

**INSTITUTO POLITÉCNICO DOM DAMIÃO FRANKLIN-Nº8028**

**COORDENAÇÃO DO CURSO DE INFORMÁTICA**

**CURSO TÉCNICO DE INFORMÁTICA**

**DESENVOLVIMENTO DE UM SISTEMA WEB DE AGENDAMENTO DE CONSULTAS**

**(Estudo de caso: Hospital Geral de Luanda no II trimestre, 2025/2026)**

**Autores:**

00783 -Ana Hossi,

00790 -Fábio Policarpo,

00795 -Hélvio Coimbra,

00807 -Milton Gaspar

**Orientador:** Prof. Lic. Teófilo Félix

Relatório da Prova de Aptidão Profissional (PAP) apresentado ao Instituto politécnico Dom Damião Franklin (IPDDF) para obtenção do grau de Técnico Médio de Informática

**LUANDA, 2025**



**INSTITUTO POLITÉCNICO DOM DAMIÃO FRANKLIN – Nº8028**

**DESENVOLVIMENTO DE UM SISTEMA WEB DE AGENDAMENTO DE CONSULTAS PARA O HOSPITAL GERAL DE LUANDA NO II TRIMESTRE DO ANO DE 2025/2026**

**Autores:**

00783 - Ana Hossi,

00790 - Fábio Policarpo,

00795 - Hélvio Coimbra,

00807 - Milton Gaspar

**Orientado por:** Prof. Lic. Teófilo Félix

Relatório da Prova de Aptidão Profissional (PAP) apresentado ao Instituto politécnico Dom Damião Franklin (IPDDF) para obtenção do grau de Técnico Médio de Informática

**LUANDA, 2025**

**FOLHA DE APROVAÇÃO**

Aprovado aos \_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_/2026

**BANCA EXAMINADORA**

Presidente

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

1ª Vogal

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

2ª Vogal

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

# **EPÍGRAFE**

"A tecnologia da saúde não substitui o cuidado humano, ela o amplifica."

**— Eric Topol**

"A jornada de mil milhas começa com um simples passo."

**— Lao-Tsé**

"Confia ao Senhor as tuas obras, e os teus pensamentos serão estabelecidos."

**— Provérbios 16:3**

# **DEDICATÓRIA**

Dedicamos este trabalho primeiramente às nossas famílias, que nos concederam força e apoio para superar todos os desafios encontrados ao longo desta jornada académica. Aos nossos pais, pelo amor incondicional, sacrifícios realizados e apoio constante em todos os momentos da nossa formação. Aos nossos amigos, que compreenderam as nossas ausências e nos incentivaram nos momentos de dificuldade.

# **AGRADECIMENTOS**

Primeiramente, agradecemos a Deus, pela vida, sabedoria, força e proteção ao longo desta jornada académica, permitindo-nos alcançar este importante marco nas nossas vidas.

Às nossas famílias, pelo amor incondicional, compreensão nos momentos de ausência e pelo incentivo constante que nos impulsionou a seguir em frente mesmo diante dos desafios.

Ao nosso orientador, Professor Licenciado Teófilo Félix, expressamos a nossa profunda gratidão pela orientação precisa, paciência, disponibilidade e pelos valiosos ensinamentos que foram fundamentais para a elaboração deste projecto.

Ao Instituto Politécnico Dom Damião Franklin (IPDDF), pela formação de qualidade

proporcionada ao longo do curso, e a todos os docentes que contribuíram para a nossa

formação técnica e humana, transmitindo conhecimentos que levaremos para toda a vida.

Ao Hospital Geral de Luanda, pela abertura e colaboração no fornecimento de informações essenciais para a realização deste estudo de caso, e aos funcionários e pacientes que disponibilizaram o seu tempo para responder aos nossos questionários e entrevistas.

Aos nossos colegas de turma TI13A, pela partilha de experiências, pelo companheirismo e pelo apoio mútuo durante estes anos de formação. Juntos enfrentámos desafios e celebrámos conquistas que jamais esqueceremos.

Muito obrigado.

# **LISTA DE FIGURAS**

[Figura 1 - Actividade da Engenharia de Requisitos 13](#_Toc216793923)

[Figura 2 - Processo incremental 16](#_Toc216793924)

[Figura 3 - Diagrama de Casos de Uso 26](#_Toc216793925)

[Figura 4 - Diagrama de Actividade: Cadastrar Paciente 29](#_Toc216793926)

[Figura 5 - Diagrama de Actividades: Notificar Médico 30](#_Toc216793927)

[Figura 6 - Diagrama de Actividades: Cadastrar paciente 31](#_Toc216793928)

[Figura 7 - Diagrama de Actividades: Agendar consulta 32](#_Toc216793929)

[Figura 8 - Diagrama de Actividades: Notificar Médico 32](#_Toc216793930)

# **LISTA DE TABELAS**

[Tabela 1 - Requisitos Funcionais do Sistema 23](#_Toc216793742)

[Tabela 2 - Requisitos Não Funcionais do Sistema 24](#_Toc216793743)

[Tabela 3 - Regras de Negócio 24](#_Toc216793744)

[Tabela 4 - Descrição de Caso de Uso: Cadastrar Paciente 27](#_Toc216793745)

[Tabela 5 - Descrição de Caso de Uso: Agendar Consultas 27](#_Toc216793746)

[Tabela 6 - Descrição de Caso de Uso: Notificar Médico 28](#_Toc216793747)

# **LISTA DE ABREVIAÇÕES E SIGLAS**

API – Application Programming Interface (Interface de Programação de Aplicações)

AWS – Amazon Web Services

BD – Base de dados

CSS – Cascading Style Sheets (Folhas de Estilo em Cascata)

GPS – Global Positioning Sytem (Sistema de Posicionamento Global)

HTML – HyperText Markup Language (Linguagem de Marcação de Hipertexto)

HTTP – HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)

HTTPS – HyperText Transfer Protocol Secure

IA - Inteligência Artificial

JIT - Just-In-Time

JS - JavaScript

MVC - Model-View-Controller (Modelo-Visão-Controlador)

NoSQL - Not Only SQL (Não Apenas SQL)

OMG - Object Management Group (Grupo de Gestão de Objectos)

OMT- Object Modelling Technique (Técnica de Modelagem de Objectos)

OOSE- Object-Oriented Software Engineering (Engenharia de Software Orientada a

Objectos)

PDO - PHP Data Objects (Objectos de Dados PHP)

PHP - Hypertext Preprocessor (Pré-processador de Hipertexto)

RAM - Random Access Memory (Memória de Acesso Aleatório)

RDBMS- Relational Database Management System (Sistema de Gestão de Base de Dados

Relacional)

SGBD- Sistema de Gerenciamento de Bases de Dados

SI - Sistemas de Informação

SMS - Short Message Service (Serviço de Mensagens Curtas)

SO- Sistema Operacional

SPA- Single Page Application (Aplicação de Página Única)

SQL - Structured Query Language (Linguagem de Consulta Estruturada)

TIC - Tecnologias de Informação e Comunicação

UC - Use Case (Caso de Uso)

UML - Unified Modelling Language (Linguagem de Modelagem Unificada)

XAMPP- Cross-Platform (X), Apache (A), MariaDB/MySQL (M), PHP (P), Perl (P)

# **RESUMO**

O presente trabalho apresenta o desenvolvimento de um sistema web de agendamento de consultas para o Hospital Geral de Luanda, com o objectivo de modernizar e optimizar o processo de marcação de atendimentos médicos. Actualmente, o hospital enfrenta desafios significativos relacionados com a gestão manual de consultas, resultando em longas filas de espera, sobrecarga no atendimento presencial e dificuldades na organização dos horários médicos. A pesquisa adoptou uma abordagem qualitativa, de natureza aplicada e com objectivos exploratórios e descritivos, utilizando como procedimentos técnicos a pesquisa bibliográfica, estudo de caso, levantamento de dados através de questionários e entrevistas. Para o desenvolvimento do sistema, foi adoptada a metodologia ágil Scrum, que permitiu entregas incrementais e iterativas, garantindo a participação activa dos utilizadores e a adaptação às necessidades identificadas. O sistema foi modelado utilizando a Linguagem de Modelagem Unificada (UML), através de diagramas de casos de uso, sequência, actividade, classe e objectos, proporcionando uma visão clara da estrutura e comportamento do sistema. As tecnologias seleccionadas incluem PHP e Laravel para o backend, React.js para o frontend, MySQL como sistema de gestão de base de dados, e ferramentas como Visual Studio Code, Git/GitHub, Canva e Astah para apoio ao desenvolvimento. O sistema contempla funcionalidades essenciais como cadastro de pacientes e médicos, agendamento, edição e cancelamento de consultas, notificações automáticas, visualização de horários e geração de relatórios. Os requisitos não funcionais garantem segurança através de controlo de acesso, usabilidade com interface intuitiva, desempenho adequado e disponibilidade 24 horas. Os resultados esperados incluem a redução significativa do tempo de espera dos pacientes, melhoria na organização dos serviços hospitalares, diminuição da sobrecarga administrativa e aumento da satisfação dos utilizadores. Este trabalho contribui tanto para o âmbito académico, permitindo a aplicação prática dos conhecimentos adquiridos, quanto para o âmbito social, promovendo o acesso mais eficiente e transparente aos serviços de saúde da população angolana.

