



**INSTITUTO
FEDERAL**

Paraná

INSTITUTO FEDERAL DO PARANÁ

KELVIN HEY

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA GERENCIAL PARA
UNIDADES DE PRONTO ATENDIMENTO**

IRATI

2023

KELVIN HEY

**ANÁLISE E DESENVOLVIMENTO DE UM SISTEMA GERENCIAL PARA
UNIDADES DE PRONTO ATENDIMENTO**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal do Paraná, como requisito parcial da avaliação, para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas

Orientador: Francis Luiz Baranoski

IRATI

2023

AGRADECIMENTOS

A Deus, e também à minha família.

Observei outra coisa debaixo do sol. Aquele que corre mais rápido nem sempre ganha a corrida, e o guerreiro mais forte nem sempre vence a batalha. Às vezes os sábios passam fome, os sensatos não enriquecem, e os instruídos não alcançam sucesso. Tudo depende de se estar no lugar certo na hora certa.

RESUMO

As Unidades de Pronto Atendimento (UPAs) são essenciais para o atendimento rápido e eficiente de pacientes em situações emergenciais de saúde. No entanto, a complexidade e o volume das atividades realizadas nessas unidades de saúde podem acabar sobrecarregando os profissionais de saúde que compõem a equipe, bem como comprometer a qualidade do atendimento prestado. Nesse contexto, o presente trabalho apresenta a análise e o desenvolvimento de um Sistema de Informação Gerencial projetado especificamente para atender às necessidades operacionais e administrativas desta importante instalação de Saúde pública. O sistema proposto tem como foco principal auxiliar na gerência dos processos internos, fornecendo meios eficazes para organizar o grande volume de informações dos pacientes que recebem atendimento diariamente, bem como melhorar a gestão de recursos e a comunicação entre os membros da equipe de saúde.

Palavras chave: Unidades de Pronto Atendimento, Sistema de Informação Gerencial, *Software*, urgência e emergência.

ABSTRACT

Emergency Care Centers are essential for the prompt and efficient treatment of patients in emergent health situations. However, the complexity and volume of activities carried out in these healthcare units can lead to overwhelming the healthcare professionals who make up the team, as well as compromising the quality of care provided. In this context, the present study introduces the analysis and development of a Management Information System designed specifically to meet the operational and administrative needs of this important public health facility. The proposed system primarily aims to assist in managing internal processes by providing effective means to organize the extensive patient information received daily, while also enhancing resource management and communication among workers of the healthcare team.

Keywords: Emergency Care Units, Management Information System, public health, process efficiency, urgency and emergency.

LISTA DE ILUSTRAÇÕES

Figura 1 - Acolhimento com Classificação de Risco	pág. 20
Figura 2 - Fluxograma das etapas de atendimento do ACR	pág. 23
Figura 3 - Funcionamento de um Sistema	pág. 23
Figura 4 - Atividades de um Sistema de Informação	pág. 24
Figura 5 - Sistema Philips Tasy	pág. 24
Figura 6 - Sistema SPDATA Gestão Hospitalar	pág. 24
Figura 7 - Modelo de desenvolvimento em cascata	pág. 34
Figura 8 - Modelo de desenvolvimento em V	pág. 35
Figura 9 - Valores da metodologia ágil	pág. 37
Figura 10 - Etapas de desenvolvimento utilizando FDD	pág. 39
Figura 11 - Níveis de funcionalidades da FDD	pág. 41
Figura 12 - UML em Cores	pág. 43
Figura 13 - Exemplo de aplicação usando MVC	pág. 45
Figura 14 - Casos de Uso	pág. 52
Figura 15 - Diagrama de Atividades	pág. 55
Figura 16 - Diagrama de Classes	pág. 56
Figura 17 - Modelo Entidade Relacionamento	pág. 57
Figura 18 - Interface do IntelliJ IDEA	pág. 60
Figura 19 - Interface do MySQL Workbench	pág. 61
Figura 20 - Sintaxe do CSS	pág. 63
Figura 21 - Interface do draw.io	pág. 64

LISTA DE TABELAS

Tabela 1 - Porte da UPA 24h	pág. 18
Tabela 2 - Média de atendimentos na UPA 24h	pág. 18
Tabela 3 - Infraestrutura da UPA 24h	pág. 19
Tabela 4 - Tempo de atendimento ACR	pág. 20
Tabela 5 - Elementos de um Sistema de Informação	pág. 24
Tabela 6 - Tipos de Sistema de Informação	pág. 26
Tabela 7 - Funcionalidades do SPDATA Gestão Hospitalar	pág. 32
Tabela 8 - Requisitos Funcionais	pág. 48
Tabela 9 - Requisitos Não Funcionais	pág. 49
Tabela 10 - Atores (casos de uso)	pág. 51
Tabela 11 - Descrição dos casos de uso	pág. 53
Tabela 12 - Tecnologias utilizadas	pág. 58

LISTA DE ABREVIATURAS E SIGLAS

UPA	UPA
UBS	Unidade Básica de Saúde
RUE	Rede de Atenção às Urgências e Emergências
SUS	Sistema Único de Saúde
STM	Sistema de Triagem Manchester
ACR	Acolhimento com Classificação de Risco
CRM	Conselho Federal de Medicina
MS	Ministério da Saúde
SIS	Sistema de Informação em Saúde
SBIS	Sociedade Brasileira de Informática
PEP	Prontuário Eletrônico do Paciente
OMS	Organização Mundial de Saúde
SIG	Sistema de Informação Gerencial
IDE	<i>Integrated Development Environment</i>
FDD	<i>Feature Driven Development</i>
MVC	<i>Model View Controller</i>
POO	Programação Orientada a Objetos
SGBD	Sistema Gerenciador de Banco de Dados
UML	<i>Unified Modeling Language</i>
CDU	Caso de Uso
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hypertext Markup Language</i>
SQL	<i>Structured Query Language</i>
IDE	<i>Integrated Development Environment</i>
JDBC	<i>Java Database Connectivity</i>
JDK	<i>Java Development Kit</i>
JEE	<i>Java Enterprise Edition</i>
JRE	<i>Java Runtime Environment</i>
JSE	<i>Java Standard Edition</i>
JVM	<i>Java Virtual Machine</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 CONTEXTUALIZAÇÃO	12
1.2 PROBLEMATIZAÇÃO	13
1.3 JUSTIFICATIVA	14
1.4 OBJETIVOS	15
1.4.1 Objetivo Geral	15
1.4.2 Objetivos Específicos	15
1.5 ORGANIZAÇÃO DO TRABALHO	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 UNIDADES DE PRONTO ATENDIMENTO	17
2.2 ACOLHIMENTO COM CLASSIFICAÇÃO DE RISCO	19
2.2 SISTEMAS DE INFORMAÇÃO	23
2.2.1 Tipos de Sistema de Informação	25
2.2.1.1 Sistema de Informação Gerencial	27
2.2.1.2 Sistema de Informação em Saúde	28
3 SISTEMAS SIMILARES	30
3.1 PHILIPS TASY	30
3.2 SPDATA GESTÃO HOSPITALAR	31
4 METODOLOGIA E PROCESSO DE DESENVOLVIMENTO DE <i>SOFTWARE</i>	33
4.1 <i>SOFTWARE</i>	33
4.2 METODOLOGIAS DE DESENVOLVIMENTO DE <i>SOFTWARE</i>	33
4.2.1 Metodologias Tradicionais	34
4.2.2 Metodologias Ágeis	35
4.2.2.1 <i>Feature Driven Development</i>	37
4.2.2.1.1 Desenvolver um modelo abrangente	39
4.2.2.1.2 Construção da lista de Funcionalidades	40
4.2.2.1.3 Planejamento por Funcionalidade	41
4.2.2.1.4 Detalhar por Funcionalidade	41
4.2.2.1.5 Construção por funcionalidade	41
4.2.2.1.6 Pessoas envolvidas	42
4.2.2.1.7 Práticas da FDD	42
4.3 PADRÕES DE ARQUITETURA DE <i>SOFTWARE</i>	45
4.3.1 <i>Model-View-Controller</i>	45
4.4 LINGUAGEM DE MODELAGEM	46
5 ANÁLISE DE REQUISITOS DO SISTEMA	47
5.1 LEVANTAMENTO DE REQUISITOS	47
5.1.1 Requisitos Funcionais	47
5.1.2 Requisitos não Funcionais	49

5.2	MODELAGEM DO SISTEMA	50
5.2.1	Diagrama de Casos de Uso	50
5.2.2	Diagrama de Atividades	54
5.2.3	Diagrama de Classes	55
5.2.4	Banco de Dados	57
5.2.2.2	Modelo Entidade Relacionamento	57
6	DESENVOLVIMENTO DO SISTEMA	58
6.1	TECNOLOGIAS UTILIZADAS	58
6.1.1	Java	59
6.1.2	JavaFX	59
6.1.3	Intellij IDEA	59
6.1.4	MySQL	60
6.1.5	MySQL Workbench	61
6.1.6	Git	61
6.1.7	GitHub	62
6.1.7	CSS	62
6.1.8	Draw.io	63

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Uma Unidade de Pronto Atendimento (UPA) é um centro de saúde que opera 24 horas por dia durante toda a semana, tendo como principal objetivo atender todas as pessoas que buscam atendimento médico. A UPA se caracteriza por concentrar atendimentos médicos de pequena e média complexidade, sendo necessário o atendimento rápido e imediato para pacientes em estado de urgência e emergência.

Estas unidades de saúde foram desenvolvidas com o propósito de aliviar o elevado fluxo de pacientes nos hospitais de grande porte, sendo possível dessa forma reduzir as extensas filas de espera nesses hospitais. As UPAs são estabelecimentos que funcionam como uma ponte entre as Unidades Básicas de Saúde (UBS) e os hospitais, estando elas inseridas na Rede de Atenção às Urgências e Emergências (RUE).

As UPAs proporcionam assistência em situações de urgência relacionadas a cuidados pediátricos, questões clínicas e também tratamento odontológico. Além disso, elas desempenham funções de acolhimento, triagem por classificação de risco, realização de testes laboratoriais e radiografias, bem como observação individual. Cada unidade inclui espaços destinados ao tratamento de casos mais graves, bem como leitos para observação pediátrica e clínica. Algumas dessas unidades também contam com instalações para administração de medicamentos e terapia de nebulização (BRASIL, 2023).

1.2 PROBLEMATIZAÇÃO

A gestão apropriada de uma UPA envolve uma série de desafios burocráticos e frequentemente a equipe de profissionais se depara com problemas complexos. Atualmente, o principal entrave enfrentado por essa unidade de saúde é a superlotação causada pelo elevado número de pessoas que procuram por atendimento médico diariamente nessas unidades.

Essa alta procura ocorre pelo fato de que uma UPA tem como propósito oferecer atendimento rápido a situações de urgência e emergência. Devido a essa característica, a população em geral costuma buscar essas unidades de saúde com frequência, o que geralmente faz com que a equipe médica fique sobrecarregada. Esse atendimento, por sua vez, resulta em uma considerável quantidade de dados, abrangendo informações como a data e hora exatos do atendimento ao paciente, ficha de acompanhamento, encaminhamentos entre diferentes salas, administração de medicamentos, alta e encaminhamento, além da realização de diversos exames, entre outros procedimentos. Tais dados, obrigatoriamente devem ser registrados pelos profissionais de saúde da unidade.

Embora seja notável o uso de sistemas informatizados nas instituições de saúde atualmente, muitas UPAs ainda utilizam fichas de atendimento manuais no dia a dia para registrar informações dos pacientes e suas consultas. Tal fato pode ocorrer devido a diversos fatores, como falta de recursos financeiros, falta de sistemas de informação disponíveis ou até mesmo falta de capacitação ou conhecimento tecnológico por parte dos profissionais.

Consoante o fato de os processos internos serem feitos com pouca ou nenhuma assistência de um sistema informatizado, os procedimentos realizados na UPA podem se tornar mais lentos e propensos a erros. Tal situação acaba gerando inúmeros esforços para que a gestão eficiente de uma UPA 24h seja possível, sendo de vital importância, portanto, o uso de mecanismos sofisticados no processamento da informação para a correta administração desses dados.

1.3 JUSTIFICATIVA

Contemporaneamente, o uso de sistemas informatizados já é uma realidade na ampla maioria dos setores da sociedade, pois esses sistemas melhoram grandemente a qualidade dos serviços prestados à população. Entre os benefícios obtidos pelo uso de um Sistema de Informação pode-se citar uma maior segurança dos dados, maior produtividade, melhoria da qualidade do serviço e também maior velocidade na geração de informações rápidas e confiáveis. Em relação à área de saúde, a implementação de um Sistema de Informação resulta em aprimoramentos na qualidade, eficiência e eficácia do atendimento prestado. Isso viabiliza a condução de pesquisas, a disponibilização de evidências e contribui para o aprimoramento do processo de ensino (MARIN, 2010).

Um Sistema de Informação Gerencial (SIG) é um tipo de sistema informatizado de gestão que fornece recursos de gerenciamento de informações para melhor tomada de decisões e também planejamento estratégico de uma organização. Nesse sentido, Oliveira (2008, p. 59) define que “o sistema de informação gerencial é representado pelo conjunto de subsistemas, visualizados de forma integrada e capaz de gerar informações necessárias ao processo decisório”.

Um Sistema de Informação Gerencial é capaz de produzir informações que atendam a diversas necessidades de tomada de decisão da gestão administrativa de uma organização (O'BRIEN, 2004). Considerando o fato de que uma UPA deve fornecer atendimento rápido a pessoas que na maioria das vezes encontram-se em situações de risco, o uso de um SIG torna-se de extrema importância, pois este pode ser utilizado no processamento dos dados gerados pelo atendimento aos pacientes, utilizando essas informações de forma a melhor atendê-los. Além disso, também é possível ter as informações produzidas e processadas de forma rápida e eficiente, tornando a informação disponível a qualquer momento para os médicos e demais membros da equipe de atendimento, reduzindo assim o tempo e agilizando os processos internos da UPA.

O *software* proposto neste trabalho se constitui, portanto, em um Sistema de Informação capaz de gerenciar de forma prática, segura e eficaz os processos realizados internamente em uma UPA 24h, agilizando assim a rotina desta através do controle do atendimento, admissão, alta, gerenciamento do estoque de medicamentos entre outros procedimentos.

O Sistema de Informação tornará os processos realizados na UPA mais ágeis e transparentes. Será possível, por exemplo, monitorar o andamento do atendimento prestado ao paciente enquanto este permanece na unidade, melhorando assim o atendimento fornecido ao

paciente. O sistema também fornecerá apoio na redução do tempo de espera e ajudará a minimizar erros administrativos, proporcionando assim uma experiência mais satisfatória aos pacientes. Além disso, os profissionais de saúde poderão obter informações valiosas a partir dos dados coletados pelo sistema. Nesse contexto, é relevante ressaltar as palavras de Marin (2010, p. 22):

[...] Os profissionais de saúde precisam de informação para poder exercer processo de cuidado, de gerenciamento, de avaliação. Ou seja, todas as atividades em saúde estão relacionadas com a busca e o uso da informação. Neste sentido, quanto melhor os sistemas informatizados conseguem registrar, armazenar e disponibilizar esta informação, tanto melhor será o ato do profissional – melhor informação, maior qualidade na tomada de decisão

Dessa maneira, as informações fornecidas pelo sistema irão desempenhar um papel facilitador na tomada de decisões estratégicas, promovendo a melhoria contínua dos serviços oferecidos pela UPA.

1.4 OBJETIVOS

Esta seção irá abordar os Objetivos Geral e Específicos do trabalho.

1.4.1 Objetivo Geral

Desenvolver um Sistema de Informação que possibilite gerenciar os processos internos de uma UPA de forma mais eficiente e ágil, apoiando assim o bom funcionamento da mesma.

1.4.2 Objetivos Específicos

Os objetivos específicos determinados para o *software* a ser desenvolvido neste trabalho correspondem a:

- a. Gerenciar o atendimento e admissão de pacientes na UPA;
- b. Gerenciar o processo de classificação de riscos dos pacientes;
- c. Gerenciar leitos e acomodações disponíveis;
- d. Gerenciar o controle de estoque de medicamentos;

- e. Permitir a geração de relatórios;

1.5 Organização dos capítulos

O capítulo 2 tem ênfase na fundamentação teórica, abordando os conceitos básicos sobre Unidades de Pronto Atendimento e também sobre sistemas de informação. O capítulo 3 mostra uma relação de sistemas similares ao proposto neste trabalho. O capítulo 4 aborda conceitos sobre *software* e diferentes metodologias de desenvolvimento, incluindo metodologias tradicionais e ágeis. O capítulo 5 apresenta os detalhes sobre o levantamento de requisitos, tanto funcionais como não funcionais. No capítulo 6 são retratadas quais tecnologias serão utilizadas no desenvolvimento do sistema. Por fim, no Capítulo 7 se faz menção às telas do sistema desenvolvido e outras informações pertinentes ao Sistema.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 UNIDADES DE PRONTO ATENDIMENTO

Uma UPA é uma unidade básica de saúde que deve funcionar 24 horas por dia, todos os dias da semana, compondo assim uma rede organizada de atenção às urgências e emergências. Funciona como uma das portas de entrada para o Sistema Único de Saúde (SUS) e tem como propósito atender casos de urgência e emergência de baixa e média complexidade, fornecendo atendimento médico rápido e eficiente às necessidades da população (BRASIL, 2023). Esta unidade de saúde foi concebida tendo como propósito descentralizar e agilizar o atendimento, aliviando a demanda dos prontos-socorros hospitalares, proporcionando assim uma resposta rápida e eficiente aos casos de menor complexidade clínica.

