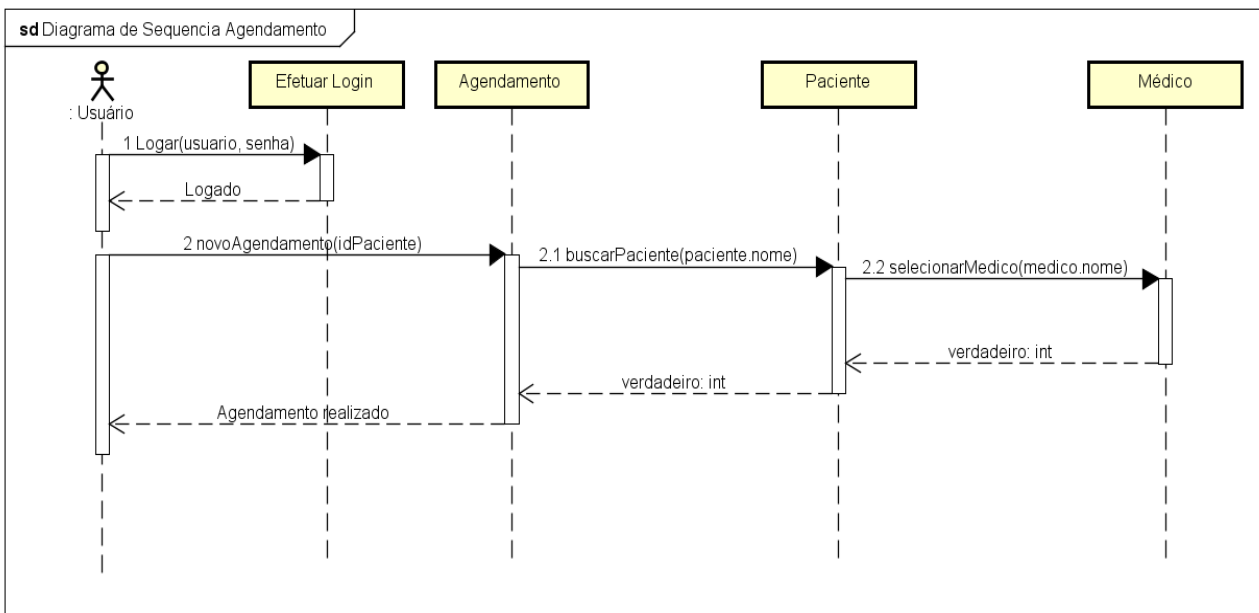


Figura 9: DIAGRAMA DE SEQUÊNCIA AGENDAMENTO

2.3.4. Diagrama de Atividade

O Diagrama de Atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo estar a ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo um processo completo. GUEDES (2018)