Palavras-chave:

# **ABSTRACT**

This study presents the development of a web-based appointment scheduling system for the Luanda General Hospital, aimed at modernizing and optimizing the medical appointment booking process. Currently, the hospital faces significant challenges related to manual appointment management, resulting in long waiting queues, overload in face-to-face service, and difficulties in organizing medical schedules. The research adopted a qualitative approach, applied in nature with exploratory and descriptive objectives, using bibliographic research, case study, and data collection through questionnaires and interviews as technical procedures. For system development, the Scrum agile methodology was adopted, enabling incremental and iterative deliveries, ensuring active user participation and adaptation to identified needs. The system was modeled using Unified Modeling Language (UML) through use case, sequence, activity, class, and object diagrams, providing a clear view of the system's structure and behavior. Selected technologies include PHP and Laravel for the backend, React.js for the frontend, MySQL as the database management system, and tools such as Visual Studio Code, Git/GitHub, Canva, and Astah to support development. The system includes essential features such as patient and doctor registration, appointment scheduling, editing and cancellation, automatic notifications, schedule viewing, and report generation. Non-functional requirements ensure security through access control, usability with an intuitive interface, adequate performance, and 24-hour availability. Expected results include significant reduction in patient waiting time, improvement in hospital service organization, decrease in administrative overload, and increased user satisfaction. This work contributes to both the academic sphere, allowing practical application of acquired knowledge, and the social sphere, promoting more efficient and transparent access to healthcare services for the Angolan population.

Keywords:

**ÍNDICE**

[**EPÍGRAFE** I](#_Toc216178521)

[**DEDICATÓRIA** II](#_Toc216178522)

[**AGRADECIMENTOS** III](#_Toc216178523)

[**LISTA DE FIGURAS** IV](#_Toc216178524)

[**LISTA DE TABELAS** V](#_Toc216178525)

[**LISTA DE ABREVIAÇÕES E SIGLAS** VI](#_Toc216178526)

[**RESUMO** VII](#_Toc216178527)

[**ABSTRACT** VIII](#_Toc216178528)

[**INTRODUÇÃO** 2](#_Toc216178529)

[**O PROBLEMA** 3](#_Toc216178530)

[**JUSTIFICATIVA** 3](#_Toc216178531)

[**OBJECTIVOS DE ESTUDO** 4](#_Toc216178532)

[**Objectivo Geral** 4](#_Toc216178533)

[**Objectivos Específicos** 4](#_Toc216178534)

[**Estrutura do trabalho** 4](#_Toc216178535)

[**CAPÍTULO I – FUNDAMENTAÇÃO TEÓRICA** 5](#_Toc216178536)

[**1.1.** **Agendamento de Consultas** 5](#_Toc216178537)

[**1.1.1.** **Benefícios de um Sistema na Gestão de Consultas** 5](#_Toc216178538)

[**1.2.** **Sistemas de Informação** 6](#_Toc216178539)

[**1.3.** **Sistemas Web** 7](#_Toc216178540)

[**1.3.1.**  **Características do processo de um sistema web** 8](#_Toc216178541)

[**1.3.2.**  **Principais vantagens de sistemas web** 8](#_Toc216178542)

[**1.3.3.**  **Principais desvantagens de sistemas web** 9](#_Toc216178543)

[**1.4.** **Base de Dados** 9](#_Toc216178544)

[**1.4.1.** **Sistemas de Gerenciamento de Bases de Dados (SGBD)** 10](#_Toc216178545)

[**1.4.2.** **PHP (Hypertext Preprocessor)** 11](#_Toc216178546)

[**1.5.** **Engenharia de Requisito** 12](#_Toc216178547)

[**1.6.** **Linguagem de Programação** 13](#_Toc216178548)

[**1.6.1.** **Plataformas de Desenvolvimento** 13](#_Toc216178549)

[**1.7.** **Processo de desenvolvimento de sistemas** 14](#_Toc216178550)

[**1.8.** **Processo iterativo e incremental** 15](#_Toc216178551)

[**CAPÍTULO II – METODOLOGIA** 20](#_Toc216178552)

[**2.1** **Tipo de Pesquisa** 20](#_Toc216178553)

[**2.1.1.** **Quanto a sua natureza** 20](#_Toc216178554)

[**2.1.2.** **Quanto aos objectivos** 20](#_Toc216178555)

[**2.1.3.** **Quanto aos procedimentos técnicos** 21](#_Toc216178556)

[**2.2.** **Métodos e Procedimentos** 21](#_Toc216178557)

[**2.3.** **Ferramentas utilizadas e Tecnologias** 22](#_Toc216178558)

[**2.4.** **Metodologia de Desenvolvimento de Software** 22](#_Toc216178559)

[**2.5.** **Levantamento e Analise de Requisitos** 23](#_Toc216178560)

[**2.5.1.** **Requisitos Funcionais do Sistema** 23](#_Toc216178561)

[**2.5.2.** **Requisitos Não Funcionais do Sistema** 24](#_Toc216178562)

[**2.6.** **Modelação do Sistema com UML** 24](#_Toc216178563)

[**2.6.1.** **Diagrama de casos de uso** 25](#_Toc216178564)

[**2.6.1.1.Descrição dos principais casos de uso do sistema** 26](#_Toc216178565)

[**2.6.2.** **Diagramas de Actividade** 28](#_Toc216178566)

[**2.6.3.** **Diagramas de Sequência** 30](#_Toc216178567)

[**2.6.4.** **Diagramas de Classe** 31](#_Toc216178568)

[**2.6.5.** **Diagramas de Objectos** 32](#_Toc216178569)

[**2.7.** **Protótipos e Interface do Sistema** 32](#_Toc216178570)

[**2.8.** **Cronograma de Desenvolvimento** 32](#_Toc216178571)

[**CAPÍTULO III – APRESENTAÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS OBTIDOS** 33](#_Toc216178572)

[**REFERÊNCIAS BIBLIOGRÁFICAS** 34](#_Toc216178573)

# **INTRODUÇÃO**

O presente trabalho tem como temática, desenvolvimento de um sistema web de agendamento de consultas para o Hospital Geral De Luanda no IIº Trimestre do ano de 2025/2026.

Com o aparecimento das tecnologias de informação e comunicação (TIC), actualmente a gerência das empresas é feita através de máquinas electrónicas denominado por computador, onde estas máquinas possuem módulos determinados para administração (Pereira, 1998).

Segundo Souza et al. (2020), um sistema de agendamento de consultas é uma ferramenta digital que tem como objectivo optimizar o processo de marcação de atendimentos, permitindo que pacientes escolham datas e horários disponíveis de forma rápida e prática, sem depender exclusivamente do contato telefônico ou presencial.

Os sistemas de agendamento online possibilitam uma gestão mais eficiente dos atendimentos, reduzindo o tempo de espera e evitando a superlotação dos serviços de saúde. Essa tecnologia traz benefícios tanto para os profissionais de saúde quanto para os pacientes: reduz filas, evita conflitos de agenda, aumenta a taxa de comparecimento e melhora a experiência do usuário. Quando integrado a funcionalidades como lembretes automáticos, histórico de atendimentos e gestão de disponibilidade, o sistema transforma-se em um recurso indispensável para clínicas, hospitais e consultórios que desejam modernizar a sua gestão e oferecer um serviço mais eficiente (Oliveira & Martins, 2019).

O Hospital Geral de Luanda (HGL) é uma das principais instituições de saúde pública de Angola, responsável por atender diariamente um elevado número de pacientes provenientes de diversas regiões da capital. Entretanto, o processo de agendamento de consultas ainda é realizado de forma manual, o que ocasiona atrasos, longas filas e, muitas vezes, perda de informações.