A implantação das UPAs começou a partir de 2003, tendo seu funcionamento baseado na Política Nacional de Urgência e Emergência assim como na Política Nacional de Humanização do Ministério da Saúde, sendo parte integrante da Rede de Atenção às Urgências e estão inseridas no SUS (BRASIL, 2023).

Uma UPA deve fornecer, conforme constam nas diretrizes de modelo assistencial e financiamento de UPA 24h de Pronto Atendimento, atendimento ininterrupto 24 horas e em todos os dias da semana, incluindo feriados e pontos facultativos, acolhimento de pacientes, classificação de risco e equipe Assistencial Multiprofissional com quantitativo de profissionais compatível com a necessidade de atendimento (BARROS, 2017).

As UPAs variam em seus portes de acordo com a quantidade de habitantes da cidade onde estão localizadas (Tabela 1), visando assim atender a demanda específica de pacientes em cada localidade (Tabela 2).

Tabela 1 - Porte da UPA 24h

Porte	População	Nº mínimo de leitos de observação	Nº mínimo de leitos na sala de emergência
PORTE I	50.000 a 100.00 habitantes	7 leitos	2 leitos
PORTE II	100.001 a 200.000 habitantes	11 leitos	3 leitos
PORTE III	200.001 a 300.000 habitantes	15 leitos	4 leitos

Fonte: Adaptado de BRASIL, 2017

Tabela 2 - Média de atendimentos na UPA 24h

Porte	Nº de atendimentos médicos em 24 horas	Nº mínimo de médicos das 7hs às 17hs	Nº mínimo de médicos das 19hs às 7hs
PORTE I	Média de 150 pacientes	2 médicos	2 médicos
PORTE II	Média de 250 pacientes	4 médicos	2 médicos
PORTE III	Média de 350 pacientes	6 médicos	3 médicos

Fonte: Adaptado de BRASIL, 2017

Segundo consta na portaria nº 2048 de 05 de novembro de 2002 (Brasil, 2002), os profissionais essenciais para o funcionamento de uma UPA são os seguintes:

As Unidades Não-Hospitalares de Atendimento às Urgências e Emergências deverão contar, obrigatoriamente, com os seguintes profissionais: coordenador ou gerente, médico clínico geral, médico pediatra, enfermeiro, técnico/auxiliar de enfermagem, técnico de radiologia, auxiliar de serviços gerais, auxiliar administrativo e, quando houver laboratório na unidade, também deverão contar com bioquímico, técnico de laboratório e auxiliar de laboratório. Outros profissionais poderão compor a equipe, de acordo com a definição do gestor local ou gestores loco-regionais, como: assistente social, odontólogo, cirurgião geral, ortopedista, ginecologista, motorista, segurança e outros.

Para propiciar um atendimento e atenção adequados aos pacientes, uma UPA deve dispor de uma série de elementos físicos e também organizacionais para que o funcionamento adequado da unidade seja possível (Tabela 3).

Tabela 3 - Infraestrutura da UPA 24h

Ambiente	Descrição
Recepção	Área onde os pacientes fazem o registro inicial, fornecem informações básicas sobre suas condições de saúde e passam por uma triagem rápida para determinar a gravidade de sua condição.
Consultórios Médicos	Espaços para consultas médicas e de enfermagem, onde os pacientes são avaliados, tratados e encaminhados conforme necessário. Pode haver consultórios especializados para diferentes áreas médicas.
Sala de Observação	Espaço onde os pacientes podem ser monitorados por um período após o atendimento inicial, especialmente se houver dúvidas sobre o diagnóstico ou a necessidade de internação.
Sala de Emergência	Área destinada ao atendimento de pacientes em estado mais grave, que requerem cuidados imediatos. Essa área é equipada com recursos para reanimação, estabilização e suporte de vida.
Sala de Raios-X	Equipamentos para a realização de radiografias.
Laboratório	Local para a realização de exames laboratoriais rápidos, como análises de sangue e urina, para auxiliar no diagnóstico.
Farmácia	Espaço onde os medicamentos prescritos pelos médicos podem ser dispensados aos pacientes.

Fonte: Brasil, 2023

2.2 Acolhimento com Classificação de Risco

A rotina de atendimento de uma UPA funciona seguindo as normas de Acolhimento com Classificação de Risco (ACR), onde cada paciente é atendido de acordo com o grau de gravidade do seu estado de saúde e não pela sua ordem de chegada, sendo atribuído a cada paciente uma cor específica.

A classificação dos pacientes em uma UPA é baseada no Sistema de Triagem Manchester (STM), que utiliza cinco cores para identificar o grau de cada paciente. Geralmente, elas são: vermelho, laranja, amarelo, verde e azul (Tabela 4). A cor vermelha representa os casos mais graves, e a azul, os mais leves (BRASIL, 2022). Essas cores representam um nível de prioridade e urgência, permitindo que a equipe médica concentre-se primeiro nos pacientes que precisam de atendimento imediato.

Tabela 4 - Tempo de atendimento ACR

Cor	Prioridade	Tempo estimado
Vermelho	Emergência	Atendimento imediato
Laranja	Muito urgente	10 minutos
Amarelo	Urgência moderada	60 minutos
Verde	Pouco urgente	120 minutos
Azul	Não urgente	240 minutos

Fonte: Elaborado pelo autor (2023)

Figura 1 - Acolhimento com Classificação de Risco

Fonte: Prefeitura de São Mateus do Sul (2019)

Segundo consta nas normas de Acolhimento e Classificação de Risco nos Serviços de Urgência (2009, p. 24, 25):

A classificação de risco é uma ferramenta que, além de organizar a fila de espera e propor outra ordem de atendimento que não a ordem de chegada, tem também outros objetivos importantes, como: garantir o atendimento imediato do usuário com grau de risco elevado; informar o paciente que não corre risco imediato, assim como a seus familiares, sobre o tempo provável de espera; promover o trabalho em equipe por meio da avaliação contínua do processo; dar melhores condições de trabalho para os profissionais pela discussão da ambiência e implantação do cuidado horizontalizado; aumentar a satisfação dos usuários e, principalmente, possibilitar e instigar a pactuação e a construção de redes internas e externas de atendimento.

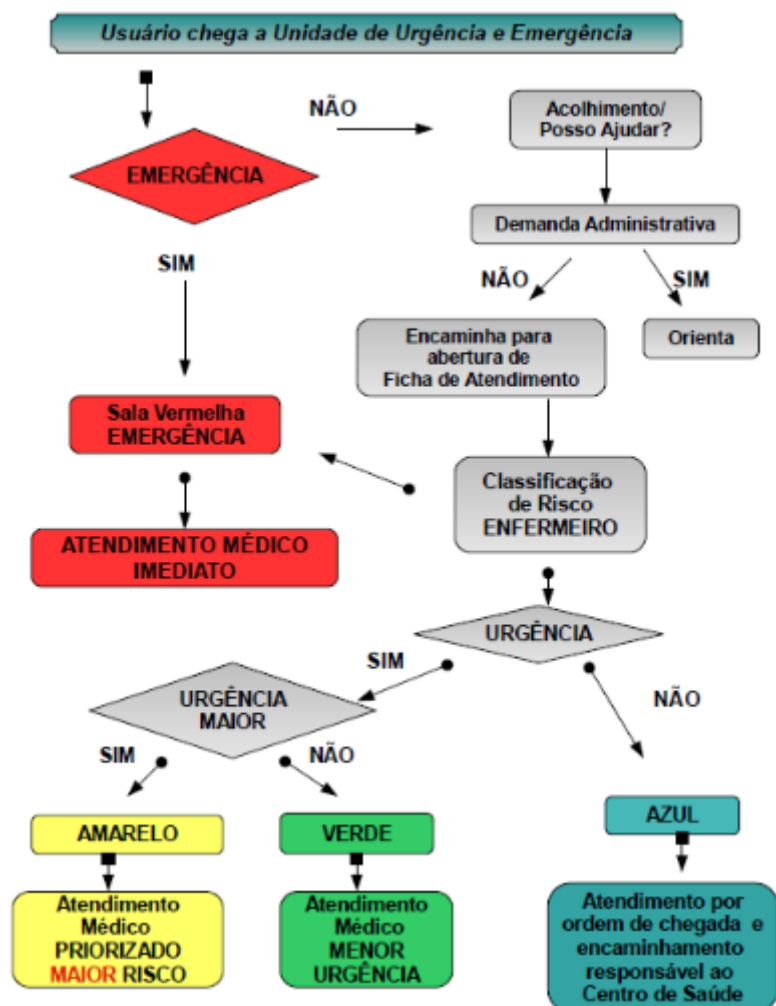
O acolhimento é geralmente realizado por algum profissional de saúde, como enfermeiros, técnicos de enfermagem, e em alguns casos podendo ser feito até mesmo pelos médicos. Esses profissionais são treinados para receber os pacientes, avaliar suas condições médicas e determinar a gravidade do caso. No entanto, a Classificação de Risco é competência privativa do profissional de enfermagem.

Nesse sentido, declara Feitosa et al. (2017, 138):

O acolhimento pode ser realizado por qualquer profissional de saúde, mas cabe ao enfermeiro por meio da Consulta de Enfermagem realizar a Classificação de Risco, com base num protocolo predefinido com cores que caracterizam a gravidade ou agravamento do paciente (vermelho: emergência; amarelo: urgência; verde: pouco urgente; e azul: não urgente), o que designa a ordem do atendimento. É importante que essa organização seja divulgada com clareza para os usuários.

De acordo com o estado de saúde apresentado pelo paciente quando da sua chegada a UPA, seu atendimento pode ser imediato, se for uma emergência ou então irá ser atendido de acordo com a gravidade de seu quadro clínico, não fazendo jus, neste caso, a um atendimento imediato.

Figura 2 - Fluxograma das etapas de atendimento do ACR



Fonte: Secretaria de Estado de Saúde do Distrito Federal (2012)

2.2 SISTEMAS DE INFORMAÇÃO

De forma simplificada, um Sistema de Informação é um sistema capaz de fornecer informações para a tomada de decisões em uma organização, sendo projetado para coletar, processar, armazenar e disponibilizar dados importantes aos gestores e outros profissionais responsáveis por tomar decisões estratégicas, táticas e operacionais em uma empresa. Entre as principais atividades desempenhadas por um SI estão a entrada de dados, processamento de dados e a saída de dados.

De acordo com O'Brien (2004), um sistema pode ser entendido como um grupo de elementos inter-relacionados ou em constante interação que formam um todo unificado. Um sistema desse tipo possui três componentes ou funções básicas em interação:

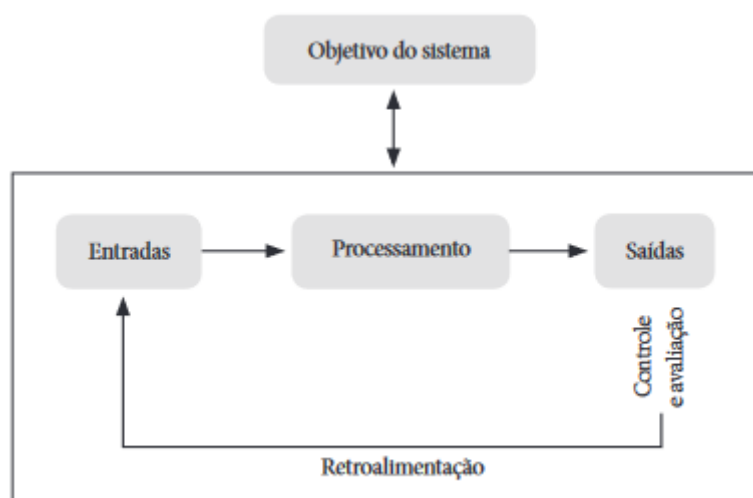
Entradas: envolve a captação e agrupamento de elementos que ingressam no sistema a fim de serem processados (dados, instruções).

Processamento: envolve processos de transformação que convertem insumos (entradas) em produtos (programas, equipamentos).

Saídas: envolve a transferência de elementos produzidos por um processo de transformação até seu destino final (relatórios, gráficos, cálculos).

Um sistema opera por meio do processamento de recursos, que são as entradas que o sistema recebe. Esse processamento resulta na geração de saídas ou produtos do sistema, que incluem as entradas, o processo pelo qual elas são transformadas e as saídas resultantes desse processo (Padoveze, 2015).

Figura 3 - Funcionamento de um Sistema

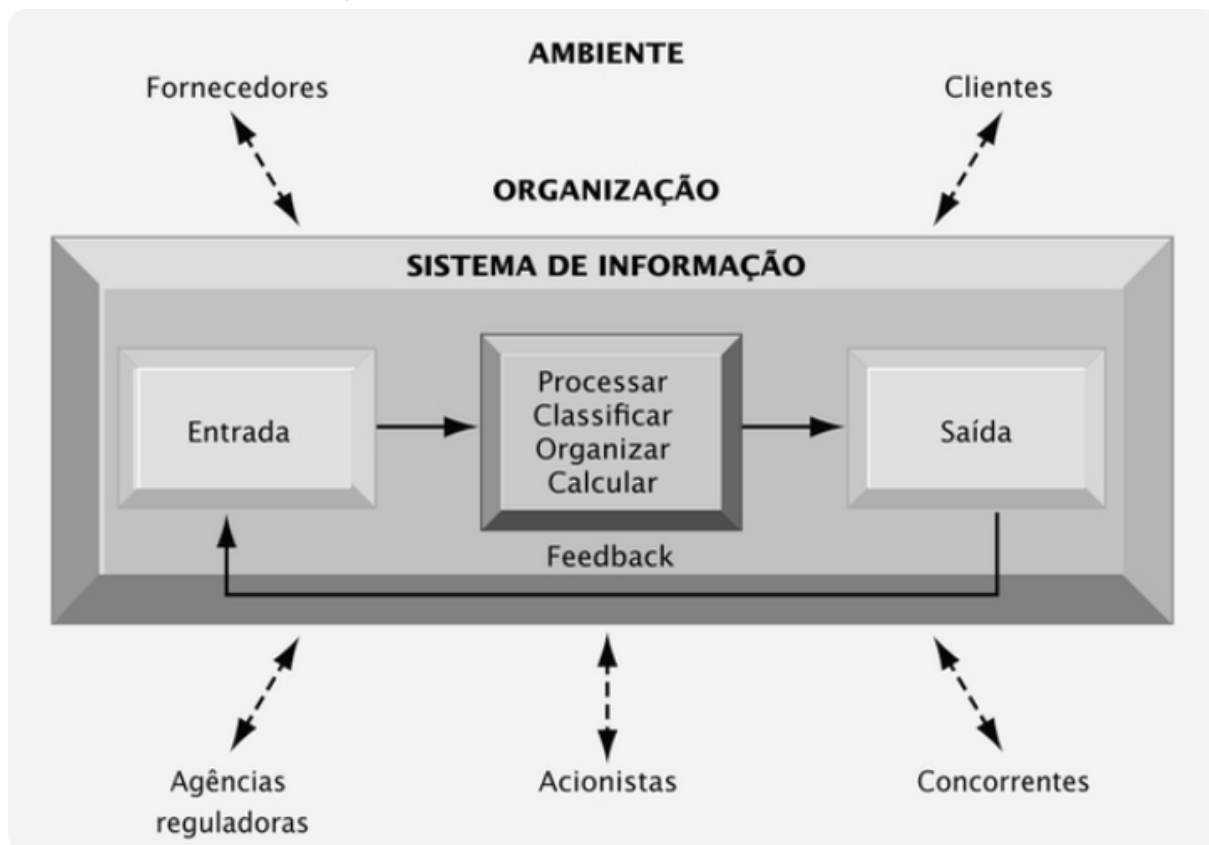


Fonte: Padoveze (2013)

Nesse sentido, também declara Laudon e Laudon (2006, p. 31) o seguinte:

A entrada captura ou coleta dados brutos de dentro da organização ou de seu ambiente externo. O processamento converte esses dados brutos em uma forma mais significativa. A saída transfere as informações processadas às pessoas que as utilizarão ou às atividades nas quais elas serão empregadas.

Figura 4 - Atividades de um Sistema de Informação



Fonte: Laudon e Laudon (2007)

Com relação aos elementos que um Sistema de Informação deve possuir (Tabela 5), pode-se citar segundo Kroenke (2012) os seguintes:

Tabela 5 - Elementos de um Sistema de Informação

Elemento	Descrição
Hardware	São os componentes físicos do sistema, como computadores, servidores, dispositivos de armazenamento, redes, impressoras e outros equipamentos relacionados ao processamento de dados
Software	São os programas e aplicativos que permitem aos usuários realizar tarefas específicas. Isso inclui sistemas operacionais, aplicativos de software personalizados e software de produtividade.