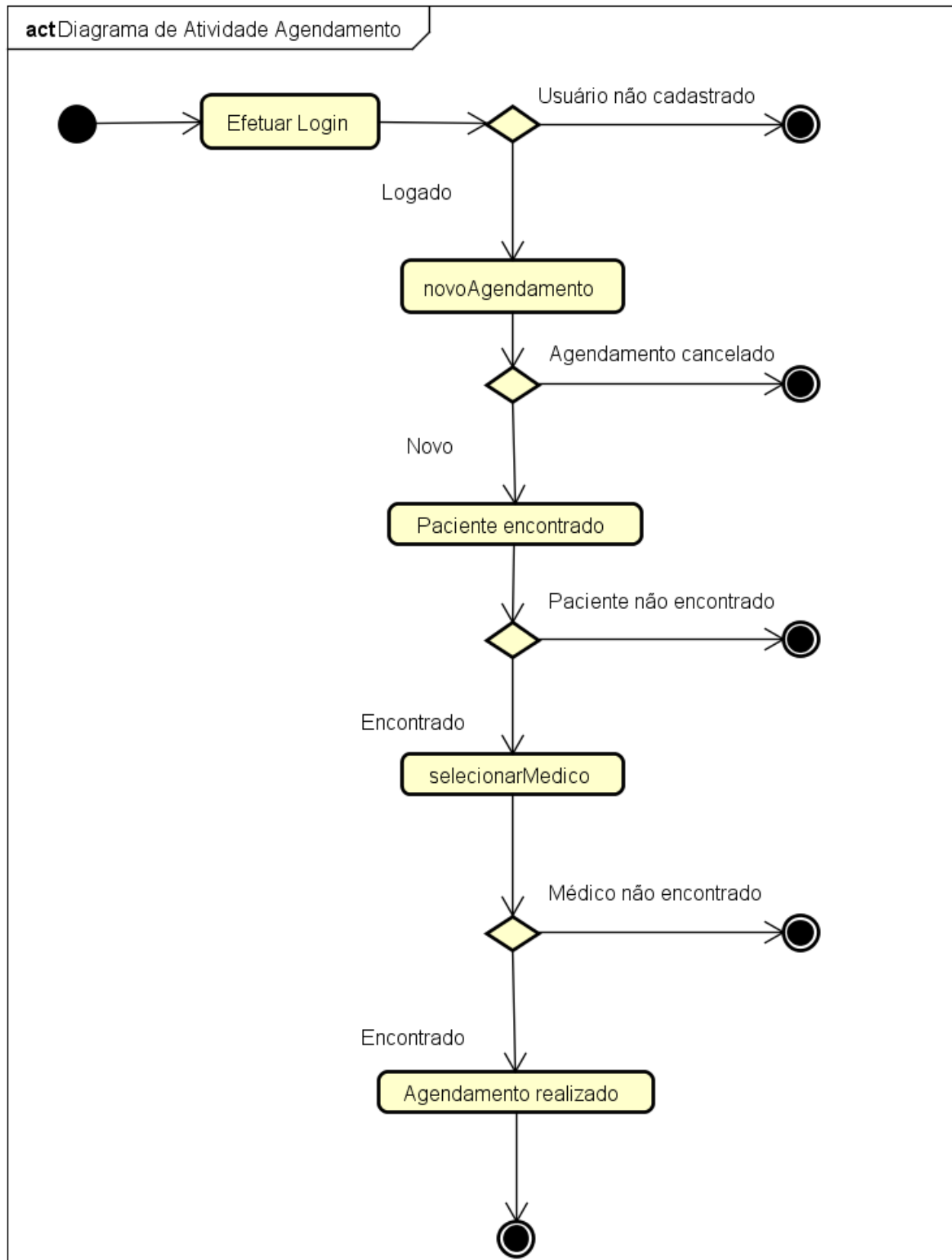


Figura 10: DIAGRAMA DE ATIVIDADE AGENDAMENTO

3. CONCLUSÃO

O presente trabalho de conclusão de curso contribuiu por meio dos estudos das ferramentas e tecnologias utilizadas nesse projeto a desenvolver um sistema que seja capaz de gerenciar e controlar as informações durante o agendamento de um paciente.

O aplicativo será capaz de cadastrar e realizar consultas de paciente, médico e agendamento, promovendo mudanças e melhorias essenciais para o bom desenvolvimento na área da saúde.

A vantagem de utilizar recursos online é simplesmente de acessar a informação de um modo fácil e rápido em qualquer dispositivo que tenha acesso a internet.

Esse trabalho foi desenvolvido com a intenção de cooperar com a unidade de saúde de Maracá sem fins lucrativos.

Por meio desse trabalho obtive conhecimento para desenvolver , implementar e gerenciar uma infraestrutura de tecnologia de informação.

4. REFERÊNCIAS

ARAUJO, Everton Coimbra de. A evolução da linguagem de programação C#. 2013. Disponível em: <<https://www.devmedia.com.br/a-evolucao-da-linguagem-de-programacao-csharp/28639>>. Acesso em: 13 abr. 2013.

CARNEIRO, Bruno Seabra (Ed.). O QUE É ASTAH. 2015. Talita Lombardi. Disponível em: <<https://www.startupsstars.com/2015/10/o-que-e-o-astah-posttecnico-por-bruno-seabra/>>. Acesso em: 29 ago. 2015.

C#(CSHARP) o que é esta linguagem? 2007. Disponível em: <https://www.oficinadanet.com.br/artigo/526/c_sharp_csharp_o_que_e_esta_linguagem>. Acesso em: 26 set. 2007.

EVOLUÇÃO HISTÓRICA E IMPACTO DA TECNOLOGIA NA ÁREA DA SAÚDE E DA ENFERMAGEM. Goiânia, Go: Revista Eletrônica de Enfermagem, v. 8, n. 3, 2006. Trimestral (quatro Números Por Ano). Faculdade de Enfermagem/UFG. Disponível em: <http://www.fen.ufg.br/revista/revista8_3/v8n3a13.htm>. Acesso em: 24 mar. 2018.

EVOLUÇÃO HISTÓRICA E IMPACTO DA TECNOLOGIA NA ÁREA DA SAÚDE E DA ENFERMAGEM (Daniela C, Eliane R, Josiane J, Gelson L, Alacoque L, 2006).

FOWLER, Martin. UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman, 2005. 160 p. João Tortello.

GUEDES, Gilleanes T. A. UML 2: UMA ABORDAGEM PRÁTICA. 3. Ed. São Paulo: Novatec Editora Ltda, 2018. 496 p. (978-85-7522-644-5). Rubens Prates. Disponível em: <<https://books.google.com.br/>>. Acesso em: 25 mar. 2018.

Introdução à linguagem C# e ao .NET Framework. 2015. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework#feedback>>. Acesso em: 19 jul. 2015.

MIRANDA, Willian. O que é um banco de dados e qual sua importância para uma empresa. 2016. Disponível em: <<https://www.linkedin.com/pulse/o-que-%C3%A9-um-banco-de-dados-e-qual-sua-import%C3%A2ncia-para-william-miranda>>. Acesso em: 3 maio 2016.

PMI. PMBOK. 4. ed. Newtown Square, Pennsylvania: PMI, 2008. 497 p.

RIBEIRO, César. A história do Visual Basic. 2017. Disponível em: <<https://www.portalgsti.com.br/2017/07/a-historia-do-visual-basic.html>>. Acesso em: 22 jul. 2017.

TROELSEN, Andrew. Profissional C# e a plataforma .NET 3.5 Curso Completo: Dominando completamente o universo .NET. Rio de Janeiro: Alta Books, 2009. 1292p.

Bianca Capitânio.

Van Ginneken, A.M., Moorman, P.W. The Patient Record. In: van Bommel, J.H., Musen, M.A.(eds.). **Handbook of Medical Informatics**. Houten, the Netherlands: Bohn Stafleu Van Loghum, 1997.