Com o crescimento da procura por serviços médicos e a necessidade de melhorar a gestão hospitalar, torna-se essencial adoptar soluções tecnológicas que promovam maior eficiência e qualidade no atendimento. Conforme algumas pesquisas, após a pandemia de COVID-19, os serviços de saúde em todo o mundo aceleraram a adoção de plataformas digitais para garantir acesso remoto e optimizar fluxos de trabalho. Nesse contexto, o desenvolvimento de um sistema web de agendamento de consultas surge como uma alternativa viável para automatizar o processo, reduzir o tempo de espera dos pacientes e optimizar as tarefas administrativas do hospital.

## **O PROBLEMA**

No contexto actual do Hospital Geral de Luanda, o processo de marcação de consultas apresenta diversos desafios, tais como longas filas de espera, sobrecarga no atendimento presencial, dificuldades de comunicação entre pacientes e profissionais de saúde e ausência de um sistema eficaz de controlo e organização dos horários médicos.

Essas limitações comprometem a eficiência dos serviços prestados e afectam negativamente a experiência dos pacientes, segundo alguns autores, isso demonstra maior satisfação quando têm acesso a plataformas digitais que facilitam o agendamento e reduzem o tempo de espera.

Com base ao problema acima exposto, levanta-se a seguinte questão científica:

* Como o desenvolvimento de um sistema web de agendamento de consultas pode optimizar o processo e a eficiência na gestão dos pacientes no Hospital Geral de Luanda no II Trimestre do ano 2025/2026?

## **JUSTIFICATIVA**

A escolha do tema justifica-se pela necessidade de modernizar a gestão hospitalar e optimizar o atendimento no Hospital Geral de Luanda, que actualmente enfrenta desafios significativos, como o elevado fluxo de pacientes, longas filas de espera e limitações nos seus processos administrativos. Esses factores comprometem a qualidade do serviço prestado à população e evidenciam a importância da adoção de soluções tecnológicas eficientes.

No âmbito académico, o desenvolvimento deste projecto oferece-nos a oportunidade de aplicar, na prática, os conhecimentos adquiridos ao longo da formação, aliando teoria e prática na resolução de problemas reais e relevantes para a sociedade. Este trabalho também promove o desenvolvimento de competências técnicas e analíticas fundamentais para a formação profissional.

No âmbito social, a relevância do estudo está na sua contribuição directa para a melhoria do acesso da população aos serviços de saúde. A digitalização do processo de marcação de consultas garante maior organização, transparência e eficiência no atendimento, especialmente num contexto pós-pandemia, em que o uso de plataformas digitais se tornou essencial para a continuidade dos serviços de saúde de forma segura e acessível.

O estudo mostra-se pertinente por abordar problemas reais da administração hospitalar, como a desorganização do agendamento manual e a falta de integração entre sectores. A proposta apresenta relevância técnica e social, contribuindo para a eficiência interna da instituição e para a satisfação dos utentes, assegurando um serviço mais rápido, seguro e transparente.

## **OBJECTIVOS DE ESTUDO**

Tendo em conta o problema levantado, pretendemos alcançar os seguintes objectivos:

### **Objectivo Geral**

* Desenvolver um sistema web de agendamento de consultas para o Hospital Geral de Luanda no II trimestre do ano de 2025/2026, com vista a melhorar a gestão do atendimento e aumentar a eficiência do processo de gestão dos pacientes no Hospital.

### **Objectivos Específicos**

* Recolher todas as informações necessárias relacionadas ao processo de agendamento de consultas;
* Definir as tecnologias a serem utilizadas para o desenvolvimento do sistema;
* Criar uma interface intuitiva para agendamentos;
* Efectuar a modelação do projecto, no sentido de visualizar como será feita a construção do sistema web;
* Testar as funcionalidades do sistema web, com vista verificar o nível de segurança do mesmo.

## **Estrutura do trabalho**

O trabalho está estruturado em três (3) capítulos tal como se segue abaixo:

**Capítulos 1** - este capítulo trata da fundamentação teórica, onde são descritas a apresentação do sistema, sua definição, bem como o suporte teórico detalhado, que justifica a opção pelas tecnologias optadas.

**Capítulos 2** -este capítulo apresenta a análise metodológica (tipo ou método de pesquisa efectuada, seguido do campo de estudo e suas delimitações) bem como apresentamos os diagramas UML em função da análise que foi feita, fizemos a descrição das tecnologias envolvidas e a sua razão, e mais outros factores importantes.

**Capítulos 3** -são apresentados os resultados da análise feita anteriormente, forma funcional, a interface juntamente com alguma explicação do funcionamento do sistema.

# **CAPÍTULO I – FUNDAMENTAÇÃO TEÓRICA**

Este capítulo apresenta os conceitos e teorias que fundamentam o desenvolvimento deste trabalho, fornecendo a base necessária para a compreensão dos temas abordados e para a análise dos resultados obtidos.

## **Agendamento de Consultas**

O agendamento de consultas tem-se tornado um elemento essencial na modernização dos serviços de saúde, reflectindo directamente na qualidade e eficiência do atendimento prestado aos pacientes.

Segundo a Clínica Renata Zac (2024), o agendamento de consultas é um processo fundamental na organização do atendimento médico, permitindo que os pacientes marquem horários com profissionais de saúde de forma prática e estruturada. Essa sistematização evita longas esperas, reduz a sobrecarga das clínicas e garante que o atendimento ocorra de maneira mais coordenada e satisfatória.

Parente et al., (2020), reforça que, um agendamento de consultas é o processo onde os pacientes agendam um horário para serem atendidos por um profissional de saúde, como médicos, dentistas ou fisioterapeutas. Esse procedimento é essencial para garantir que o paciente receba o atendimento necessário no momento adequado.

Segundo Ala & Chen (2022, p. 1), "o agendamento de consultas visa construir um sistema que optimize um padrão de qualidade específico em uma aplicação de saúde de agendamento de tarefas". Isso significa que o objectivo não é apenas marcar horários, mas garantir a qualidade do atendimento. O sistema substituirá processos manuais, como cadernos, planilhas ou ligações telefônicas por uma experiência mais prática, rápida e segura tanto para a equipe do hospital quanto para os pacientes.

O processo de marcação de consulta pode variar de acordo com a instituição de saúde e o tipo de serviço oferecido”. Em algumas clínicas o agendamento de consultas é feito geralmente por telefone, em hospitais públicos o agendamento é feito pessoalmente ou mesmo por um sistema online para agendar o horário desejado, o que no caso do hospital geral de Luanda é ainda não dispõe desse serviço online de marcação de consultas. É possível escolher o tipo de consulta, o especialista disponível na área selecionada, cancelar ou reagendar quando haver algum imprevisto (Oleskovicz & Pedroso, 2023).

## **Benefícios de um Sistema na Gestão de Consultas**

Ao integrar sistemas tecnológicos na gestão de agendamentos, observa-se uma transformação significativa nas rotinas das instituições de saúde.

A adoção de um sistema informatizado de agendamento traz múltiplos benefícios tanto para os profissionais quanto para os pacientes (Ravanello, 2025). Entre os principais benefícios, o autor menciona os seguintes:

1. **Redução de falta:** possibilitada pelo envio de lembretes automáticos via e-mail, SMS ou WhatsApp, que reduzem as taxas de ausência em até 38%. Essa automatização não apenas melhora a organização da agenda, mas também aumenta a eficiência operacional da clínica.
2. **Optimização do tempo administrativo:** é outro ganho relevante, uma vez que a automatização do processo reduz a carga de trabalho da equipa médica, que deixa de lidar com o agendamento manual por telefone.
3. **Flexibilidade oferecida aos pacientes:** que passam a poder marcar, remarcar ou cancelar consultas a qualquer hora do dia, sem restrições de horário de funcionamento. Essa flexibilidade amplia o acesso aos serviços e melhora a experiência do paciente, tornando o processo mais conveniente e transparente.
4. **Capacidade de Gerenciar** **salas e recursos clínicos:** o que permite uma alocação mais eficiente de equipamentos, profissionais e horários. Ao mesmo tempo, a geração de relatórios e métricas automáticas possibilita análises estratégicas sobre a demanda, cancelamentos e picos de atendimento, como também informações cruciais para a tomada de decisões gerenciais. Assim, os sistemas de agendamento não apenas automatizam tarefas, mas também fornecem inteligência operacional à gestão de clínicas e hospitais.

## **Sistemas de Informação**

Do ponto de vista teórico, a compreensão desses sistemas pode ser aprofundada a partir do conceito de Sistemas de Informação (SI). Segundo Grigoryan (2024), os sistemas de informação são combinações estruturadas de hardware, software, infraestrutura e pessoal treinado, voltadas para apoiar o planejamento, controle e tomada de decisões nas organizações. Essa definição mostra que o sistema de agendamento é, na verdade, uma aplicação prática dos princípios dos SI, voltada especificamente para o sector da saúde.