Dados	Os dados são a matéria-prima do sistema de informação. Eles representam fatos brutos que podem ser transformados em informações significativas. Os dados podem ser estruturados (como informações em bancos de dados) ou não estruturados (como documentos de texto ou mídia).
Redes	São componentes críticos dos sistemas de informação modernos. Elas permitem a comunicação e o compartilhamento de informações entre diferentes dispositivos e sistemas. Isso inclui hardware de rede (como roteadores e switches) e software de rede (como protocolos de comunicação).
Pessoas	São uma parte crítica de qualquer sistema de informação. Isso inclui os usuários finais que interagem com o sistema, bem como os profissionais de TI que o desenvolvem, mantêm e suportam.

Fonte: Adaptado de Kroenke (2012)

2.2.1 Tipos de Sistemas de Informação

Existem diversos tipos de Sistemas de Informação que desempenham papéis importantes no ambiente empresarial e organizacional. Esses sistemas são projetados para coletar, processar, armazenar e fornecer informações relevantes para apoiar as atividades de uma organização. Esses sistemas podem ser classificados de diversas formas, como níveis organizacionais, áreas funcionais ou tipos de suporte que fornecem a uma organização.

Segundo Turban; McLean; Wetherbe (2004) os principais tipos de Sistemas de Informação segundo o tipo de suporte oferecido correspondem a (Tabela 6):

Tabela 6 - Tipos de Sistema de Informação

Tipo de Sistema	Sigla	Funcionamento
Sistema de Suporte Inteligente	SSI	Esse tipo de sistema utiliza tecnologias avançadas de inteligência artificial para ajudar a fornecer suporte a usuários de diversas maneiras. É projetado para entender as necessidades do usuário e fornecer soluções relevantes, respostas precisas e assistência personalizada em tempo real
Sistema de Apoio a Grupos	GSS	Esse tipo de sistema é projetado para facilitar a comunicação, colaboração e tomada de decisões em grupos ou equipes de pessoas. Esses sistemas são especialmente úteis quando as decisões precisam ser tomadas por um grupo de indivíduos com habilidades, conhecimentos e perspectivas diferentes.
Sistema de Informação Empresarial	SIE	Este tipo de sistema é projetado para coletar, armazenar, processar e disseminar informações dentro de uma organização, com o objetivo de apoiar a tomada de decisões e melhorar o desempenho geral do negócio.
Sistema de Apoio a Decisões	SAD	Este tipo de sistema de informação é projetado para auxiliar indivíduos e organizações na tomada de decisões complexas, fornecendo informações, análises e ferramentas interativas para ajudar a resolver problemas e avaliar alternativas. Esses sistemas são desenvolvidos para lidar com decisões que envolvem grande quantidade de dados, incerteza, ambiguidade e múltiplos critérios.
Sistema de Informação Gerencial	SIG	Este tipo de sistema fornece informações para apoiar a tomada de decisões de nível médio e operacional. Eles coletam dados de vários sistemas internos e externos, os processam e os apresentam em relatórios e painéis que auxiliam os gerentes na tomada de decisões estratégicas.

Fonte: adaptado de Turban; McLean; Wetherbe (2004)

Outros sistemas correspondem ao **Sistema de automação de escritório (SAE)**, **Sistema de administração do conhecimento (KMS)** e **Sistema de processamento de transações (SIT)**.

2.2.1.1 Sistema de Informação Gerencial

Um Sistema de Informação Gerencial (SIG) é uma ferramenta que ajuda as organizações a coletar, processar, armazenar e analisar dados para facilitar a tomada de decisões eficazes e informadas em todos os níveis da empresa. Ele desempenha um papel fundamental na gestão de negócios, fornecendo informações relevantes e oportunas para os gestores e tomadores de decisão. Dentre as principais características de um Sistema de Informação Gerencial pode-se citar a coleta de dados, processamento e organização de dados, análise e geração de relatórios, compartilhamento de informações, apoio à tomada de decisões, integração de processos, segurança e privacidade.

Trata-se de um sistema voltado a coletar, armazenar, recuperar e processar as informações necessárias ou demandadas pelos líderes na concretização de suas responsabilidades. Esse sistema engloba a transformação de dados em informações que se tornam pertinentes para o processo de tomada de decisão organizacional, oferecendo apoio administrativo essencial para aprimorar os resultados almejados (BATISTA, 2012).

Um SIG engloba tanto a criação como a utilização de sistemas de informação que forneçam auxílio para as organizações atingirem suas metas e objetivos empresariais (KROENKE, 2012).

De acordo com Batista (2012, p. 25) um SIG é projetado da seguinte maneira:

Eles são desenvolvidos com base no conhecimento específico do negócio, chamado de método de trabalho. E, com uma metodologia adequada para selecionar informações estratégicas e atividades críticas da empresa, os sistemas de informação gerencial utilizam ferramentas que permitem a visão analítica dos dados/processos, gerando visão agregada, integrada e gráfica dos principais indicadores de desempenho da empresa.

Dentre os benefícios que uma organização pode obter por meio do uso de um SIG, podem ser citados alguns deles, conforme apontados por Oliveira (2008): facilidade de acesso às informações, melhoria na tomada de decisão, aumento da produtividade e geração de relatórios para análise.

Em suma, um SIG representa uma peça fundamental para o sucesso e eficácia das empresas. Sua capacidade de coletar, processar, organizar e analisar dados com o intuito de fornecer informações relevantes para a tomada de decisões permeia todas as camadas hierárquicas da empresa.

2.2.1.2 Sistema de Informação em Saúde

Um Sistema de Informação em Saúde (SIS) consiste em um sistema desenvolvido para gerenciar dados relacionados à saúde. Isso inclui sistemas projetados para coletar, armazenar, gerenciar e transmitir o prontuário médico eletrônico (PEP) de um paciente, e até o gerenciamento operacional de um hospital ou unidade de saúde por completo.

De acordo com Marin (2010, p. 21) um SIS pode ser definido como:

[...] um conjunto de componentes inter-relacionados que coletam, processam, armazenam e distribuem informação para apoiar o processo de tomada de decisão e auxiliar no controle das organizações de saúde. Assim, os sistemas de informação em saúde congregam um conjunto de dados, informações e conhecimento utilizados na área de saúde para sustentar o planejamento, o aperfeiçoamento e o processo decisório dos múltiplos profissionais da área da saúde envolvidos no atendimento aos pacientes e usuários do sistema de saúde.

Também a Organização Mundial de Saúde (OMS) define um SIS da seguinte maneira:

Um conjunto de componentes que atuam de forma integrada por meio de mecanismos de coleta, processamento, análise e transmissão da informação necessária e oportuna para implementar processos de decisões no Sistema de Saúde. Seu propósito é selecionar dados pertinentes e transformá-los em informações para aqueles que planejam, financiam, provêm e avaliam os serviços de saúde.

Muitas são as informações e os dados coletados por um SIS, encontrando-se nesse cenário a Informação Clínica (IC). Esta pode ser entendida como qualquer informação que os profissionais da área de saúde como médicos e enfermeiros produzem no seu dia a dia de trabalho, bem como as informações por eles produzidas. Trata-se de um tipo de informação que possui o propósito de prover uma melhor assistência médica aos pacientes, tendo que possuir desse modo alta qualidade e efetividade (GALVÃO, 2012).

Em relação às funções que um SIS deve desempenhar, pode-se citar segundo Ferreira (1999) as seguintes:

- Organizar a produção de informações compatíveis com as necessidades dos diferentes níveis, garantindo uma avaliação permanente das ações executadas e do impacto destas sobre a situação de saúde;

- Assessorar o desenvolvimento de sistemas voltados para as especificidades das diferentes unidades operacionais do sistema de saúde;
- Contribuir para o desenvolvimento dos profissionais de saúde, para a construção de uma consciência sanitária coletiva, como base para ampliar o exercício do controle social e da cidadania. Também para resgatar uma relação mais humana entre a instituição e o cidadão.

De forma simplificada, um SIS desempenha um papel essencial no aprimoramento da qualidade, eficiência e também eficácia do atendimento médico, tornando mais fácil realizar pesquisas, fornecer evidências e apoiar o ensino na área da saúde (MARIN, 2010).

3 SISTEMAS SIMILARES

Vários são os sistemas para a administração e gerência de hospitais e unidades de saúde disponíveis. No entanto, encontrar um sistema criado especialmente para atender às exigências de uma Unidade de Pronto Atendimento não se mostra uma tarefa simples. Neste trabalho, optou-se por listar os principais sistemas de informação atualmente disponíveis para a área de saúde no Brasil, com o intuito de delinear seu escopo.

3.1 Philips Tasy

O Philips Tasy é um sistema de gestão hospitalar desenvolvido pela Philips, uma das empresas líderes em tecnologia e saúde. Esse sistema é voltado para hospitais e instituições de saúde, visando otimizar a administração e operação de diversos aspectos relacionados à prestação de cuidados médicos. O Philips Tasy é conhecido por sua abordagem completa e integrada para a gestão hospitalar.

Figura 5 - Sistema Philips Tasy

Status	Nível de urgência	Status paciente	Paciente	Hora entrada	Idade	Motivo atendimento	Tipo de prior
<input type="checkbox"/>	COVID - 19	Fim triagem	Super Homem Teste	27/07/2021 14:11:51	19a 5m 25d	Abcesso	CHAMADA
<input type="checkbox"/>	Emergência - Vermelho	Prescrição liberada	Antonio de Jesus da Luz	27/07/2021 10:31:50	45a 9m 26d	Dor em membros	
<input type="checkbox"/>	Emergência - Vermelho	Fim triagem	Maria Teste	27/07/2021 14:00:51	2a 1m 26d	Abcesso	PRIORIDADE
<input type="checkbox"/>	Urgência - Amarelo	Prescrição liberada	Vanderleia Santana Carvalho	27/07/2021 08:53:57	35a 8m 27d	Dor nas Costas	CHAMADA
<input type="checkbox"/>	Urgência - Amarelo	Fim consulta	Edison Luiz Pedroso da Rosa	27/07/2021 09:24:47	61a 3m 29d	Dor no peito	CHAMADA
<input type="checkbox"/>	Urgência - Amarelo	Fim consulta	Efrida dos Santos Antunes	27/07/2021 10:35:18	64a 11m 21d	Outras queixas oculares	
<input type="checkbox"/>	Urgência - Amarelo	Prescrição liberada	Raphael Martins Ramos	27/07/2021 10:37:46	33a 6m 2d	Febre	
<input type="checkbox"/>	Urgência - Amarelo	Fim triagem	Eloiza Vaz Machado	27/07/2021 11:46:46	16a 3m 15d	Tosse	
<input type="checkbox"/>	Urgência - Amarelo	Fim consulta	Marcelo Cristiano Silveira Carvalho	27/07/2021 13:02:04	44a 4m 23d	Traumas em Geral	
<input type="checkbox"/>	Urgência - Amarelo	Prescrição liberada	Andreia Santana de Avila	27/07/2021 13:11:26	40a 11m 17d	Crise de Coluna	
<input type="checkbox"/>	Urgência - Amarelo	Fim consulta	Bruno de Almeida Alves	27/07/2021 13:32:32	26a 7m 6d	Traumas em Geral	
<input type="checkbox"/>	Urgência - Amarelo	Prescrição liberada	Viviane de Fatima Weber Valentim	27/07/2021 13:39:07	30a 5m 21d	Dor nas Costas	
<input type="checkbox"/>	Urgência - Amarelo	Prescrição liberada	Ivone de Aparecida Dobzinski	27/07/2021 13:46:10	48a 1m 28d	Dor em membros	
<input type="checkbox"/>	Urgência - Amarelo	Prescrição liberada	Dayane Jesus Martins	27/07/2021 14:00:20	21a 5m 4d	Colica Renal	

Fonte: <https://www.philips.com.br/healthcare/resources/landing/solucao-tasy> (2023)

Algumas das características e funcionalidades do Philips Tasy incluem:

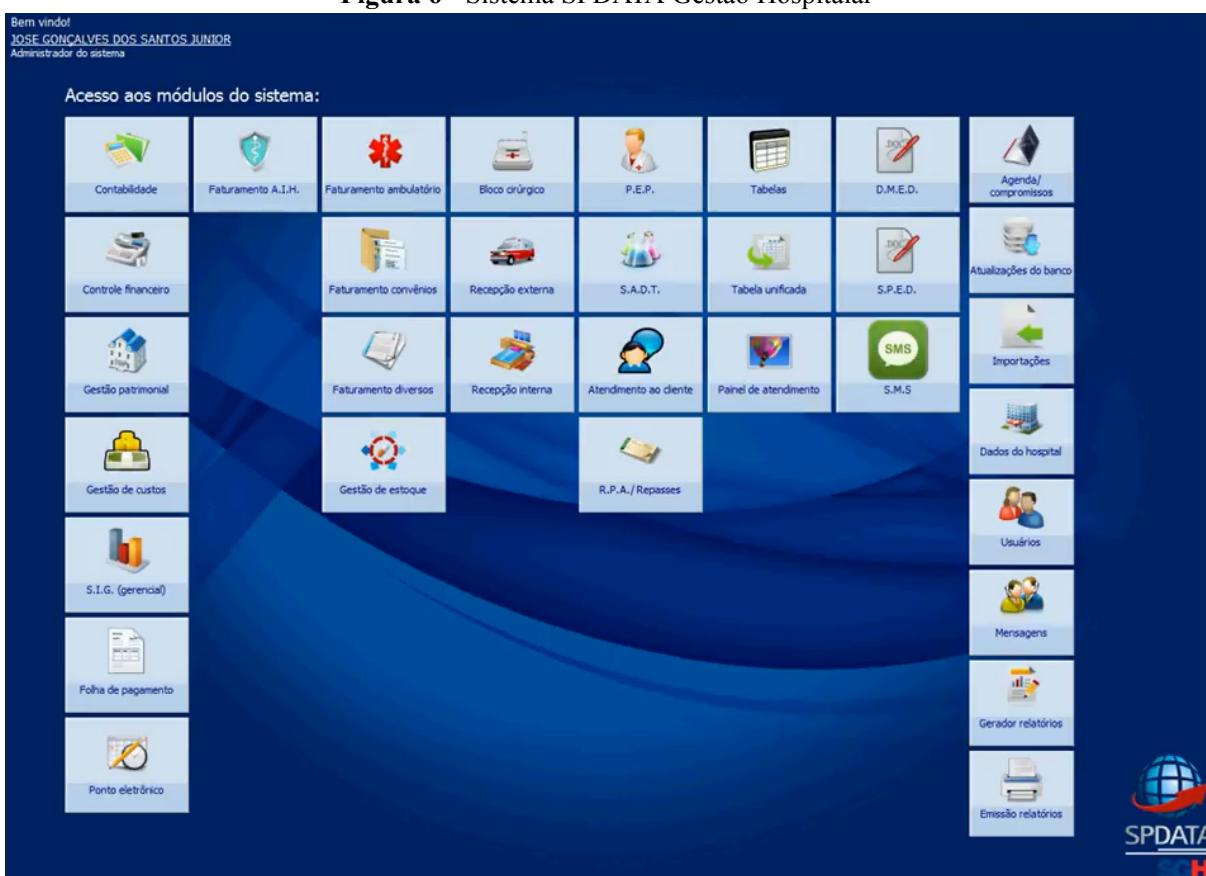
- Gestão de Leitos
- Prontuário Eletrônico
- Prescrição Eletrônica

- Agendamento e Atendimento
- Integração com Equipamentos Médicos
- Gestão Financeira e Faturamento
- Indicadores e Relatórios
- Controle de Estoques

3.2 SPDATA Gestão Hospitalar

O SPDATA Gestão Hospitalar é um sistema de gestão hospitalar desenvolvido pela SPDATA, uma empresa brasileira especializada em soluções de tecnologia da informação para a área da saúde. Esse sistema foi projetado para atender às necessidades específicas de hospitais, clínicas e instituições de saúde em geral. A principal finalidade do SPDATA Gestão Hospitalar é proporcionar uma gestão mais eficiente e integrada das operações hospitalares, abrangendo diversos aspectos da administração e atendimento ao paciente.