Complementando essa visão, O’Brien (2000) descreve que os sistemas de informação são compostos por seis elementos principais, que interagem entre si para colectar, processar, armazenar e distribuir informações, sendo elas:

* **Pessoas:** são os utilizadores e especialistas que operam o sistema, interpretam os dados e tomam decisões. Incluem operadores, analistas, administradores e gestores.
* **Hardware:** refere-se aos dispositivos físicos (computadores, servidores, impressoras e redes) que executam os programas e armazenam dados.
* **Software:** são os programas e aplicativos que controlam o hardware e executam tarefas (como sistemas operacionais, bases de dados e softwares empresariais).
* **Dados:** o recurso central do sistema. São as informações brutas (números, textos, imagens, registros) que são processadas para gerar conhecimento.
* **Redes:** permitem a comunicação entre computadores e usuários, tanto internamente (intranet) como externamente (internet).
* **Processos:** representam o conjunto de actividades e regras que definem como os dados são tratados, armazenados e distribuídos.

No contexto do agendamento de consultas, esses componentes se manifestam de forma clara, sendo que as pessoas (usuários e gestores) utilizam o software (sistema web) que opera em um hardware conectado por redes de comunicação, que manipula dados de pacientes e horários, conforme processos predefinidos de marcação e cancelamento. Essa integração harmoniosa de componentes é o que possibilita o funcionamento eficaz do sistema.

Para Ferreira (2007), um SI bem construído traz consigo benefícios, como:

* Maior eficiência, produtividade e segurança;
* Carga de trabalho reduzido, custos reduzidos;
* Maior precisão, menos erros;
* Melhores decisões com base em informação de qualidade, mais oportunidades;
* Vantagens competitivas: melhor serviço, melhor qualidade.

## **Sistemas Web**

Segundo McDowell e Panchall (2010, pp 313-349), “Sistemas web são definidos como sistemas que operam pela internet ou intranet usando tecnologias da web, normalmente empregando uma arquitectura cliente-servidor onde a internet serve como espinha dorsal para comunicação entre clientes e um servidor web central”. Em outras palavras, é um software hospedado na internet, que permite que os usuários interajam com ele de qualquer lugar desde que tenham uma conexão à internet. No contexto hospitalar, não só facilitaria a comunicação entre o paciente e o profissional de saúde, mas também a coordenação entre os diversos departamentos do hospital.

## **1.3.1. Características do processo de um sistema web**

Os sistemas web apresentam características distintas que os tornam adequados para aplicações hospitalares. Sendo elas:

* **Arquitectura Cliente-Servidor**: são navegadores(clientes) que se comunicam com os servidores centrais através de protocolos HTTP/HTTPS.
* **Compatibilidade Multiplataforma:** ele funciona em diversos sistemas operacionais (SO) como Windows, Mac e Linux, e também em dispositivos desktop, tablet e mobile sem a necessidade de desenvolvimento separado.
* **Centralização de dados:**  as informações são armazenadas num servidor central, facilitando backup, manutenção e garantindo consistência dos dados.
* **Interação em tempo real:** ele possui uma interface dinâmica que processa informações e apresenta resultados actualizados instantaneamente.
* **Páginas web dinâmicas:**  são páginas web que geram o conteúdo dinamicamente em tempo real, baseado em entradas do utilizador, dados de base de dados. Ao contrário de páginas estáticas que mostram sempre o mesmo conteúdo, páginas dinâmicas podem apresentar informações diferentes para cada utilizador ou em cada acesso.

Essas características são extremamente valiosas em ambientes hospitalares, onde profissionais necessitam aceder informações de diferentes localizações.

## **1.3.2. Principais vantagens de sistemas web**

As vantagens dos sistemas web são numerosas e significativas, por isso selecionamos as principais vantagens que são:

* **Facilidade de manutenção:** as actualizações e correções são implementadas centralmente no servidor, beneficiando todos os utilizadores simultaneamente.
* **Colaboração em tempo real:** múltiplos profissionais podem aceder e actualizar informações simultaneamente.
* **Backup e recuperação:**  os dados centralizados facilitam processos de backup automático e recuperação de desastres.
* **Actualizações instantâneas:** as novas funcionalidades ficam disponíveis imediatamente para todos os utilizadores após implementação no servidor.
* **Escalabilidade:** facilita a expansão do sistema conforme crescimento da demanda, sem investimentos substânciais em infraestrutura.

Para o hospital, estes benefícios permitem que os funcionários e os seus pacientes se sintam seguros em passar suas informações sem terem medo de perdê-las.

## **1.3.3. Principais desvantagens de sistemas web**

Apesar dos benefícios, os sistemas web apresentam limitações que devem ser consideradas como:

* **Dependência de conexão à internet:**  o sistema não funciona sem conexão à internet; falhas de rede comprometem o acesso.
* **Segurança:**  sistemas expostos na internet apresentam maior superfície de ataque para ameaças cibernéticas.
* **Variação de desempenho:**  a performance depende da qualidade da conexão à internet e especificações do servidor; pode ser mais lento.
* **Compatibilidade entre navegadores:**  diferenças na interpretação do código entre navegadores podem causar inconsistências na aparência ou funcionalidade.
* **Acesso limitado do hardware:** sistemas web têm acesso restrito aos recursos do dispositivo como a câmera e o GPS, se diferenciando bastante das aplicações nativas.

Todos os sistemas possuem benefícios e malefícios, no entanto, essas desvantagens não tornam o nosso sistema menos eficiente.

## **Base de Dados**

Segundo Elmasri & Navathe (2016), uma base de dados é uma coleção organizada de dados relacionados que possuem significado implícito. Esses dados podem representar informações como nomes, contatos e endereços, geralmente armazenados em dispositivos digitais e geridos por softwares específicos.

Segundo Silva (1994) reforça a ideia, definindo a base de dados como um conjunto de arquivo que contém todas as informações indispensáveis para a organização. Esses arquivos podem ser guardados em algum dispositivo magnético de armazenamento e a sua manipulação é feita através de um sistema de gestão de base de dados (SGBD).

De acordo com Alura (2024), existem diversos tipos de bases de dados, cada uma projectada para contextos distintos, sendo elas:

* **Relacionais (RDBMS):** organizam dados em tabelas com chaves primárias e estrangeiras (ex.: MySQL, PostgreSQL).
* **NoSQL:** voltados a dados não estruturados, como documentos (MongoDB) ou grafos (Neo4j).
* **Em Memória:** armazenam dados directamente na RAM, como Redis e Memcached.
* **Distribuídas:** espalham informações por múltiplos servidores, como Hadoop e HBase.
* **Em Nuvem:** hospedadas em plataformas como AWS, Azure e Google Cloud.  
  Outros tipos incluem bases de dados em colunas, de séries temporais, de tempo real, blockchain e de texto completo.

Segundo Nutanix (2023), as bases de dados são compostas por cinco componentes principais:

1. **Hardware** – servidores e dispositivos físicos;
2. **Software** – o SGBD que gerencia o acesso;
3. **Dados** – as informações propriamente ditas;
4. **Linguagem de acesso** – como o SQL;
5. **Procedimentos** – regras e normas de utilização.

As bases de dados são fundamentais na era moderna por permitirem armazenamento estruturado, recuperação eficiente, consistência, segurança e suporte à tomada de decisões. Além disso, garantem backup, compartilhamento controlado, transações seguras e integração entre sistemas (Alura, 2024).

## **Sistemas de Gerenciamento de Bases de Dados (SGBD)**

Conforme Caldeira (2023), um SGBD é o sistema responsável por armazenar, manipular e gerenciar dados de forma segura e eficiente. Ele permite criar, consultar, alterar e excluir informações, além de gerir utilizadores e relacionamentos entre tabelas.

Damas (2005) define SGBD como sendo uma aplicação ou um conjunto de aplicações informáticas que fornece a interface entre a BD e o utilizador, permitindo definir, aceder e gerir os dados existentes numa BD.

As funções de um SGBD, segundo Pereira (2023), são:

* 1. Reunir informações computacionais em arquivos.
  2. Fornecer aos usuários a capacidade de controlar o acesso de leitura / gravação;
  3. Definir a geração de relatórios e realizar procedimentos de análise dos dados.
  4. Criar arquivos de texto que permitem o armazenamento de dados.
  5. Fornecer as condições para o usuário criar arquivos de dados.