Figura 6 - Sistema SPDATA Gestão Hospitalar



Fonte: https://www.youtube.com/watch?v=mHhb0_6ESSs (2015)

A tabela 7 apresenta as principais funcionalidades do deste sistema:

Tabela 7 - Funcionalidades do SPDATA Gestão Hospitalar

Funcionalidade	Descrição
Prontuário Eletrônico do Paciente (PEP)	O sistema permite o registro e acesso eletrônico às informações dos pacientes, incluindo histórico médico, exames, prescrições médicas e outras informações clínicas. Isso ajuda a melhorar a qualidade do atendimento ao paciente e a reduzir erros médicos.
Agendamento de Consultas e Procedimentos	Facilita o agendamento de consultas médicas, procedimentos cirúrgicos e exames, garantindo uma distribuição eficiente de recursos e evitando filas e atrasos.
Gestão Financeira	Permite o controle das finanças hospitalares, incluindo faturamento, cobrança, contas a pagar e a receber, além de integrar com sistemas de convênios e planos de saúde.
Gestão de Estoque	Ajuda a gerenciar o estoque de medicamentos, materiais médicos e outros suprimentos hospitalares, garantindo o abastecimento adequado e evitando desperdícios.
Controle de Leitos e Internações	Facilita o gerenciamento de leitos hospitalares, permitindo o acompanhamento das internações, alta de pacientes e a disponibilidade de leitos.
Relatórios e Indicadores	Oferece recursos para a geração de relatórios gerenciais e indicadores de desempenho que auxiliam na tomada de decisões estratégicas.

Fonte: Adaptado de spdata.com.br/solucoes/hospitais/controle-assistencial (2023)

4 METODOLOGIA E PROCESSO DE DESENVOLVIMENTO DE *SOFTWARE*

4.1 *SOFTWARE*

O *software* desempenha um papel extremamente importante na sociedade moderna, estando ele presente em praticamente todos os aspectos da nossa vida. Ele consiste em instruções lógicas que controlam o funcionamento de dispositivos eletrônicos e computadores, fornecendo assim comunicação e interação.

De acordo com Pressman (2016, p. 29) o *software* de computador é compreendido como:

[...] o produto que profissionais de *software* desenvolvem e ao qual dão suporte no longo prazo. Abrange programas executáveis em um computador de qualquer porte ou arquitetura, conteúdos (apresentados à medida que os programas são executados), informações descritivas tanto na forma impressa (*hard copy*) como na virtual, abrangendo praticamente qualquer mídia eletrônica.

Devido ao fato de que o software é um meio digital, ou seja, ele é composto por instruções que serão interpretadas pelos sistemas computacionais, não existem muitas restrições a seu desenvolvimento. Desse modo, o processo de engenharia de software se torna mais simples, mas também complexo devido a facilidade de desenvolvimento (SOMMERVILLE, 2011).

4.2 METODOLOGIAS DE DESENVOLVIMENTO DE *SOFTWARE*

Para desenvolver um *software*, são necessárias diversas técnicas, uma vez que diferentes tipos de sistemas podem exigir abordagens distintas. Não há um método definitivo que se aplique a todos os casos, pois cada projeto tem suas particularidades e desafios únicos. Portanto, é crucial compreender as necessidades específicas do software a ser criado e selecionar as técnicas mais adequadas para atingir os objetivos desejados.

Desenvolver *software* de qualidade é uma tarefa trabalhosa e de grande importância, envolvendo várias etapas, como análise de requisitos, design, codificação, teste e implantação. A respeito disso, declara Pressman (2016, p. 7) o seguinte:

Quando um software é bem-sucedido — atende às necessidades dos usuários, opera perfeitamente durante um longo período, é fácil de modificar e, mais fácil ainda, de utilizar —, ele é realmente capaz de mudar as coisas para melhor. Porém, quando um software falha — quando seus usuários estão insatisfeitos, quando é propenso a

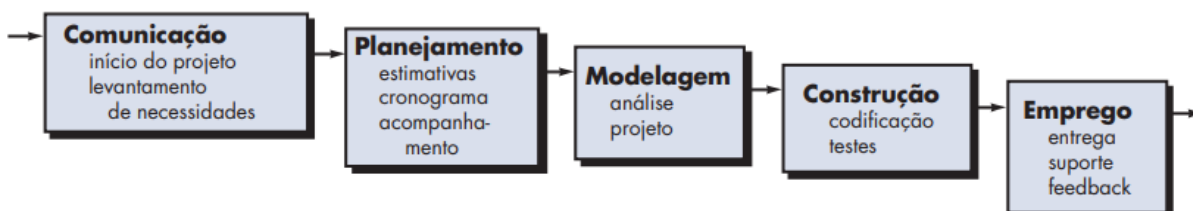
erros, quando é difícil modificá-lo e mais difícil ainda utilizá-lo —, fatos desagradáveis podem e, de fato, acontecem.

4.2.1 Metodologias Tradicionais

As metodologias tradicionais de desenvolvimento de software surgiram a partir das décadas de 1960 e 1970, quando as empresas começaram a perceber a necessidade de abordagens mais estruturadas para gerenciar projetos de software complexos. Nesse período, o desenvolvimento de software estava se expandindo rapidamente e as equipes enfrentavam desafios em termos de organização, planejamento, controle de qualidade e entrega pontual.

Em geral, as abordagens tradicionais de desenvolvimento de software são caracterizadas por seguir etapas sequenciais e enfatizar a criação detalhada de documentação ao longo do processo. A metodologia tradicional mais famosa e mais antiga da Engenharia de Software é o modelo cascata, que consiste em um processo sequencial de desenvolvimento de software, dividido em etapas lineares, incluindo análise, design, implementação, testes, implantação e manutenção (Figura 7). Cada fase depende da conclusão da anterior, resultando em pouca flexibilidade para ajustes após o início do projeto.

Figura 7 - Modelo de desenvolvimento em Cascata



Fonte: Pressman (2011)

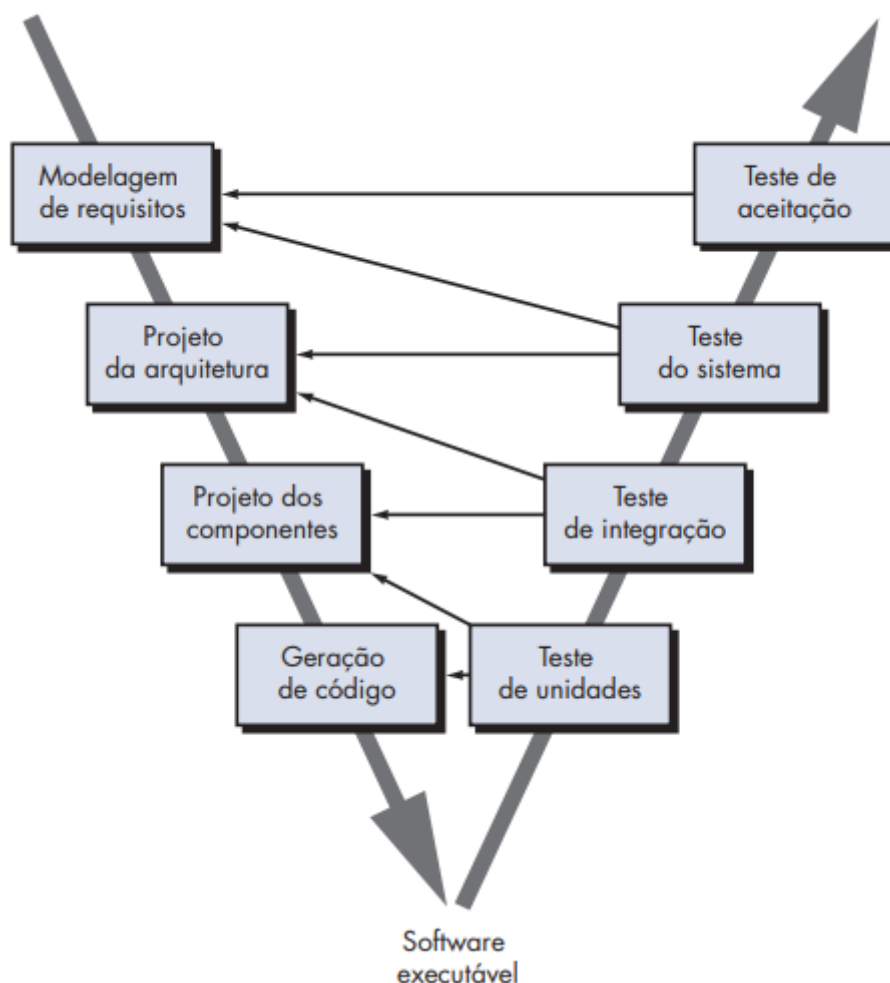
A respeito do modelo cascata, declara Sommerville (2011, p.20) o seguinte:

Por causa do encadeamento entre uma fase e outra, esse modelo é conhecido como “modelo em cascata”, ou ciclo de vida de software. O modelo em cascata é um exemplo de um processo dirigido a planos — em princípio, você deve planejar e programar todas as atividades do processo antes de começar a trabalhar nelas.

Além do modelo cascata, também pode-se citar o modelo em V (Figura 8). Este modelo é uma abordagem de desenvolvimento de software que segue uma estrutura semelhante ao modelo cascata, mas enfatiza a relação entre as fases de desenvolvimento e teste de forma

mais explícita. Cada fase de desenvolvimento (como análise, projeto, codificação) tem uma fase correspondente de teste (como testes de unidade, testes de integração, testes de sistema) associada a ela. As fases de desenvolvimento são representadas no lado esquerdo do "V", enquanto as fases de teste estão no lado direito, formando uma espécie de "V".

Figura 8 - Modelo de desenvolvimento em V



Fonte: Pressman (2011)

4.2.2 Metodologias Ágeis

Atualmente, os *softwares* são uma parte indispensável de praticamente toda a infraestrutura global, infraestrutura esta que está sempre em constante evolução. Devido a esse fato, se torna imprescindível que as empresas acompanhem esse ritmo e respondam rapidamente às mudanças, o que requer a entrega ágil de novos softwares. O desenvolvimento e entrega rápidos são, portanto, o requisito mais crítico para o desenvolvimento de sistemas de software (SOMMERVILLE, 2011).

As metodologias ágeis se destacam das metodologias de desenvolvimento tradicionais por sua abordagem iterativa, que ocorre de forma contínua ao longo do processo de desenvolvimento de software, em vez de se concentrar apenas no início do projeto. Dessa forma se torna possível obter funcionalidades do *software* entregues de forma rápida e dinâmica, preenchendo uma das principais fraquezas presente nas metodologias de desenvolvimento tradicionais.

Segundo afirma Goyal (2008, p. 3) uma das principais fraquezas das metodologias de desenvolvimento tradicionais reside no fato de que:

No modelo tradicional de desenvolvimento de software em cascata, o projeto inteiro é dividido em várias etapas: coleta de requisitos do usuário, design e documentação, desenvolvimento, testes e implantação. Nessa abordagem, presume-se que cada etapa esteja 100% concluída antes que a próxima comece. Uma das principais fraquezas dessa abordagem é que os erros de design frequentemente só são descobertos durante a implantação. Nesse momento, o projeto está quase completo e os erros são frequentemente caros de serem corrigidos.

Uma das grandes mudanças no paradigma das metodologias tradicionais para as metodologias ágeis ocorreu por meio do advento do Manifesto Ágil, que foi publicado em 2001 intitulado como *Manifesto for Agile Software Development*, contendo uma série de valores e princípios que norteiam o desenvolvimento ágil de *software*.

Figura 9 - Valores da Metodologia Ágil

Fonte: www.sankhya.com.br/blog/metodologias-ageis (2022)

Dentre as metodologias ágeis mais utilizadas pode-se citar a FDD, Scrum, Kanban, *Extreme Programming* (XP) e *Lean Software Development*, todas estas baseadas nos valores e princípios do Manifesto Ágil.

4.2.2.1 *Feature Driven Development*

Na concepção deste projeto, optou-se por empregar como metodologia de desenvolvimento de software o Desenvolvimento Dirigido a Funcionalidades, do inglês *Feature Driven Development* (FDD). Embora originalmente projetada para equipes de vários integrantes, caracterizando-se como uma metodologia ágil, sua aplicação foi adaptada para se adequar às necessidades de uma equipe menor, como é o caso deste trabalho.

Esta metodologia é baseada em um processo de desenvolvimento de *software* ágil e altamente adaptável que tem como principais características ser altamente curto e iterativo, enfatizando a qualidade em todas as etapas, entregando assim resultados frequentes e tangíveis (GOYAL, 2008).

Consiste em uma metodologia ágil, robusta e muito utilizada nos dias atuais, existindo inúmeras vantagens na sua utilização. Por exemplo, essa metodologia proporciona benefícios a gerentes, desenvolvedores e clientes, atendendo equipes pequenas, médias ou grandes. É possível obter um *software* de qualidade bem como acompanhar o progresso do desenvolvimento do projeto (SBROCCO E MACEDO, 2012).

De acordo com Koppensteiner, Sonja & Udo, Nathalie (2003) a FDD:

É baseada em processos repetíveis, colocando assim mais ênfase na modelagem do que outras metodologias ágeis. Os projetos começam com um modelo esquelético que é continuamente atualizado à medida que o *design* amadurece. Os recursos são definidos como funcionalidades de negócios que podem ser entregues em incrementos de duas semanas ou menos.

As funcionalidades de negócios, correspondem a uma função valorizada pelo cliente e passível de ser implementada em duas semanas ou menos. Uma funcionalidade deve ser pequena, facilitando assim sua inspeção e revisão. Além disso, as funcionalidades podem ser organizadas em forma hierárquica, guiando e auxiliando no planejamento, cronograma e acompanhamento do projeto. Para a escrita de uma funcionalidade, ou conjunto de funcionalidades, deve-se usar um modelo que consiste em **<ação> o <resultado> <por | para quem | de | para que> um <objeto>**, onde <objeto> é uma pessoa, local ou coisa (PRESSMAN, 2016). No caso de um sistema para gerenciamento de uma UPA poderíamos ter como funcionalidades: Cadastre o paciente no sistema, Mostre as informações do paciente e Imprima o prontuário do paciente.

No processo da FDD, existe uma ênfase nas atividades de garantia da qualidade de *software* por meio do encorajamento de uma estratégia de desenvolvimento incremental, do uso de inspeções do código e do projeto, da aplicação de auditorias para garantia da qualidade de software, da coleta de métricas e do uso de padrões (PRESSMAN, 2016).

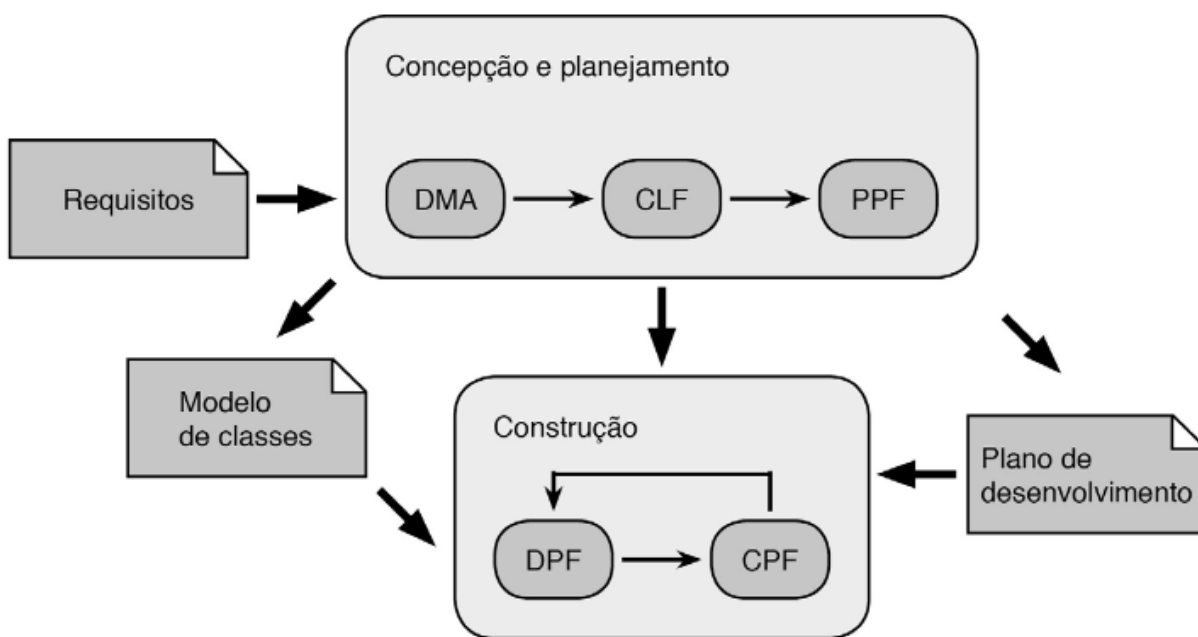
A metodologia FDD é indicada principalmente para novos projetos de *Software*, onde o aprimoramento e atualização de código é constante. Também é indicada para projetos onde necessita-se de novas versões de um aplicativo existente. As empresas/organizações devem

adotar o método de forma gradativa em pequenos pedaços e finalmente em sua totalidade à medida que o projeto de desenvolvimento avança (PALMER E FELSING, 2003).

Os criadores desta metodologia Palmer e Felsing (2003, p. 23) declaram que a FDD é “[...] digna de uma consideração séria por qualquer organização de desenvolvimento de *software* que precisa entregar sistemas de *software* de qualidade, essenciais para o negócio e dentro do prazo.”

Ao todo, a FDD possui duas grandes fases, sendo a fase de Concepção e planejamento e a fase de Construção. Essas duas fases levam em média de uma a duas semanas e são divididas nos processos de Desenvolvimento de um modelo abrangente, Construção de lista de funcionalidades, Planejamento por funcionalidades, Detalhar por funcionalidade e Construir por funcionalidade (WAZLAWICK, 2013).

Figura 10 - Etapas de desenvolvimento utilizando FDD



Fonte: Wazlawick (2013)

4.2.2.1.1 Desenvolvimento de um modelo abrangente

Consiste primeiramente no desenvolvimento de um modelo de negócio abrangente, que então é seguido pelo desenvolvimento de modelos mais específicos (de domínio). São formados grupos compostos por desenvolvedores e também especialistas de domínio.

A construção desses modelos é feita por meio de diagramas de classe, por um profissional experiente em modelagem utilizando orientação a objetos. O modelo de negócio

será posteriormente refinado no processo de Detalhar por Funcionalidade. As atividades desempenhadas neste processo correspondem a:

a) Formar a equipe de modelagem: composta por especialistas de domínio, clientes e desenvolvedores.

b) Estudo dirigido sobre o domínio: atividade realizada pela equipe de modelagem, onde um especialista de domínio apresenta sua área de domínio para a equipe, notoriamente os aspectos conceituais.

c) Estudar a documentação: consiste no estudo da documentação sobre o domínio do problema. Não é uma atividade obrigatória.

d) Desenvolver o modelo: são criados modelos candidatos para áreas de domínio, que no final do processo serão consolidados em um modelo único.

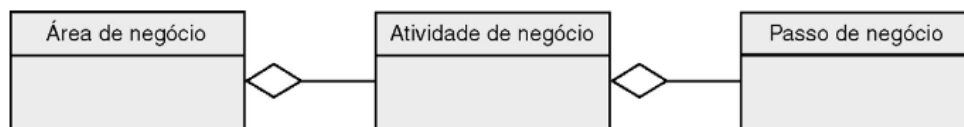
e) Refinar o modelo de objetos abrangente: nesta atividade são tomadas várias decisões tanto pela equipe de modelagem como pelo arquiteto líder, o que pode afetar a forma do modelo geral do negócio, que deve então ser refinado. Esta atividade conta também com a participação do cliente.

Outras atividades incluem a criação de um modelo conceitual, diagrama de classes, métodos e atributos para as classes. Também de forma opcional diagramas de sequência ou máquina de estados.

4.2.2.1.2 Construção da lista de Funcionalidades

Visa identificar as funcionalidades que atendem aos requisitos previamente estabelecidos. O domínio é decomposto em áreas de negócio conforme realizadas na etapa de Desenvolvimento do Modelo Abrangente.

As funcionalidades são listadas em 3 níveis, sendo elas Áreas de negócio, Atividades de negócio e Passos de atividades de negócio. Espera-se que o tempo para implementação de uma funcionalidade não seja maior que 2 semanas. Caso uma funcionalidade possa vir a demorar mais de 2 semanas para ser concluída, esta deve ser quebrada em funcionalidades menores.

Figura 11 - Níveis de funcionalidades da FDD

Fonte: Wazlawick, 2013

4.2.2.1.3 Planejamento por funcionalidades

A terceira fase, planejamento por funcionalidades, tem por objetivo gerar o plano de desenvolvimento para os ciclos iterativos, visando indicar quais atividades de negócio definidas na lista de funcionalidades serão implementadas e quando. Os pontos a serem considerados pelo planejador são a complexidade das funcionalidades, dependências entre as funcionalidades em termos de classes e carga de trabalho da equipe. As atividades desempenhadas neste processo correspondem a formar a equipe de planejamento, determinar a sequência de desenvolvimento, atribuir atividades de negócio aos programadores líderes e atribuir classes aos desenvolvedores.

4.2.2.1.4 Detalhar por funcionalidade

A quarta fase, detalhar por funcionalidade, consiste basicamente em produzir o design de implementação da funcionalidade, que costuma ser realizado por diagramas de sequência ou comunicação (WAZLAWICK, 2011).

4.2.2.1.5 Construção por funcionalidade

A quinta fase, construir por funcionalidade é executada dentro da fase iterativa da FDD, seu objetivo é a produção/desenvolvimento de código para as funcionalidades identificadas nas etapas passadas. Os desenvolvedores devem construir e testar o código necessário para cada classe atribuída. Dentre as atividades executadas nesse processo pode-se destacar Implementar classes e métodos, Inspeções de código, Testes de unidade e Promover a versão atual (*build*).

Após executadas as atividades necessárias, obtém-se as classes que passaram com sucesso nos testes de unidade e integração, e que foram promovidas a versão atual (*build*). E a disponibilização de um conjunto de funcionalidades com valor para o cliente.

4.2.2.1.6 Pessoas envolvidas

Assim como outras abordagens ágeis, a FDD adota uma filosofia que enfatiza a colaboração entre pessoas da equipe (PRESSMAN, 2016). Dentre as pessoas que participam dos processos da FDD estão o Gerente de projeto, Arquiteto-chefe/especialista no domínio, Equipe de modelagem/planejamento, Programador-chefe e Equipe de funcionalidades (SBROCCO E MACEDO, 2012).

O gerente de Projeto tem como funções estabelecer contato direto com o cliente do projeto e captar todos os requisitos e restrições. Também é responsável por formar a equipe de desenvolvimento de acordo com as *features* (funcionalidades).

O arquiteto-chefe/especialista no domínio é o profissional especialista no assunto do projeto a ser desenvolvido, sendo responsável por sanar as dúvidas sobre as regras de negócio.

A equipe de modelagem/planejamento corresponde a pessoas capazes de elaborar a lista de funcionalidades do sistema/projeto a ser desenvolvido, e também dividir e distribuí-las para a equipe de funcionalidades, do inglês *Team Features*.

O programador chefe tem como atribuições, refinar a lista de *features* e transformá-las em modelo de objetos. Também trabalha na parte organizacional do trabalho, escolhendo a linguagem e as formas de armazenamento, bem como as integrações “*build*”, revisando e aprovando a liberação para o cliente.

A equipe de modelagem/planejamento são os desenvolvedores de *software* propriamente ditos. Estes devem conhecer as tecnologias necessárias para o desenvolvimento do projeto. Devem ser capazes de desenvolver as funcionalidades necessárias e também realizar os testes unitários para validar o projeto e inspecionar o mesmo em todas as suas fases.

4.2.2.1.7 Práticas da FDD

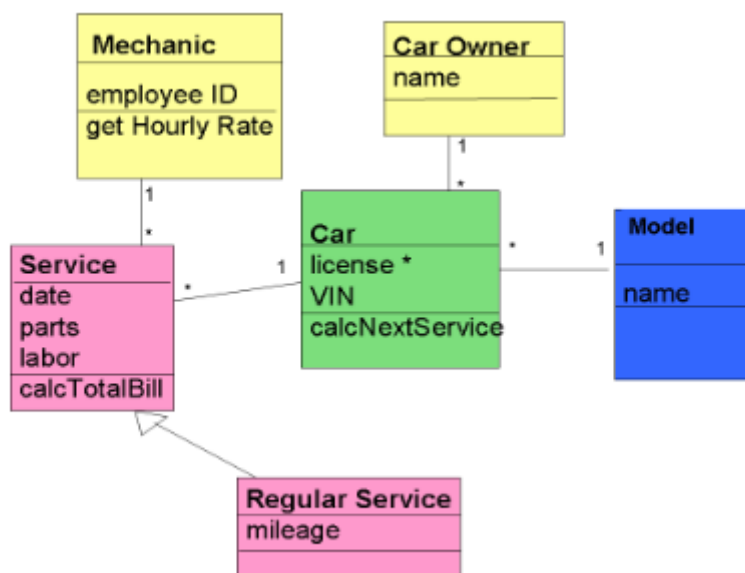
A FDD é construída em torno de um conjunto de melhores práticas, cada prática complementa e reforça as outras (PALMER E FELSING, 2003). Essas práticas correspondem a modelagem dos objetos de domínio, UML em cores, propriedade individual da classe, equipes de características, inspeções, construções regulares, administração de configuração e relatórios de resultados.

A modelagem dos objetos de domínio consiste na construção de diagramas de classes representando os tipos significativos de objetos dentro de um domínio de problema e os relacionamentos entre eles.

UML em cores consiste em UML regular com classes codificadas por cores. O uso da cor permite uma rápida compreensão da dinâmica do domínio do problema. Todas as classes são divididas em diferentes categorias; cada categoria tem sua própria cor. As classes auxiliares e interfaces são incolores (Goyal, 2008).

A cor amarelo representa um papel sendo desempenhado, azul representa uma descrição tipo catálogo, o verde representa um lugar ou coisa e o rosa representa um momento no tempo ou um intervalo de tempo geralmente associado a algum processo de negócios.

Figura 12 - UML em cores



Fonte: Goyal (2008)

O desenvolvimento através de característica ocorre após serem identificadas as classes na Modelagem dos objetos de domínio, cada uma destas pode ser desenhada e implementada. Então, após ter completado um conjunto de classes essas podem ser integradas compondo uma parte do projeto.

A propriedade individual da classe denota quem (pessoa ou função) é responsável pelo conteúdo de uma classe (porção de código). A FDD usa propriedade individual, ou seja, os desenvolvedores recebem a propriedade de um conjunto de classes do modelo de objeto de domínio (GOYAL, 2008).

As equipes de características são compostas pelos “donos” de cada classe. Dependendo do quão complexa seja a funcionalidade, o programador chefe pode convocar um especialista de negócio para que este conduza um novo *Domain Walkthrough* (Estudo dirigido sobre o domínio) para a equipe, para não ficar dúvidas a respeito daquela funcionalidade em específico.

As Inspeções têm o propósito primário de detectar defeitos, quando feita da maneira correta, também existem 2 benefícios secundários das inspeções que são disseminar a cultura de desenvolvimento e experiência e conformidade com padrões.

As construções regulares podem ser diárias, outras semanais e outras continuamente a depender do tamanho do projeto. Em intervalos regulares, todo o código fonte para as funcionalidades que foram completadas, juntamente com os componentes e bibliotecas que este depende são construídos e completam o sistema.

A administração de configuração apenas se faz necessária em um projeto que segue a FDD para identificar o código-fonte de todos os recursos que foram concluídos até o momento e manter um histórico de alterações nas classes à medida que as equipes de recursos os aprimoram.

A equipe do projeto deve colocar todas as práticas acima em uso para cumprir com as regras de desenvolvimento da FDD. No entanto, a equipe pode adaptá-los de acordo com seu nível de experiência (ABRAHAMSSON, 2002).

4.4 PADRÕES DE ARQUITETURA DE *SOFTWARE*

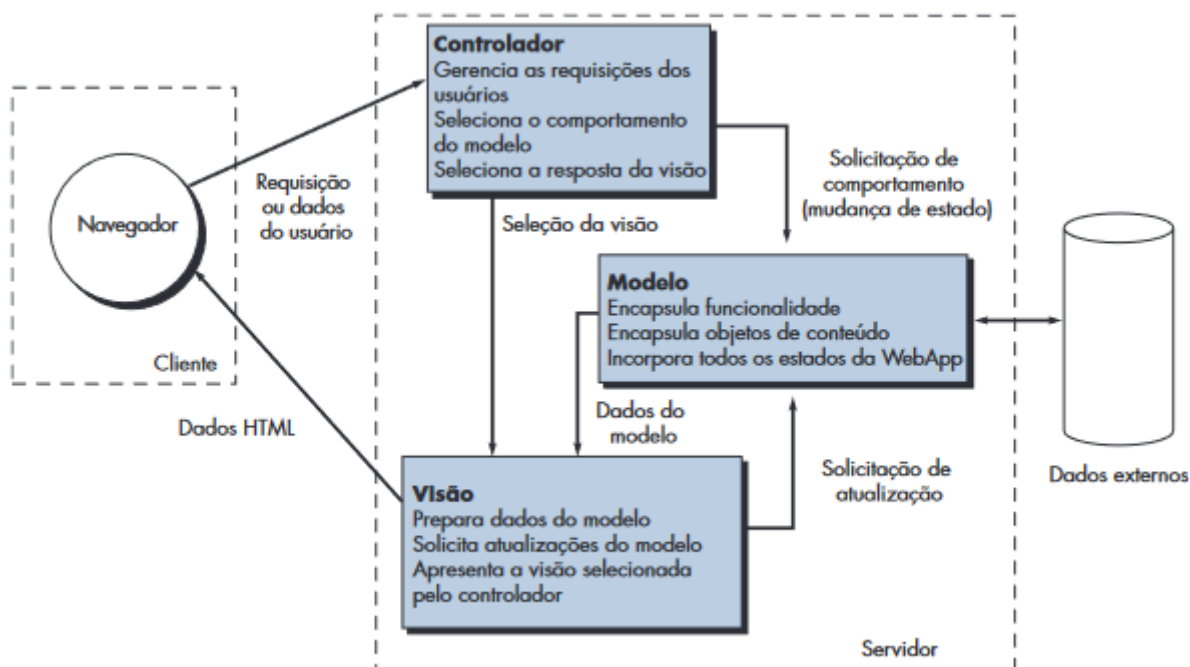
4.4.1 *Model-View-Controller*

Para o desenvolvimento do sistema proposto neste trabalho, será utilizado o padrão de arquitetura de *software Model-View-Controller* (MVC). Este é um dos padrões mais utilizados no desenvolvimento de sistemas, especialmente em desenvolvimento web. O objetivo do padrão MVC é separar as diferentes responsabilidades de um sistema, tornando-o mais organizado, flexível e fácil de manter.

De acordo com Erich Gamma (2007, p.20):

A abordagem MVC é composta por três tipos de objetos. O Modelo é o objeto de aplicação, a Visão é a apresentação na tela e o Controlador é o que define a maneira como a interface do usuário reage às entradas do mesmo. Antes da MVC, os projetos de interface para o usuário tendiam a agrupar esses objetos. A MVC separa esses objetos para aumentar a flexibilidade e a reutilização.

Figura 13 - Exemplo de aplicação usando MVC



Fonte: Pressman (2011)

A camada de visão torna possível exibir o conteúdo e a lógica de processamento. Também possibilita acessar os dados/informações externas necessárias para interação do usuário com a aplicação. O modelo contém todo o conteúdo e a lógica de processamento específicos à aplicação e também funcionalidades específicas. O controlador gerencia o acesso ao modelo e à visão e coordena o fluxo de dados entre eles (PRESSMAN, 2011).

4.5 LINGUAGEM DE MODELAGEM

Para a modelagem do sistema a ser desenvolvido, optou-se por utilizar a *Unified Modeling Language* (UML), que é uma linguagem de modelagem amplamente utilizada para descrever o comportamento e estrutura de uma ampla gama de sistemas e processos por meio de vários tipos de diagramas. Ela permite uma representação gráfica clara e consistente, permitindo a comunicação eficiente de ideias e conceitos entre desenvolvedores, engenheiros e analistas.

De acordo com Wazlawick (2011, p. 3) a UML possui três famílias de diagramas sendo eles, respectivamente:

Diagramas estruturais: compreendendo os diagramas de pacotes, classes, objetos, estrutura composta, componentes e distribuição.

Diagramas comportamentais: compreendendo os diagramas de casos de uso, atividades e máquina de estados.

Diagramas de interação: compreendendo os diagramas de comunicação, sequência, tempo e visão geral de integração.

5 ANÁLISE DE REQUISITOS DO SISTEMA

5.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma etapa fundamental no processo de desenvolvimento de software, tendo como objetivo identificar, coletar e documentar todas as necessidades, expectativas e funcionalidades que um sistema ou projeto deve atender. É uma etapa de extrema importância para garantir que o projeto a ser desenvolvido esteja de acordo com as necessidades e desejos dos usuários e das partes interessadas envolvidas.

Consiste em um procedimento abrangente que envolve todas as ações direcionadas para a coleta, registro e manutenção das exigências de um sistema. Isso implica na identificação e compreensão das necessidades das partes envolvidas, na conversão dessas necessidades em requisitos definidos e também sucintos (WIEGERS; BEATTY 2013).

De acordo com Sommerville (2011, p. 57) os requisitos de sistema podem ser entendidos como:

[...] as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações.

Refere-se ao procedimento de identificar as funções que o sistema necessita executar, bem como as limitações e condições que são impostas a essas funções (WAZLAWICK, 2011).

No que diz respeito aos requisitos do sistema a ser desenvolvido, estes devem alinhar-se ao propósito de proporcionar uma maneira organizada e eficaz de lidar com vários aspectos do atendimento ao paciente, otimizando o fluxo de trabalho interno e garantindo uma melhor qualidade de atendimento. Além disso, o sistema deverá permitir a obtenção de informações valiosas provenientes dos registros dos pacientes que receberem tratamento na unidade.

5.1.1 Requisitos funcionais

Requisitos funcionais (Tabela 7) definem as funções específicas que um sistema ou componente deve executar, incluindo as entradas, processamentos e saídas. Eles descrevem o comportamento do sistema em termos de serviços que ele deve oferecer aos usuários ou a

outros sistemas. Esses requisitos geralmente detalham as operações do sistema, as regras de negócios e as interações com os usuários (WIEGERS; BEATTY 2013).