O SGBD foi desenvolvido para ultrapassar alguns dos problemas e limitações com que os utilizadores de sistema enfrentavam em ficheiros de dados. O seu objectivo é fornecer um ambiente apropriado para aceder e armazenar informação na BD de forma fiável e eficiente (Damas, 2005).

De acordo com o mesmo autor, alguns requisitos fundamentais de um SGBD são:

* **Eficiência-** ser capaz de aceder, processar e alterar grandes volumes de dados de forma eficiente;
* **Robustez-** manter os dados de forma consistente, mesmo após falhas de hardware ou software;
* **Controlo de acessos-** controlar o acesso dos utilizadores;
* **Persistência-** permitir que os dados armazenados durem longo período.

A Higher School of Network (2023), podem ser classificados por modelo, sendo eles:

* **Relacional:** estrutura em tabelas com chaves;
* **Orientado a Objectos:** trabalha com dados complexos através de objectos;
* **Hierárquico:** organiza dados em forma de árvore;
* **Em Rede:** utiliza múltiplos caminhos de acesso;
* **Multidimensional:** voltado à análise de dados e *business intelligence*;
* **NoSQL:** ideal para grandes volumes de dados não estruturados;
* **Na Memória:** executa operações directamente na RAM, aumentando o desempenho.

De acordo com Juliano (2014), os requisitos essenciais de um SGBD incluem:

* **Segurança dos dados**;
* **Integridade**, garantindo dados válidos e consistentes;
* **Controlo de Concorrência**, permitindo múltiplos acessos;
* **Recuperação e Tolerância a Falhas**, através de *backup* e logs de transações.

Ainda Obermeier (2025) acrescenta que, ao escolher um SGBD, devem ser considerados fatores como:

1. Modelo de dados (relacional ou NoSQL);
2. Requisitos de segurança;
3. Disponibilidade e backup;
4. Proteção e conformidade legal;
5. Multiacesso e integração;
6. Desempenho e escalabilidade;
7. Usabilidade e suporte;
8. Custo total de propriedade.

## **PHP (Hypertext Preprocessor)**

Segundo o Manual do PHP (2025), o PHP é uma linguagem de script *open source* voltada ao desenvolvimento web, criada em 1994 por Rasmus Lerdorf. Inicialmente chamada Personal Home Page Tools, evoluiu para PHP/FI e, em 1997, tornou-se PHP: Hypertext Preprocessor após ser reescrita por Andi Gutmans e Zeev Suraski, criadores do Zend Engine.

O PHP 5 (2004) introduziu a programação orientada a objectos, o PHP 7 (2015) trouxe alta performance e tipagem escalar, e o PHP 8 (2020) implementou o motor JIT, aumentando a eficiência. Actualmente, o PHP é amplamente utilizado em sistemas como WordPress, Drupal e Joomla, e é compatível com SGBDs como MySQL e MariaDB.

Segundo Workana (2023), o PHP distingue-se por poder ser embutido directamente em HTML, por ser fácil de aprender e compatível com a maioria dos servidores e bases de dados. É também open source, versátil, SEO-friendly e continuamente atualizado pela comunidade.

Estrella (2025) complementa que o PHP pode ser integrado a um SGBD através de extensões como MySQLi e PDO, permitindo interações dinâmicas com dados (inserções, consultas e atualizações) via comandos SQL.

## **Engenharia de Requisito**

Os requisitos são definidos durante os estágios iniciais do desenvolvimento de um sistema, como uma especificação do que dever ser implementado. Eles descrevem um comportamento, uma propriedade ou atributo do sistema. Podem também definir restrições ao processo de desenvolvimento do sistema (Sommerville, 2004).

Os requisitos de software são classificados em requisitos funcionais e requisitos não funcionais (Sommerville, 2004).

1. **Requisitos funcionais:** Define as funções que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como deve se comportar em determinadas situações. Os requisitos funcionais devem descrever detalhadamente os serviços e a funcionalidade que devem ser fornecidas pelo sistema.
2. **Requisitos não funcionais:** São requisitos que expressam condições que o software deve atender ou qualidades específicas que o software deve ter. Em vez de informar o que o sistema fará, os requisitos não-funcionais colocam restrições no sistema.

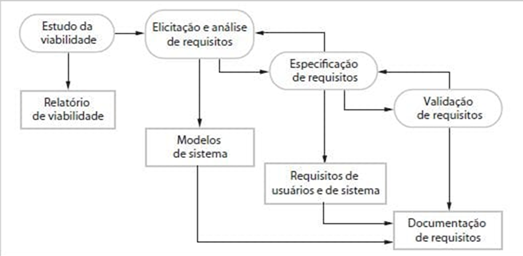


Figura 1 - Actividade da Engenharia de Requisitos

Fonte: Sommerville (2011, p. 24).

## **Linguagem de Programação**

Utilizada no desenvolvimento de software (desktop, mobile, web, games, etc.), análise de dados, IA, robótica, automação, etc. Linguagem de programação define-se como conjunto de instruções descritas pelo programador de forma simples, legível, com tipo e estrutura de dados, verificação de tipos e manipulação de excepções que lhe permitem dar instruções ao computador (Sebesta, 2016, pp.8-15).

## **Plataformas de Desenvolvimento**

* **HTML (HyperText Markup Language):** é o bloco de construção mais básico da web. Ducket (2011), define o significado e a estrutura do conteúdo da página web. Criado por Tim Berners-Lee em 1993, o HTML, para dar mais robustez às páginas web trabalha com as seguintes tecnologias:
* **CSS (Cascading Style Sheets):** proposta por Hakon Wium Lie e Bert Bos em 1994, o **CSS,** é uma linguagem de folha de estilo usada para descrevera apresentação de um documento em HTML, estilizar os elementos HTML, layouts, etc. (Ducket, 2011)**.**
* **JS (JavaScript):** criado por Brendan Eich, desenvolvedor da Netscape (agora Mozilla), em 1995, o **JS** pode definir-se de forma genérica como uma linguagem de script focada em especificar o comportamento das páginas web, manipulando elementos e respondendo a eventos (Eich, 1995). Mas, no entanto, actualmente o JS já é utilizado do lado do servidor no desenvolvimento de sistemas web, APIs etc, por intermédio do Node.js, sendo este um interpretador de JavaScriptque funciona de forma independente do navegador, criado por Ryan Dahl em 2009 com este propósito (Lima, 2019).
* **PHP (Hypertext Preprocessor):** criado por Rasmus Lerdorf, em 1995, o PHP, também uma linguagem de script, que roda directamente a partir do servidor, independente de uma plataforma. Precisando apenas de um servidor web como o **Apache**, um editor de texto e um servidor de base de dados como: MySQL, SQL Server, etc. (Lerdorf, 2016).
* **React.js:** desenvolvido em 2014 pelo Facebook (actual Meta), o React.js, comummente chamado apenas de Reacté uma biblioteca JavaScript constituída por componentes (parte da interface do usuário), que são funções JavaScript, com foco na criaçãode interfaces e SPAs (Single Page Application).

## **Processo de desenvolvimento de sistemas**

Segundo Sommerville (2011) um processo de software é um conjunto de actividades relacionadas que levam à produção de um produto de software. Essas actividades podem envolver o desenvolvimento de software a partir do zero em uma linguagem padrão de programação como Java ou C.

Segundo o mesmo autor, existem muitos processos de software diferentes, mas todos devem incluir quatro actividades fundamentais para a engenharia de software:

* 1. **Especificação de software -** a funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
  2. **Projecto e implementação de software -** o software deve ser produzido para atender às especificações.
  3. **Validação de software -** o software deve ser validado para garantir que atenda às demandas do cliente.
  4. **Evolução de software -** o software deve evoluir para atender às necessidades de mudança dos clientes.

## **Processo iterativo e incremental**

Processo iterativocorresponde à ideia de melhorar (ou refinar) pouco-a-pouco o sistema. Um excelente exemplo de aplicação do processo iterativo encontra-se no trabalho artístico, em que o resultado final de uma obra de arte sofre inúmeras iterações (algumas das quais, por vezes, destrutivas para o seu resultado final). O âmbito do sistema não é alterado, mas o seu detalhe vai aumentando em iterações sucessivas.

O desenvolvimento iterativo apresenta ainda outras vantagens significativas para o desenvolvimento de software (Kruchten, 2000):

* Os riscos e dúvidas com maior importância são identificados no início do processo, nas primeiras iterações, quando é possível tomar medidas para os corrigir.
* Esta abordagem encoraja a participação activa dos utilizadores de modo a identificar os verdadeiros requisitos do sistema.
* A execução de testes contínuos e iterativos permitem uma avaliação objectiva do estado do projecto.
* As inconsistências entre a análise, o desenho e a implementação são identificadas atempadamente.
* O esforço dos diversos elementos da equipa é distribuído ao longo do tempo.
* A equipa pode aprender com experiências anteriores e melhorar continuamente o processo.