Tabela 8 - Requisitos Funcionais

Código	Nome	Descrição	Prioridade
RF001	Autenticação	O sistema deverá gerenciar o cadastro e acesso do usuário no sistema, utilizando as credenciais de login: e-mail e senha.	MÉDIA
RF002	Registro de pacientes	O sistema deve permitir o cadastro e registro de informações pessoais dos pacientes, como nome, idade, gênero, endereço e contato de emergência.	ALTA
RF003	Triagem	O sistema deve possibilitar a triagem dos pacientes com base em critérios de gravidade e prioridade de atendimento.	ALTA
RF004	Registro de atendimentos	O sistema deve possibilitar o registro de todas as informações relevantes sobre o atendimento médico prestado a cada paciente, incluindo sintomas, diagnóstico, tratamento e prescrições médicas.	ALTA
RF005	Gestão de leitos	O sistema deve permitir o gerenciamento e acompanhamento dos leitos disponíveis na UPA, possibilitando a reserva, ocupação e liberação de leitos conforme a demanda.	ALTA
RF006	Acompanhamento de pacientes	O sistema deve permitir o acompanhamento de casos de pacientes e seu estado de saúde.	ALTA
RF007	Prontuário eletrônico	O sistema deve disponibilizar um prontuário eletrônico para cada paciente, possibilitando o acesso rápido e seguro às informações médicas.	ALTA
RF008	Registro de ocorrências e eventos adversos	O sistema deve permitir o registro de ocorrências e eventos adversos ocorridos durante o atendimento, auxiliando na análise e melhoria	ALTA

		contínua dos processos.	
RF009	Prescrição eletrônica	O sistema deve permitir que os médicos emitam prescrições médicas eletrônicas, facilitando a dispensação de medicamentos.	MÉDIA
RF010	Gerenciamento de estoque de medicamentos	O sistema deve possibilitar o controle do estoque de medicamentos, alertando quando houver baixa quantidade ou vencimento próximo de algum medicamento.	MÉDIA
RF011	Geração de relatórios estatísticos	O sistema deve gerar relatórios estatísticos que auxiliem na análise do fluxo de pacientes, tempos de espera, demanda por serviços médicos, entre outros indicadores relevantes para a gestão da UPA.	MÉDIA

Fonte: O autor (2023)

5.1.2 Requisitos não funcionais

Durante o processo de desenvolvimento de sistemas, além de ser necessário é crucial levar em conta os requisitos não funcionais, que englobam áreas como desempenho, disponibilidade, usabilidade e modificabilidade, entre diversos outros. Esses requisitos desempenham um papel fundamental ao orientar as decisões de projeto.

Os requisitos não funcionais do sistema (Tabela 8) correspondem a todos os requisitos que não são essenciais para o funcionamento do sistema, porém podem impactar na qualidade do sistema. Eles podem abordar aspectos como segurança, usabilidade, escalabilidade, desempenho, disponibilidade, confiabilidade e outros atributos que impactam a experiência global do usuário e a eficiência do sistema.

Tabela 9 - Requisitos Não Funcionais

Código	Nome	Descrição	Prioridade
RNF001	Estabilidade	O sistema deve ser estável, evitando travamentos, erros frequentes e comportamentos inesperados.	ALTA
RNF002	Performance de	O sistema deve oferecer respostas	ALTA

	consulta	rápidas e eficientes às consultas realizadas pelos profissionais de saúde, permitindo uma análise ágil das informações.	
RNF003	Usabilidade	O sistema deve ser intuitivo e de fácil utilização pelos profissionais de saúde, independentemente do seu nível de experiência técnica	MÉDIA
RNF004	Criptografia de dados	O sistema deverá armazenar as credenciais do usuário de forma criptografada no banco de dados	MÉDIA
RNF005	Segurança	O sistema deve garantir a segurança dos dados dos pacientes, adotando medidas de criptografia, controle de acesso e prevenção contra ameaças cibernéticas.	MÉDIA

Fonte: O autor (2023)

5.2 MODELAGEM DO SISTEMA

Modelagem de sistemas é o processo de criar representações de sistemas de informação visando compreender seus aspectos mais específicos e essenciais. Essas representações podem incluir diagramas e gráficos, onde geralmente se utiliza a UML.

Um dos princípios da Engenharia de Software é a modelagem visual, onde existe o conceito da abstração, o qual exige que o *software* seja entendido de forma visual. A UML é utilizada para representar visualmente várias características da arquitetura de um sistema (WAZLAWICK, 2013).

Para o escopo do sistema a ser desenvolvido, foram escolhidos 3 diagramas da UML, sendo eles o Diagrama de Casos de Uso, Diagrama de Classes e Diagrama de Atividades.

5.2.1 Diagrama de Casos de Uso

Este diagrama, se trata de uma representação externa de alto nível do sistema. Permite representar de forma gráfica os atores, casos de uso e os relacionamentos entre esses elementos. É possível exemplificar de maneira ampla as interações entre os elementos externos e as funcionalidades do sistema (BEZERRA, 2007).

O primeiro passo ao escrever um caso de uso é definir o conjunto de “atores” envolvidos na história. Atores são as diferentes pessoas (ou dispositivos) que usam o sistema ou produto no contexto da função e comportamento a ser descrito. Os atores representam os papéis que pessoas (ou dispositivos) desempenham enquanto o sistema opera (PRESSMAN, 2011).

A tabela 9 apresenta os atores envolvidos no sistema a ser desenvolvido.

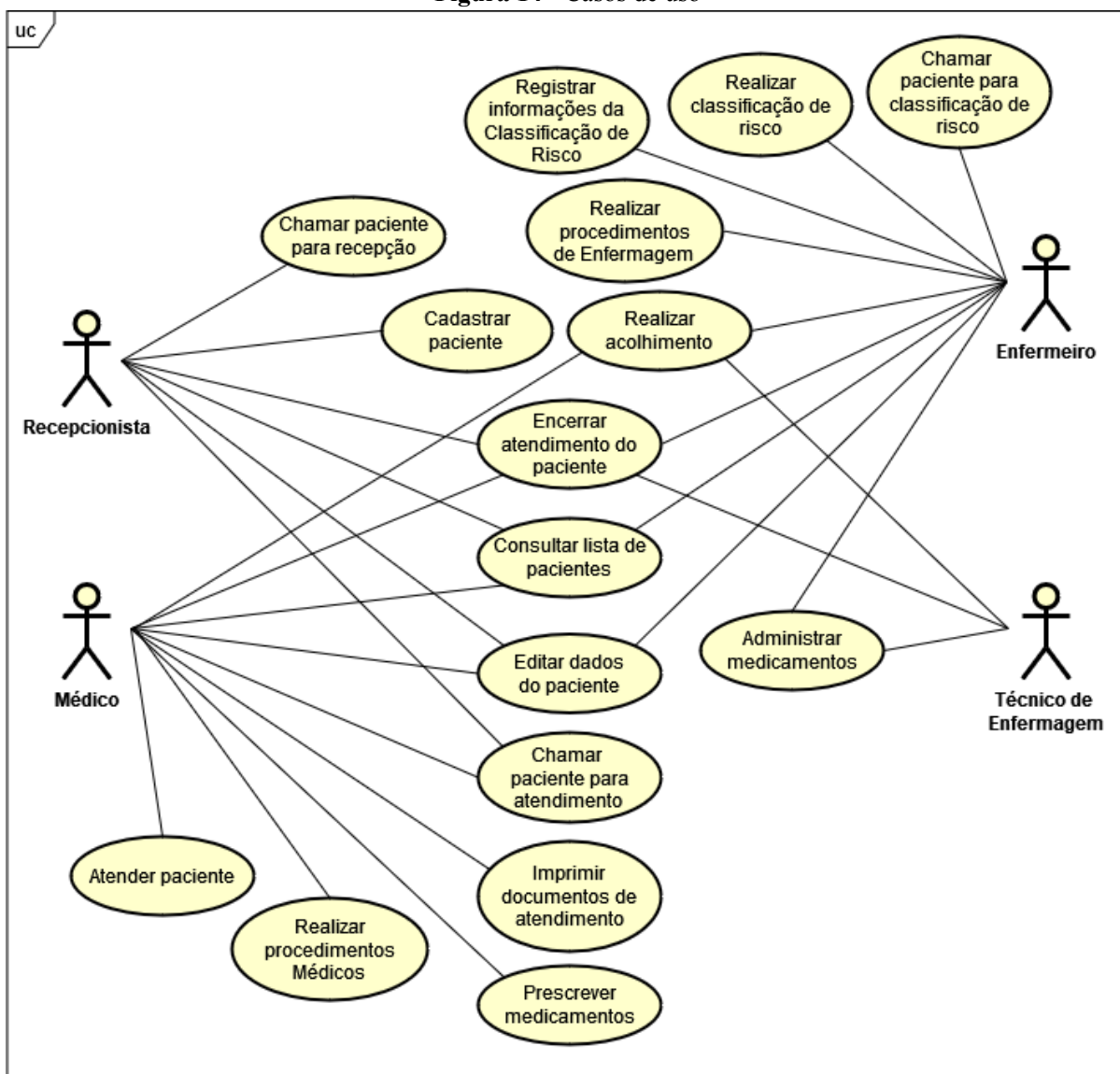
Tabela 10 - Atores (casos de uso)

Ator	Descrição
Recepcionista	É responsável por receber os pacientes que chegam à UPA, fazer o registro de informações básicas, como nome, motivo da visita e dados de contato, e encaminhar o paciente para a triagem inicial. Além disso, ela também pode orientar os pacientes sobre os procedimentos internos, tempos de espera e direcioná-los para os setores adequados.
Enfermeiro	Desempenha várias funções clínicas, incluindo a triagem inicial dos pacientes, avaliação de seus sinais vitais, histórico médico e sintomas. Além disso, os enfermeiros podem administrar medicamentos, realizar curativos, coletar amostras para exames laboratoriais, monitorar pacientes em observação e ajudar na coordenação geral dos cuidados médicos
Técnico de Enfermagem	Auxiliam os enfermeiros no cuidado direto aos pacientes. Isso pode incluir ajudar com a coleta de informações vitais, auxiliar na administração de medicamentos, fazer curativos simples, preparar os pacientes para exames e procedimentos, e realizar tarefas gerais de apoio à equipe médica.
Médico	É responsável por avaliar os pacientes, fazer diagnósticos, prescrever tratamentos e procedimentos, solicitar exames adicionais quando necessário e tomar decisões clínicas importantes. Eles coordenam a equipe de cuidados de saúde e determinam se os pacientes precisam ser encaminhados para atendimento especializado ou internação em hospitais.

Fonte: Elaborado pelo autor (2023)

A figura 14 representa o diagrama de casos de uso para o sistema a ser desenvolvido.

Figura 14 - Casos de uso



Fonte: Elaborado pelo autor (2023)

Visando aprimorar a compreensão dos cenários de utilização do sistema, procedemos à elaboração das descrições dos casos de uso (Tabela 10). Essa etapa envolve identificar os atores (usuários ou sistemas) envolvidos e descrever as interações entre esses atores e os casos de uso do sistema. Essas descrições fornecem um entendimento claro de como o sistema funcionará em situações do mundo real, ajudando a guiar o desenvolvimento, testes e comunicação entre as equipes.

Tabela 11 - Descrição dos casos de uso

Caso de Uso	Ator(s)	Descrição
CDU 01 Chamar paciente para recepção	Recepcionista	Chama o paciente na recepção, adicionando o mesmo na lista de pacientes do sistema
CDU 02 Cadastrar paciente no sistema	Recepcionista	Realiza o cadastro dos dados do paciente no sistema
CDU 03 Realizar acolhimento	Técnico de Enfermagem, Enfermeiro	Realiza o acolhimento do paciente, registrando as informações iniciais no sistema
CDU 04 Chamar paciente para classificação de risco	Enfermeiro	Chama o paciente para a sala de classificação de risco
CDU 05 Realizar classificação de risco	Enfermeiro	Realiza os procedimentos da classificação de risco para o paciente registrando no sistema
CDU 06 Registrar informações da classificação de risco no sistema	Enfermeiro	Registra as informações coletadas/obtidas por meio da classificação de risco no sistema
CDU 07 Chamar paciente para atendimento	Médico, Recepcionista	Chama o paciente até o consultório médico a fim de receber seu atendimento
CDU 08 Atender paciente	Médico	Realiza o atendimento médico do paciente registrando as informações no sistema
CDU 09 Realizar procedimentos médicos	Médico	Realiza os procedimentos necessários para o paciente e registra no sistema
CDU 10 Prescrever medicamentos	Médico	Permite ao médico realizar a prescrição de medicamentos, registrando assim no sistema
CDU 11 Imprimir documentos de atendimento	Médico	Permite gerar em PDF documentos contendo informações relacionadas ao atendimento do paciente, sendo eles: Prescrição de medicamentos, atestado e comprovante de comparecimento
CDU 12 Administrar medicamentos	Enfermeiro, Técnico de Enfermagem	Permite registrar no sistema o medicamento(s) administrado ao paciente

CDU 13 Consultar lista de pacientes	Recepcionista, Enfermeiro, Médico	Permite visualizar uma lista com todos os pacientes que foram recebidos na unidade no dia em questão
CDU 14 Editar dados do paciente	Recepcionista, Enfermeiro, Médico	Permite editar os dados cadastrais dos pacientes
CDU 15 Encerrar atendimento do paciente	Recepcionista, Enfermeiro, Técnico de enfermagem e Médico	Permite encerrar o protocolo/atendimento do paciente quando o mesmo já tiver sido atendido ou porventura se ausentou da unidade

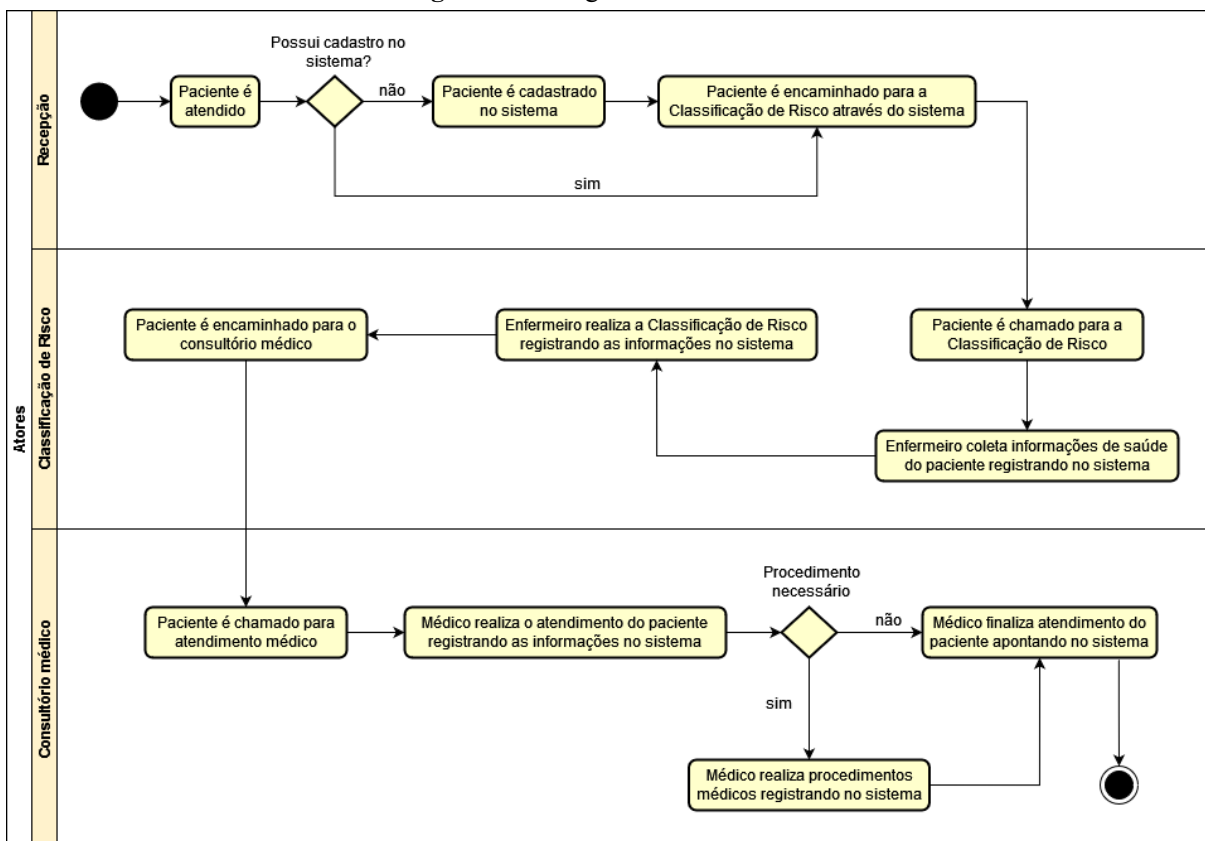
Fonte: Elaborado pelo autor (2023)

5.2.2 Diagrama de Atividades

O diagrama de atividades é uma ferramenta da UML (Linguagem de Modelagem Unificada) que visualiza o fluxo de atividades em um processo ou sistema. Ele usa símbolos como ações, decisões e bifurcações para representar a sequência e o paralelismo das atividades. É útil para compreender processos complexos e suas interações, tornando-os mais claros para equipes de desenvolvimento e as partes interessadas.