O processo incremental corresponde à ideia de aumentar (ou alargar) pouco-a-pouco o âmbito do sistema. Uma boa imagem para este atributo é a de uma mansão que foi construída por sucessivos incrementos a partir de uma primeira casa com apenas duas divisões.

Este processo é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido (Figura 1). Actividades de especificação, desenvolvimento e validação são intercaladas, e não separadas, com rápido feedback entre todas as actividades.

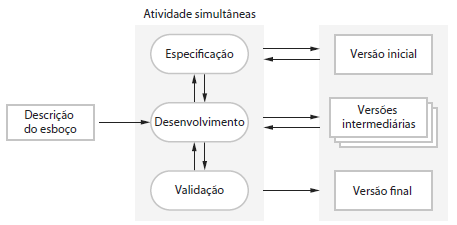


Figura 2 - Processo incremental

Fonte: Sommerville (2011, p.22)

Processo iterativo e incremental para que a equipa envolvida possa refinar e alargar pouco-a-pouco a qualidade, detalhe e âmbito do sistema envolvido. Por exemplo, numa primeira iteração deve-se identificar a visão global e determinar a viabilidade económica do sistema; efectuar a maior parte da análise e um pouco de desenho e implementação.

Numa segunda iteração, deve-se concluir a análise, fazer uma parte significativa do desenho, e um pouco mais de implementação. Numa terceira iteração, deve-se concluir o desenho, fazer-se parte substancial da implementação, testar e integrar um pouco; etc. Ou seja, a principal consequência da aproximação iterativa é que os produtos finais de todo o processo vão sendo amadurecidos e completados ao longo do tempo, mas cada iteração produz sempre um conjunto de produtos finais.

# **CAPÍTULO II – METODOLOGIA**

Para Severino (2007), a metodologia de pesquisa refere-se ao conjunto de processos organizados e sistematizados que orientam a investigação científica. Ela enfatiza que a escolha da metodologia depende do tipo de pesquisa (exploratória, descritiva e explicativa) e dos métodos utilizados para colectar e analisar dados.

## **Tipo de Pesquisa**

Segundo Lakatos & Marconi (2007, p. 157), a pesquisa pode ser considerada “um procedimento formal com método de pensamento reflexivo que requer um tratamento científico e se constitui no caminho para se conhecer a realidade ou para descobrir verdades parciais”. Isso significa que ela serve não só para procurar a verdade, mas também para descobrir respostas para perguntas ou soluções para problemas levantados.

A pesquisa escolhida para desenvolver o nosso projecto é a bibliográfica, sendo ela um processo metodológico que colecta e analisa informações já publicadas sobre um determinado tema, usando como base livros, artigos científicos, teses, e outros materiais.

## **Quanto a sua natureza**

Quanto a abordagem, a nossa pesquisa é qualitativa. É qualitativa porque os dados foram analisados indutivamente por meio de um questionário e entrevistas com os moradores residentes no condomínio, pois não traduz em números as opiniões e informações dadas pelos sujeitos da pesquisa (Severino, 2007).

## **Quanto aos objectivos**

Quanto aos objectivos, utilizou-se a pesquisa aplicada, exploratória e desenvolvimento tecnológico. Para Gil (2010), a pesquisa descritiva descrever factos observados sem interferir neles, envolvendo o uso de técnicas padronizadas de colecta de dados como questionários. É a descrição das características de determinada população ou fenómeno ou, então, o estabelecimento de relações entre variáveis.

Exploratóriaporque tem como objectivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a constituir hipóteses. Pode-se dizer que estas pesquisas têm como objectivo principal o aprimoramento de idéias ou a descoberta de intuições (Gil, 2010, p. 41).

Estes dois tipos de pesquisa contribuíram para que o trabalho fosse o mais eficiente possível, desde artigos usados até os dados colectados para o desenvolvimento dele.

## **Quanto aos procedimentos técnicos**

Quanto aos procedimentos técnicos utilizou-se a pesquisa bibliográfica. De acordo com Lakatos & Marconi (2017) a pesquisa bibliográfica é desenvolvida a partir de material já elaborado, constituído principalmente de livros e artigos científicos.

Estudo de caso:oHospital Geral de Luanda é o nosso estudo de caso, é onde o sistema será implementado. Segundo Yin (2015, p. 15), “estudo de caso é a estratégia escolhida ao se examinarem acontecimentos contemporâneos, mas quando não se pode manipular comportamentos relevantes”

Levantamento:uso de questionários e/ou entrevistas com os funcionários e pacientes do hospital. Procede-se à solicitação de informações a um grupo significativo de pessoas acerca do problema estudado para, em seguida, mediante análise quantitativa, obterem-se as conclusões correspondentes aos dados colectados (Gil, 1987, p. 50).

## **Métodos e Procedimentos**

O desenvolvimento do sistema de agendamento de consultas seguiu uma abordagem estruturada, dividida em etapas técnicas que garantiram a organização e o cumprimento dos objectivos definidos. Segundo Sommerville (2019), a aplicação de métodos bem definidos na engenharia de software é essencial para assegurar a qualidade e previsibilidade dos resultados.  
 Inicialmente, foi realizado o levantamento de requisitos junto aos utilizadores, por meio de entrevistas e observações directas no ambiente hospitalares. Esta etapa permitiu identificar as principais funcionalidades necessárias, como o registro de pacientes, agendamento de consultas e notificações automáticas.

Em seguida, procedeu-se à análise e modelagem do sistema, utilizando a Linguagem de Modelagem Unificada (UML) para representar graficamente os processos e interacções. Foram elaborados diagramas de casos de uso, sequência, actividade, classe e objectos, permitindo compreender o comportamento dinâmico do sistema.

O sistema foi desenvolvido de forma incremental, com entregas parciais e revisões contínuas, seguindo os princípios de Scrum, uma metodologia ágil que enfatiza a colaboração e o desenvolvimento iterativo (Schwaber & Sutherland, 2020).

De acordo com Pressman e Maxim (2016), seguir etapas técnicas bem definidas, desde o processo de levantamento de requisitos até à validação, assegura a qualidade do ciclo de vida do software.

## **Ferramentas utilizadas e Tecnologias**

Para o desenvolvimento do nosso projecto usamos as seguintes linguagens de programação e ferramentas:

As ferramentas e tecnologias seleccionadas foram escolhidas de acordo com a sua robustez, compatibilidade e facilidade de uso, visando optimizar o processo de desenvolvimento do sistema.

* + 1. **Linguagens e Tecnologias**
* **JavaScript:** linguagem de programação utilizada tanto no backend como no frontend.
* **PHP:** linguagem de script do lado do servidor, utilizada para o desenvolvimento de funcionalidades robustas e dinâmicas.
* **SGBD MySQL:** sistema de gestão de base de dados leve e eficiente, uma base de dados relacional que oferece desempenho, estabilidade e suporte a linguagens SQL padronizadas.
* **React.js:** biblioteca JavaScript para a construção de interfaces de utilizador web modernas e responsivas.
  + 1. **Ferramentas de Desenvolvimento**
* **Visual Studio Code:** editor de código-fonte utilizado na escrita e organização do projecto.
* **Git e GitHub:** controle de versões e repositório remoto para trabalho colaborativo.
* **Canva:** ferramenta de design e prototipagem de interfaces.
* **Astah:** utilizada para a elaboração dos diagramas UML (casos de uso, sequência e actividade).
* **XAMPP:** ferramenta que integra servidor web (Apache), base de dados (MySQL/MariaDB) e suporte a PHP, permitindo configurar um ambiente local para o desenvolvimento e testes de aplicações web de forma rápida e simples.

Conforme destaca Pressman e Maxim (2016), a selecção adequada de ferramentas e tecnologias é determinante para o sucesso do desenvolvimento, garantindo eficiência, produtividade e facilidade de manutenção do sistema ao longo prazo.

## **Metodologia de Desenvolvimento de Software**

A metodologia que utilizamos para o desenvolvimento do sistema é a Metodologia Ágil, propriamente o Scrum. O Scrum foi criado por Jeff Sutherland, Ken Schwaber e Mike Beeble, e foi escolhido por ser o que melhor se identificou ao desenvolvimento do sistema. Pois permite um processo iterativo e incremental, garantindo entregas rápidas, participação activa dos utilizadores do Hospital Geral de Luanda, flexibilidade para lidar com mudanças nos requisitos e foco na qualidade e segurança essenciais em sistemas.

No Scrum, os projectos são divididos em ciclos chamados de Sprints, que ocorrerem entre uma semana e um mês, precedido por uma reunião de planeamento, onde tarefas são identificadas e um compromisso para o objectivo do Sprint é definido, seguido por uma reunião de revisão ou de retrospectiva, onde o progresso é revisto e lições para os próximos sprints são identificadas. Todas as sprints são postas em Product Backlog,

## **Levantamento e Analise de Requisitos**

O levantamento e a análise de requisitos são etapas essenciais no desenvolvimento de sistemas, pois determinam o que o sistema deve fazer e quais características deve possuir. O levantamento consiste na colecta de informações por meio de entrevistas, observações, questionários e análise de documentos. Já a análise organiza, verifica e valida essas informações, garantindo que os requisitos sejam claros, completos e viáveis. A correcta execução dessas fases reduz erros, evita retrabalhos e contribui para a criação de sistemas mais eficientes e alinhados aos objectivos do negócio.

## **Requisitos Funcionais do Sistema**

São actividades ou funcionalidades que o sistema deve realizar.

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **Identificação** | | Descrição |
| RF01 | O sistema deve permitir o controlo de acesso através de login; |
| RF02 | O sistema deve permitir o cadastro de pacientes; |
| RF03 | O sistema deve permitir o cadastro de médicos; |
| RF04 | O sistema deve permitir o agendamento de consultas; |
| RF05 | O sistema deve permitir a edição e o cancelamento de consultas; |
| RF06 | O sistema web deve listar as consultas e os pacientes; |
| RF07 | O sistema deve permitir a visualização dos horários das consultas; |
| RF08 | O sistema deve notificar o médico quando existir um novo agendamento; |
| RF09 | O sistema deve notificar o paciente quando estiver perto a data da consulta; |
| RF10 | O sistema deve permitir ao usuário o acesso ao histórico do médico; |
| RF11 | O sistema deve gerar relatório; |

Tabela 1 - Requisitos Funcionais do Sistema

Fonte: Elaboração própria, 2025

## **Requisitos Não Funcionais do Sistema**

São as funcionalidades que o programador define em relação as necessidades ou em função do problema que foi apresentado.

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Identificação** | | Nome | Descrição |
| RNF01 | **Segurança** | * O sistema deverá possuir um mecanismo de segurança por login; * O sistema deve permitir que cada conta de utilizador tenha um nível de acesso; * Só os usuários autorizados deveram acessar o sistema. |
| RNF02 | **Usuabilidade** | * O sistema possuirá uma interface que facilitará o usuário a se familiarizar com o sistema sem precisar de formação. |
| RNF03 | **Desempenho** | |  | | --- | | * O sistema deve processar as páginas   em menos de 5 segundos. O sistema deve suportar o acesso de mais de 100 usúarios | |
| RNF04 | **Disponibilidade** | * O sistema deve estar disponível sempre que o usuário tiver acesso a internet. * O sistema deve ser acessivel 24h/7 |

Tabela 2 - Requisitos Não Funcionais do Sistema

Fonte: Elaboração própria, 2025

## **Regras de Negócio**

|  |  |
| --- | --- |
| **Identificação** | **Descrição** |
| **RN01** | Só é permitido atender pessoas com consultas agendadas previamente no sistema. |
| **RN02** | O sistema deve permitir que o administrador tenha acesso a todas as telas e funcionalidades do sistema. |
| **RN03** | Cada conta de utilizador deve ter um nível acesso específico, garantido que apenas usuários autorizados acedam determinadas funcionalidades. |
| **RN04** | O sistema deve controlar a disponibilidade de horários para evitar conflitos de agendamento. |
| **RN05** | O sistema deve notificar ao paciente caso a consulta agendada for cancelada. |

Tabela 3 - Regras de Negócio

Fonte: Elaboração própria, 2025

## **Modelação do Sistema com UML**

Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema. A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica, que, actualmente, quase sempre é baseada em notações de UML (linguagem de modelação unificada, do inglês Unified Modelling Language).

A modelagem de sistema é o processo de elaboração de modelos abstratos de um sistema, normalmente representado por meio de um diagrama, em que cada um desses modelos apresenta uma visão ou perspectiva diferente do sistema. Esses modelos, normalmente, são elaborados utilizando-se uma notação gráfica, que, em nosso caso, será a UML (Sommerville, 2011, p. 82).

Conforme Dos Santos (2009), UML é a sigla de Unified Modelling Language que em português é Linguagem de Modelação Unificada é uma linguagem padrão para elaboração da estrutura de projectos de software. A UML poderá ser usada para: visualização, especificação, construção de modelos e diagramas e documentação.

O mesmo autor, afirma que, a UML é apenas uma linguagem e, portanto, é uma parte de um método para desenvolvimento de software. Ela é independente do processo, apesar de ser perfeitamente utilizada em processo orientado a casos de usos, centrado na arquitectura iterativo e incremental.

A UML é o resultado da unificação da linguagem de modelagem de objectos de três métodos líderes do mercado: Booch, Object Modelling Technique (OMT) e Objected-Oriented Software Engineering (OOSE). Em 1997, a UML v1.1 foi adoptada pela OMG (Object Management Group) e desde então tornou-se o padrão da indústria de software para a modelagem de objectos e componentes.

A UML tem dois tipos de diagramas principais que são:

**Os** **diagramas estáticos** que modela a estrutura do sistema e é composto por classes, objectos, estruturas de dados e os relacionamentos existentes entre eles.

**Os diagramas dinâmicos** que modelam o comportamento do sistema e é composto pelos seguintes elementos: diagrama de casos de uso, diagrama de classes, Diagrama de objectos, diagrama de sequência e diagrama de actividades.

## **Diagrama de casos de uso**

O Diagrama de Caso de Uso é uma ferramenta fundamental da Unified Modeling Language (UML) utilizada para representar graficamente as funcionalidades de um sistema do ponto de vista do usuário. Ele descreve as interações entre os atores (usuários ou outros sistemas) e os casos de uso (funções ou serviços oferecidos pelo sistema), proporcionando uma visão geral do comportamento esperado antes da implementação (Booch, Rumbaugh & Jacobson, 2005).

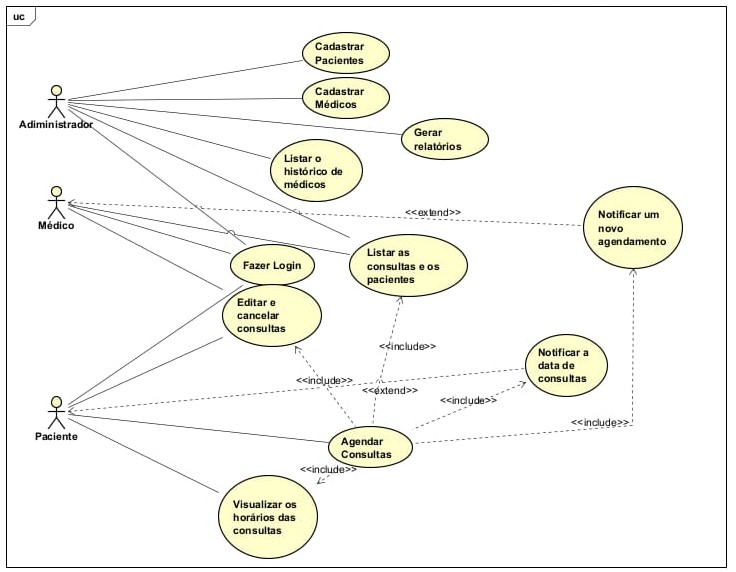


Figura 3 - Diagrama de Casos de Uso

Fonte: Elaboração própria, 2025

## **Descrição dos principais casos de uso do sistema**

Neste ponto são apresentados algumas descrições de casos de uso do nosso sistema.