A figura 15 representa o diagrama de atividades referente ao atendimento a um paciente na UPA, mostrando as etapas desde a entrada deste na unidade até a finalização do atendimento médico.

Figura 15 - Diagrama de atividades



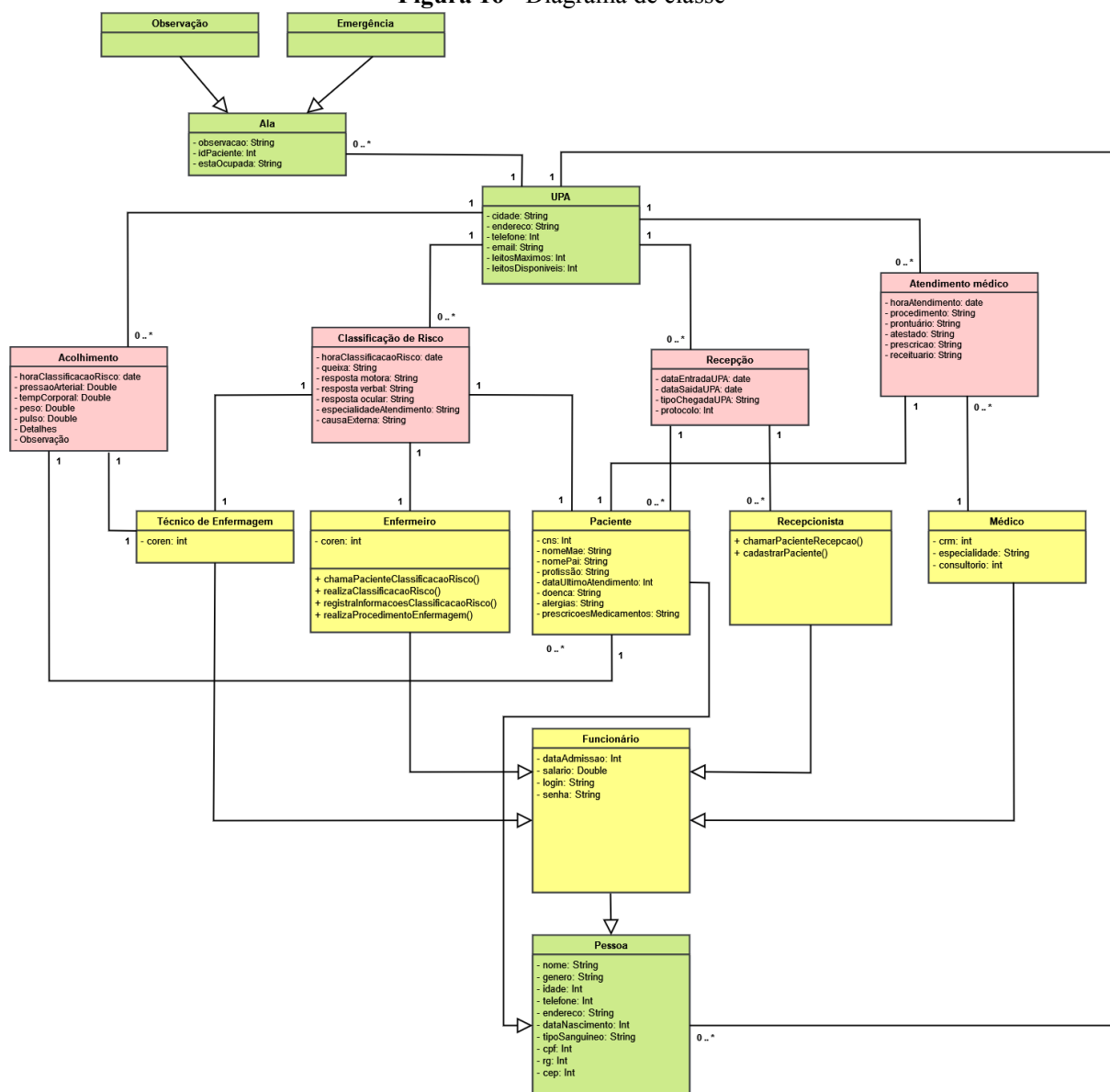
Fonte: Elaborado pelo autor (2023)

5.2.3 Diagrama de Classes

Um diagrama de classes é uma representação visual na área de desenvolvimento de software, que descreve as classes de um sistema, seus atributos e relacionamentos. Ele ajuda a compreender a estrutura do sistema e a interação entre seus componentes essenciais, facilitando o design e a comunicação entre equipes de desenvolvimento.

A figura 15 representa o diagrama de classes específico para o sistema seguindo o princípio proposto pela FDD utilizando o UML em cores.

Figura 16 - Diagrama de classe



Fonte: O autor (2023)

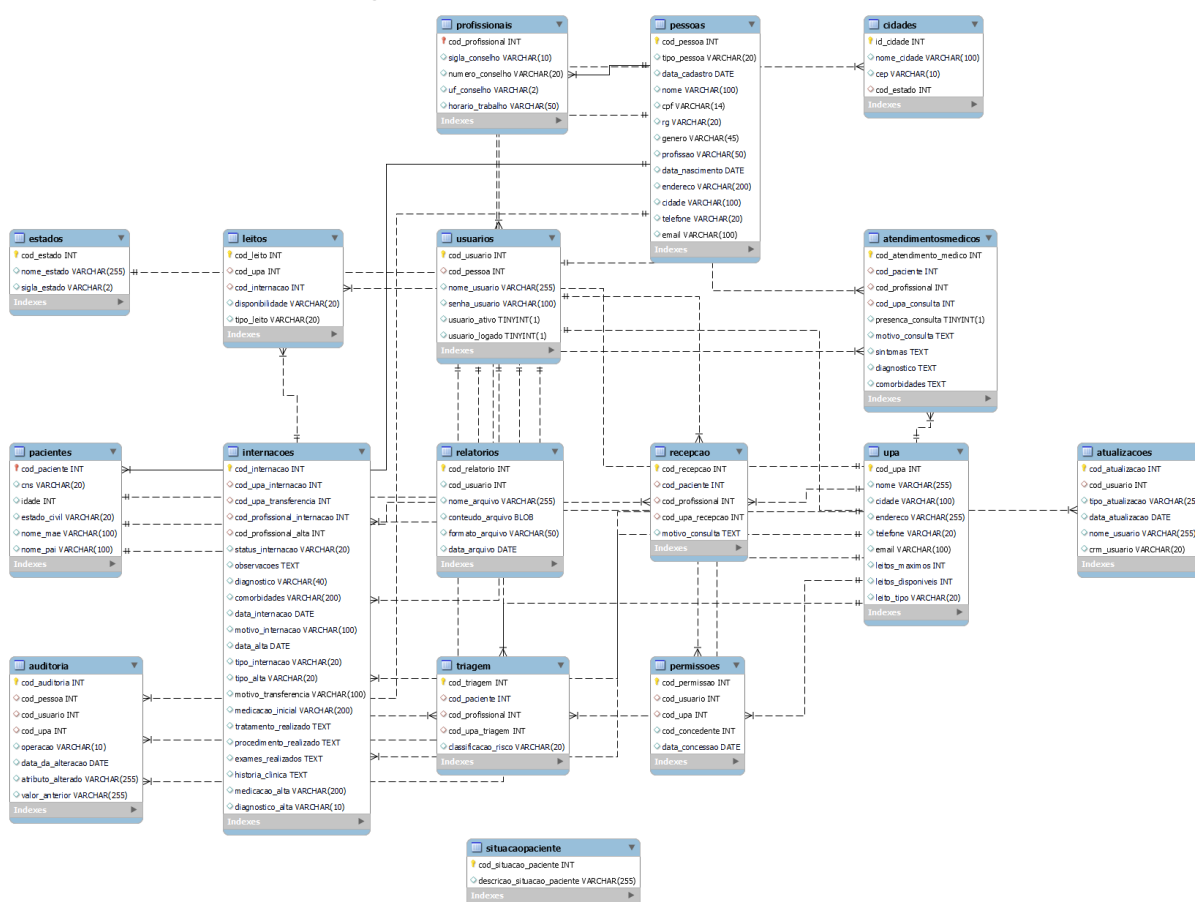
5.2.2 Banco de Dados

Um Banco de Dados desempenha um papel fundamental no desenvolvimento de sistemas de software. Ele corresponde a infraestrutura que permite armazenar, recuperar, gerenciar e manipular dados de maneira eficiente e segura. Um sistema de banco de dados compreende os seguintes componentes: o banco de dados em si, o Sistema Gerenciador de Banco de Dados (SGBD), os usuários e o hardware (SILBERSCHATZ; KORTH, 2004).

5.2.2.1 Modelo Entidade Relacionamento

Para facilitar a compreensão do banco de dados em desenvolvimento, é comum utilizar um diagrama entidade-relacionamento, como exemplificado na Figura 17. Isso proporciona uma representação visual das entidades e suas relações no sistema.

Figura 17 - Modelo Entidade Relacionamento



Fonte: Elaborado pelo autor (2023)

6 DESENVOLVIMENTO

6.1 TECNOLOGIAS UTILIZADAS

Para o desenvolvimento do Sistema Gerencial para a UPA, se faz necessário o uso de uma série de tecnologias essenciais que serão fundamentais para garantir a eficiência, segurança e funcionalidade do sistema. Estas tecnologias estão listadas na Tabela 11 e desempenham papéis cruciais em várias etapas do processo de desenvolvimento e operação do sistema, desde a coleta e armazenamento de dados até a interface do usuário e a entrega de informações em tempo real.

Tabela 12 - Tecnologias utilizadas

Tecnologia	Versão	Referência	Finalidade
Java	17	java.com	Linguagem de Programação Orientada a Objetos
JavaFx	11	openjfx.io	Biblioteca gráfica para utilizada para criar interfaces
IntelliJ IDEA	2023.1	jetbrains.com/idea	Ambiente de Desenvolvimento Integrado
MySQL	8.1.0	mysql.com	Sistema de Gerenciamento de Banco de Dados
MySQL Workbench	8.0.34	mysql.com/products/workbench	Ferramenta de design de Banco de Dados visual
Git	2.42.0	git-scm.com	<i>Software</i> utilizado para controle de versões
GitHub	3.10.0	github.com	Plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git
draw.io	21.6.8	draw.io	<i>Software</i> utilizado para elaboração de diagramas
CSS	4.15	developer.mozilla.org/en-US/docs/Web/CSS/Reference	Linguagem de marcação usada para adicionar estilos

Fonte: Elaborado pelo autor (2023)

6.1.1 Java

Java é uma linguagem de programação e plataforma computacional lançada pela *Sun Microsystems* em 1995. Ela evoluiu de origens modestas para alimentar uma grande parcela do mundo digital atual, fornecendo uma plataforma confiável na qual muitos serviços e aplicativos são construídos. Novos produtos inovadores e serviços digitais projetados para o futuro também continuam a depender do Java (ORACLE, 2023).

Uma das grandes vantagens da linguagem Java segundo Deitel (2016, p. 13) se deve ao fato de que “Um objetivo-chave do Java é ser capaz de escrever programas a serem executados em uma grande variedade de sistemas computacionais e dispositivos controlados por computador. Isso às vezes é chamado de “escreva uma vez, execute em qualquer lugar”.

Tal fato é possível graças a existência da JVM (*Java Virtual Machine*), que é um componente fundamental da plataforma Java. Este componente torna os programas independentes de plataforma, o que significa que um programa Java pode ser executado em qualquer dispositivo ou sistema operacional que tenha uma JVM compatível instalada.

6.1.2 JavaFX

JavaFX é uma plataforma de desenvolvimento de software que permite a criação de aplicativos gráficos interativos para desktop, web e dispositivos móveis usando a linguagem de programação Java. Inicialmente desenvolvido pela Sun Microsystems e agora mantido pela Oracle, o JavaFX fornece uma rica biblioteca de componentes e ferramentas para criar interfaces de usuário atraentes e responsivas.

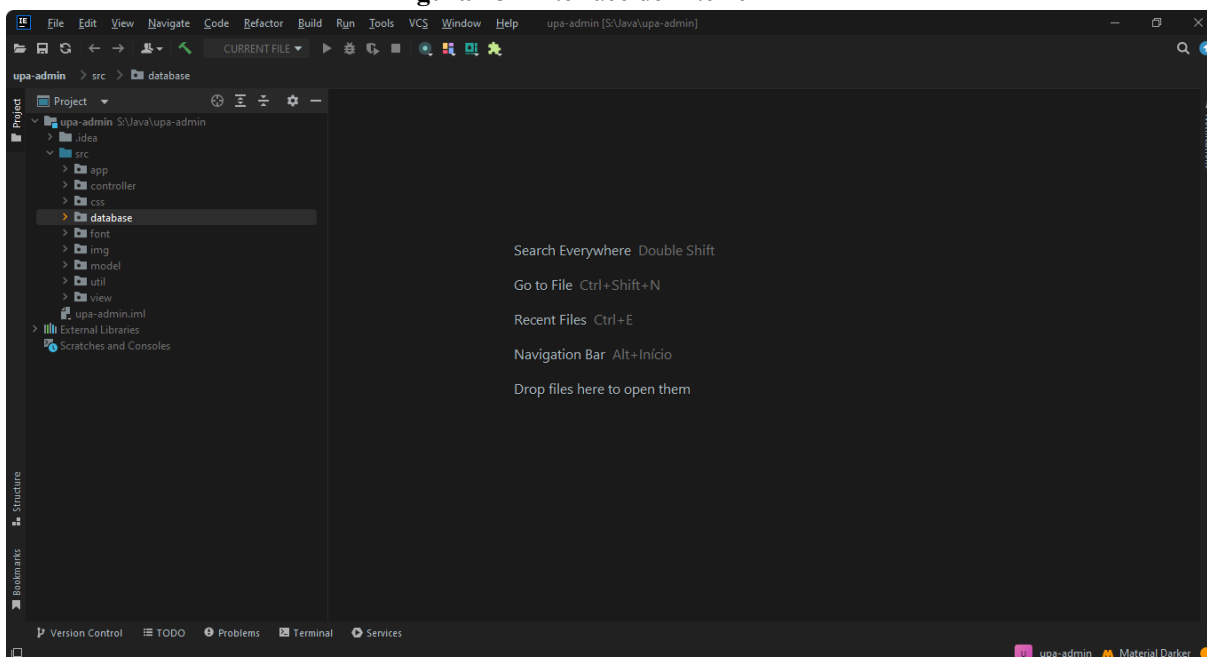
A principal vantagem do JavaFX é sua capacidade de criar interfaces gráficas altamente personalizáveis e modernas, incluindo gráficos avançados, animações, efeitos visuais e suporte para multimídia. Ele possui uma arquitetura robusta, permitindo a separação da lógica de negócios da interface do usuário através do padrão MVC (Model-View-Controller). Além disso, o JavaFX oferece suporte a CSS para estilização dos elementos da interface, proporcionando uma forma mais elegante de personalização visual.

6.1.3 IntelliJ IDEA

O IntelliJ IDEA é uma poderosa e popular IDE (*Integrated Development Environment*) desenvolvida pela JetBrains. Ela é projetada para atender às necessidades de desenvolvedores

Java, mas também oferece suporte a várias outras linguagens de programação, como Kotlin, Groovy, Scala, JavaScript, TypeScript e muito mais. Dentre os principais recursos oferecidos pela IDE pode-se citar o Code Assistance, Teste de Código, Ferramentas de Depuração, Análise de Código e Gerenciamento de Versão (JETBRAINS, 2023).

Figura 18 - Interface do IntelliJ IDEA



Fonte: O autor (2023)

6.1.4 MySQL

MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto amplamente utilizado em aplicativos web e outros projetos que exigem armazenamento e recuperação de dados. Foi desenvolvido originalmente pela empresa sueca MySQL AB e, posteriormente, adquirido pela Oracle Corporation.

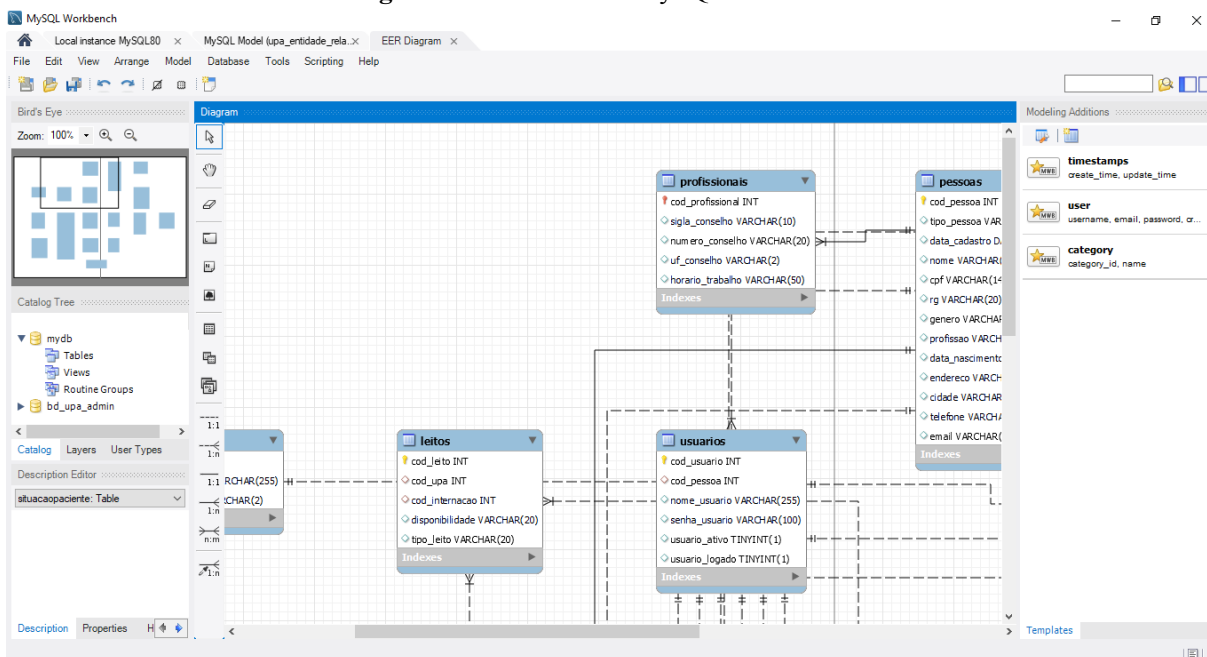
Uma das principais características do MySQL é sua facilidade de uso e compatibilidade com várias plataformas, o que o torna uma escolha popular para muitos desenvolvedores. Ele suporta a linguagem SQL (*Structured Query Language*) para consultas e manipulação de dados.

O MySQL é altamente escalável e pode lidar com grandes volumes de dados, tornando-o adequado para aplicativos com alta demanda de tráfego. Além disso, oferece suporte para replicação, permitindo a criação de cópias de dados em vários servidores para garantir a disponibilidade e a redundância.

6.1.5 MySQL Workbench

O MySQL Workbench é uma ferramenta gráfica de administração e modelagem de banco de dados desenvolvida pela Oracle. Ela é projetada para facilitar o gerenciamento de bancos de dados desenvolvidos em MySQL, desde o design e modelagem até a administração e manutenção. Um recurso interessante dessa ferramenta é a Modelagem ER (Entidade-Relacionamento), que possibilita realizar a modelagem avançada de banco de dados usando diagramas ER, que podem incluir recursos como herança, generalização, especialização, entre outros.

Figura 19 - Interface do MySQL Workbench



Fonte: o autor (2023)

6.1.6 Git

Criado por Linus Torvalds, o Git é um sistema de controle de versões distribuído utilizado no desenvolvimento de *software*. Ele permite controlar o histórico de alterações em arquivos diversos de um projeto. É o sistema de controle de versões mais utilizado no mundo, sendo *open source*.

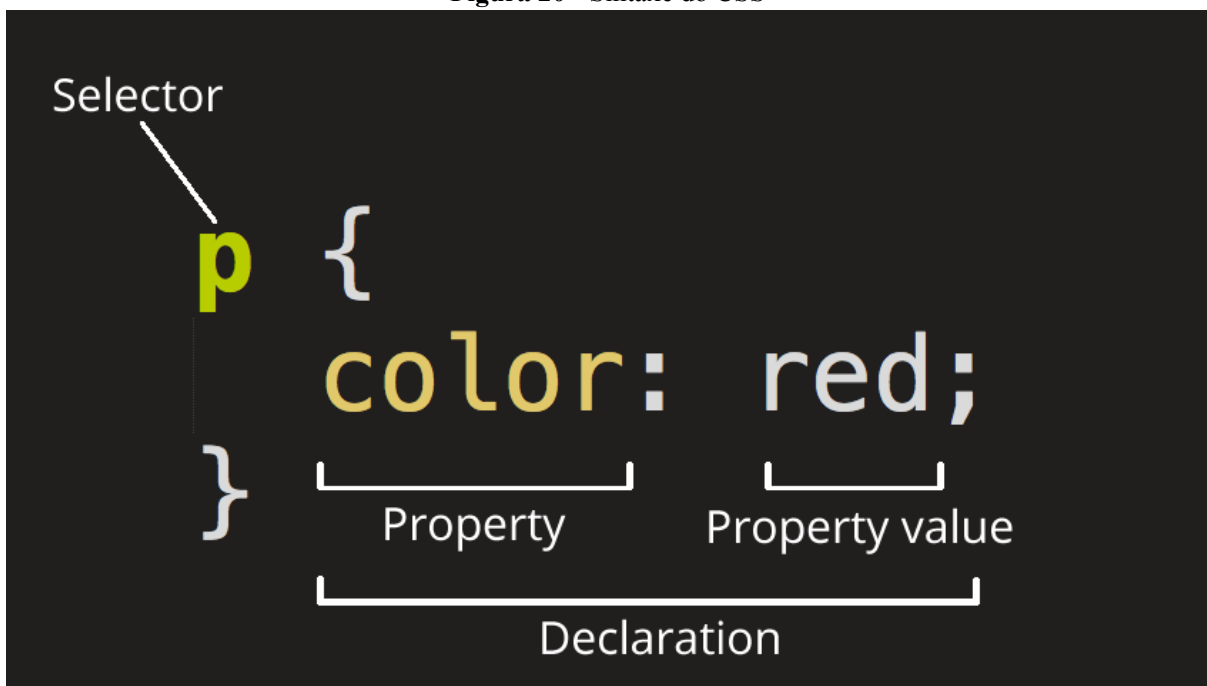
6.1.7 GitHub

É uma plataforma gratuita de hospedagem de código-fonte para o repositório Git. Ele oferece uma ampla gama de recursos para ajudar desenvolvedores a colaborar, gerenciar projetos, controlar versões de código-fonte e muito mais. Ele desempenha um papel crucial na comunidade de desenvolvimento de software e é amplamente utilizado por indivíduos, equipes e organizações em todo o mundo.

6.1.8 CSS

O CSS (*Cascading Style Sheets*), em português Folhas de Estilo em Cascatas, é uma linguagem de estilo utilizada para descrever a apresentação e a formatação de documentos HTML (*Hypertext Markup Language*). Ele desempenha um papel fundamental na criação de páginas web visualmente atraentes e bem projetadas. A sintaxe do CSS consiste em seletores e declarações. Os seletores identificam os elementos HTML aos quais as regras de estilo se aplicam, enquanto as declarações definem como esses elementos devem ser estilizados (Figura 20).

Figura 20 - Sintaxe do CSS

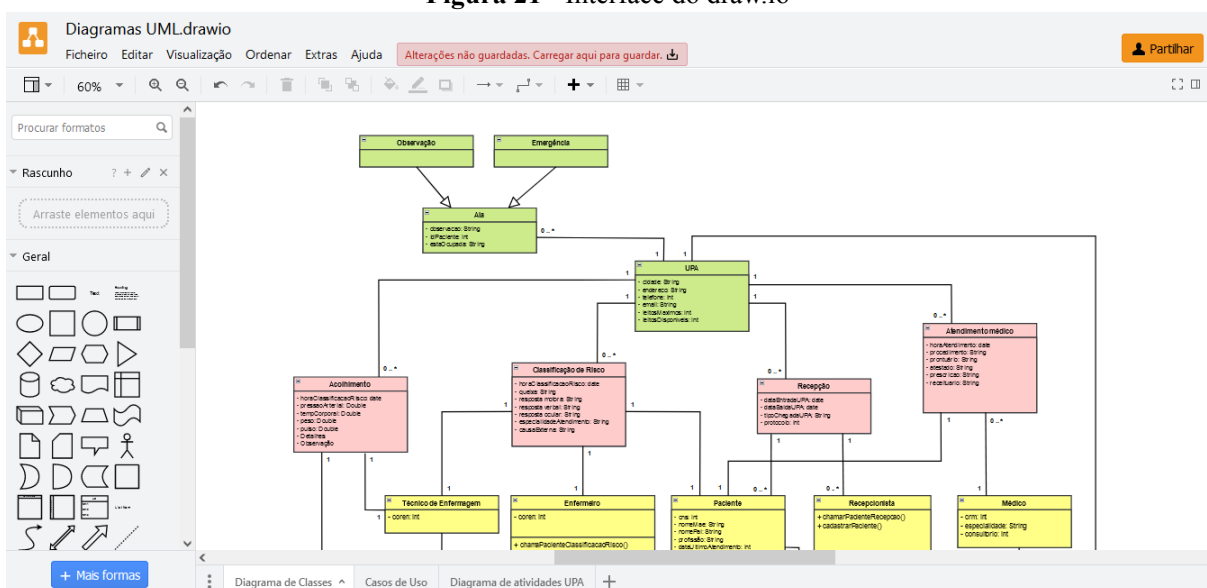


Fonte: developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/CSS_basics (2023)

6.1.9 Draw.io

O draw.io é uma ferramenta de diagramação online que permite criar diagramas, fluxogramas, organogramas, mapas mentais e outros tipos de diagramas de forma fácil e intuitiva. Ele é uma aplicação de código aberto e pode ser usado gratuitamente, sendo muito popular entre profissionais, estudantes e empresas. Uma característica interessante do draw.io é o uso de XML (Extensible Markup Language) para armazenar e representar os diagramas criados na plataforma.

Figura 21 - Interface do draw.io



Fonte: app.diagrams.net (2023)

REFERÊNCIAS

- Ministério da Saúde. UPA 24h. Disponível em:
<https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/u/upa-24h>. Acesso em: 23/07/2023.
- OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de Informações Gerenciais: estratégias, táticas operacionais**. 12ª ed. São Paulo: Atlas, 2008.
- MARIN H. **Sistemas de informação em saúde: considerações gerais**. Journal of Health Informatics. 2010; Disponível em:
<https://jhi.sbis.org.br/index.php/jhi-sbis/article/view/4/52>. Acesso em: 23/07/2023.
- BARROS, Ricardo. Ministério da Saúde. Gabinete do Ministro. **Portaria nº 10 de 3 de janeiro de 2017**. Disponível em:
http://bvsms.saude.gov.br/bvs/saudelegis/gm/2017/prt0010_03_01_2017.html. Acesso em: 23/07/2023.
- Ministério da Saúde. (2013). **Manual Instrutivo da Rede de Atenção às Urgências**. Brasília: Editora do Ministério da Saúde.
https://bvsms.saude.gov.br/bvs/publicacoes/manual_instrutivo_rede_atencao_urgencias.pdf
 Acesso em: 21/07/2023.
- Ministério da Saúde. (2009). Secretaria de Atenção à Saúde. **Política Nacional de Humanização da Atenção e Gestão do SUS. Acolhimento e classificação de risco nos serviços de urgência / Ministério da Saúde, Secretaria de Atenção à Saúde, Política Nacional de Humanização da Atenção e Gestão do SUS**. Disponível em:
https://bvsms.saude.gov.br/bvs/publicacoes/acolhimento_classificacao_risco_servico_urgencia.pdf. Acesso em: 21/07/2023.
- Conheça o Protocolo de Manchester implantado no Pronto Atendimento de São Mateus do Sul. Disponível em:
<https://www.saomateusdosul.pr.gov.br/portal/noticias/0/3/1184/conheca-o-protocolo-de-manchester-implantado-no-pronto-atendimento-de-sao-mateus-do-sul>. Acesso em: 21/07/2023.
- FEITOSA, M. M.; et. al.; **Acolhimento com Classificação de Risco na Unidade de Pronto Atendimento: Um Relato de Experiência**. Rev. Enferm. UFSM, v. 7, n. 1, p. 136-143, jan./fev., 2017.
- Você sabe o que é classificação de risco?. Disponível em:
<https://www.gov.br/ebserh/pt-br/hospitais-universitarios/regiao-nordeste/hujb-ufcg/comunicacao/noticias/voce-sabe-o-que-e-classificacao-de-risco>
 Acesso em: 04/08/2023.
- GDF - SES-DF. Projeto de Acolhimento: Plano de Implementação da Política de Humanização na Secretaria de Estado de Saúde do Distrito Federal. 2011. Disponível em:
http://www.saude.df.gov.br/005/00502001.asp?ttCD_CHAVE=13224#. Acesso em: 21/07/2023.

MOTA, L. A. N. da; PEREIRA, F. M. S.; SOUSA, P. A. F. de. **Sistemas de Informação de Enfermagem: exploração da informação partilhada com os médicos.** *Rev. Enf. Ref.*, Coimbra, v. serIV, n. 1, mar. 2014. Disponível em: http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S0874-02832014000100010&lng=pt&nrm=iso. Acesso em: 23/07/2023.

GALVAO, M.C.B. **Informação clínica: do prontuário do paciente às bases de evidência.** 23 de agosto de 2012. In: Almeida Junior, O.F. *Infohome* [Internet]. Londrina: OFAJ, 2012. Disponível em: http://www.ofaj.com.br/colunas_conteudo.php?cod=701. Acesso em: 23/07/2023.

CAVALCANTE, RICARO & NAGATA FERREIRA, MARINA & SILVA, POLIANA. **Sistemas de Informação em Saúde: possibilidades e desafios.** Revista de Enfermagem da Universidade Federal de Santa Maria. 1. 10.5902/217976922580.

O'BRIEN, James A. **Sistemas de informação e as decisões gerenciais na era da internet.** 2. ed. São Paulo: Saraiva, 2004

BATISTA, Emerson de Oliveira. **Sistemas de informação: o uso consciente da tecnologia para o gerenciamento.** São Paulo: Saraiva, 2004.

KROENKE, David. **Sistemas de informação gerenciais.** São Paulo: Saraiva, 2012.

PADOVEZE, Clóvis Luís. **Contabilidade gerencial.** 7. ed. São Paulo: Atlas, 2010

TURBAN, Efrain; MCLEAN, Ephraim; WETHERBE, James. **Tecnologia da informação para gestão.** 3. ed. Porto Alegre: Bookman, 2004.

FERREIRA, Sibebe Maria Gonçalves. **Sistema de informação em saúde: conceitos fundamentais e organização.** Revista de Saúde Pública, Belo Horizonte, v. 40, n. 4, p. 690-696, ago. 2006. Disponível em: <https://www.nescon.medicina.ufmg.br/biblioteca/imagem/2249.pdf>. Acesso em: 27 ago. 2023.

SPDATA. Controle Assistencial. Disponível em: <https://www.spdata.com.br/solucoes/hospitais/controle-assistencial>. Acesso em: 02/09/2023

JetBrains. IntelliJ IDEA. Disponível em: <https://www.jetbrains.com/idea/>. Acesso em: 01/08/2023

SOMMERVILLE, Ian. **Engenharia de Software.** 8ª ed. São Paulo: Pearson Addison-Wesley, 2011.

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

BECK, K. et al. **Manifesto for Agile Software Development.** Fev. 2001
Disponível em: <http://www.agilemanifesto.org>. Acesso em: 02/08/2023

PALMER, STEPHEN R.; FELSING, JOHN M.. **A Practical Guide to Feature-Driven Development.** Prentice Hall, 2002.

WAZLAWICK, Raul Sidnei. **Engenharia de Software: Conceitos e Práticas**. Rio de Janeiro: Campus, 2013.

WAZLAWICK, Raul Sidnei. **Análise e projeto de sistemas de informação orientados a objetos**. 2.ed. Rio de Janeiro: Elsevier, 2011.

PRESSMAN, Roger S.; Bruce R. Maxim. **Engenharia de Software, Uma Abordagem Profissional**, 8º ed. Porto Alegre: AMGH, 2016.

GOYAL, SADHNA. **Major Seminar On Feature Driven Development Agile Techniques for Project Management and Software Engineering**.

Blue Mundus News. Component Based Design and Why Your Website Needs It. 2021.

Disponível em: <https://csis.pace.edu/~marchese/CS616/Agile/FDD/fdd.pdf/>.

Acesso em: 27 ago. 2023

KHRAMTCHENKO, S. **Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments**. In: , Harvard University, EUA, 2004.

ABRAHAMSSON, P., SALO, O., RONKAINEN, J. & WARSTA, J. **Agile software development methods: Review and analysis**. VTT publication 478, Espoo, Finland, 2002.

Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf>>

KOPPENSTEINER, Sonja; UDO, Nathalie. **Will agile development change the way we manage software projects? Agile from s PMBOK guide perspective**. Projectway, LLC, 2003.

ORACLE, Java. Disponível em: https://www.java.com/en/download/help/whatis_java.html.

Acesso em: 03/08/2023

WIEGERS, Karl; BEATTY, Joy. **Software Requirements Third Edition**. 3.ed. Redmond: Microsoft Press, 2013. p. 637.

SILBERSCHATZ, A.; KORTH, H.F. **Sistema de Banco de Dados**. São Paulo: Mackron Brooks, 2004. 786p