* **Descrição do caso de uso Cadastrar Paciente**

|  |  |
| --- | --- |
| UC | Cadastrar Paciente |
| Actor: | * Paciente - Anonimo |
| Pré-Condição | * Estar logado. |
| Pós-Condição | * O paciente deve estar cadastrado no sistema. |
| Fluxo Principal | 1. Aceder a página de cadastro de paciente; 2. Inserir os dados; 3. Submeter os dados; 4. Validar os dados; 5. Guardar dados. |
| Fluxo Alternativo | Erro no ponto 4: Dados inválidos; Informar o usuário sobre o(s) erro(s) encontrados (por exemplo: campos obrigatórios em branco, formato de contacto inválido, documento já existente) e retorna ao ponto 2 para correcção dos dados. |

Tabela 4 - Descrição de Caso de Uso: Cadastrar Paciente

Fonte: Elaboração própria, 2025

* **Descrição do caso de uso agendar consulta**

|  |  |
| --- | --- |
| UC | Agendar Consulta |
| Actor: | * Paciente ( usuário cadastrado) |
| Pré-Condição | * O paciente deve estar logado no sistema e possuir cadastro activo. |
| Pós-Condição | * A consulta deve estar registrada no sistema com data, hora e médico seleccionado. |
| Fluxo Principal | 1. Aceder à página de Agendamento de consultas;  2. Seleccionar o tipo de consulta e o profissional desejado;  3. Escolher a data e o horário disponíveis;  4. Confirmar os dados do agendamento;  5. Submeter o pedido de agendamento;  6. O sistema valida as informações e grava os dados da consulta. |
| Fluxo Alternativo | Erro no ponto 4: Dados incorrectos ou o horário já esteja ocupado, o sistema informa o usuário sobre o erro e retorna ao ponto 3 para nova selecção. |

Tabela 5 - Descrição de Caso de Uso: Agendar Consultas

Fonte: Elaboração própria, 2025

* **Descrição do caso de uso notificar médico**

|  |  |
| --- | --- |
| UC | Notificar Médico |
| Actores: | * Sistema e o médico |
| Pré-Condição | * Deve existir uma consulta agendada e registrada no sistema. |
| Pós-Condição | * O médico é notificado sobre o novo agendamento. |
| Fluxo Principal | 1. Sistema verifica se existe um novo agendamento; 2. Sistema obtém as informações da consulta(hora, data, tipo de consulta, médico e dados do paciente); 3. Sistema identifica o médico associado a consulta; 4. Sistema envia uma notificação ao médico sobre a consulta; 5. Médico recebe a notificação sobre consulta; |
| Fluxo Alternativo | Erro ponto 4: caso a notificação não chegue ao médico, o sistema volta para o ponto 3. |

Tabela 6 - Descrição de Caso de Uso: Notificar Médico

Fonte: Elaboração própria, 2025

## **Diagramas de Actividade**

É um diagrama que descreve o fluxo sequencial das actividades ou processos dentro de um sistema, sendo útil na representação de algoritmos e fluxos de trabalho. Diferencia-se do fluxograma tradicional por poder representar atividades simultâneas, tornando-se essencial para modelar sistemas complexos ou distribuídos (Blaha & Rumbaugh, 2006; Silva & Videira, 2001).

A figura 4 é referente a um diagrama onde representamos o fluxo de actividades necessários para a realização do caso de uso cadastrar paciente.

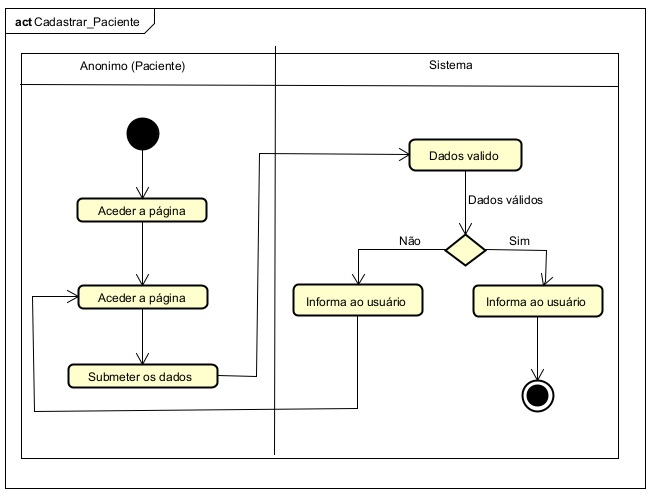


Figura 4 - Diagrama de Actividade: Cadastrar Paciente

Fonte: Elaboração própria, 2025

A figura 5 é referente a um diagrama onde representamos o fluxo de actividades necessários para a realização do caso de uso agendar consulta.

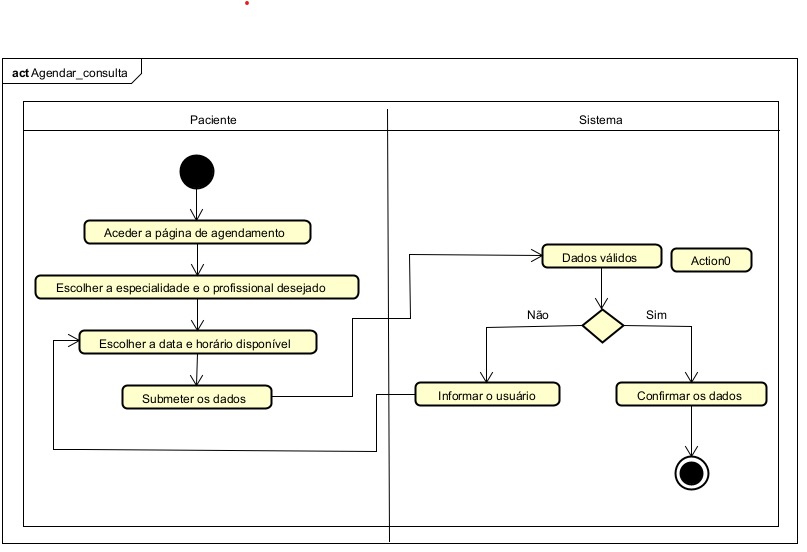


Figura 5 - Diagrama de Actividades: Agendar consulta

Fonte: Elaboração própria, 2025

A figura 6 é referente a um diagrama aonde representamos o fluxo de actividades necessários para a realização do caso de uso notificar o médico na existência de um novo agendamento.



Figura 6 - Diagrama de Actividades: Notificar Médico

Fonte: Elaboração própria, 2025

## **Diagramas de Sequência**

É um diagrama que ilustra as interações entre objectos e atores ao longo do tempo, representando a troca de mensagens entre eles por meio de linhas de vida. É ideal para detalhar cenários específicos de casos de uso, incluindo fluxos alternativos ou exceções, como a rejeição de uma operação por falta de fundos (Miles & Hamilton, 2005).

Na figura 7 é representado a sequência de processos para que se realize o caso de uso cadastrar paciente.

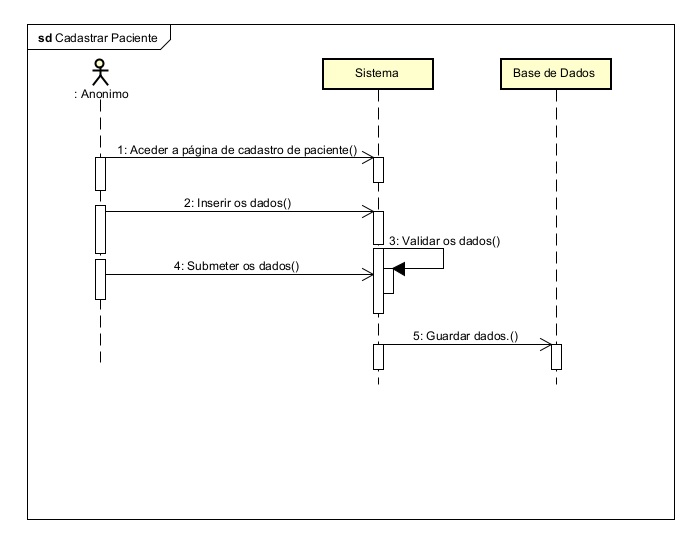


Figura 7 - Diagrama de sequência: Cadastrar paciente

Fonte: Elaboração própria, 2025

Na figura 8 é representado a sequência de processos para que se realize o caso de uso Agendar consulta.

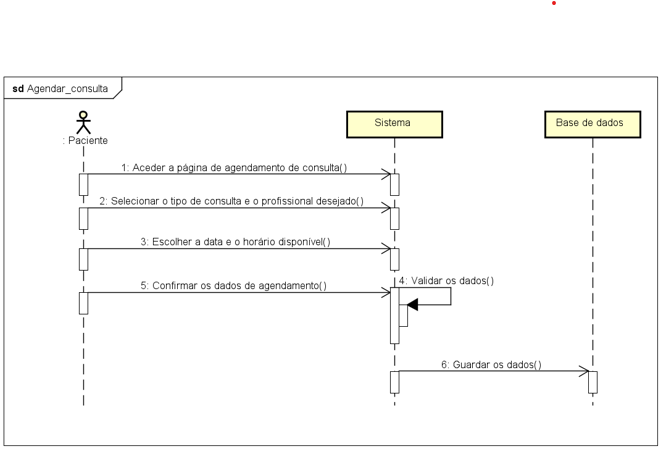


Figura 8 - Diagrama de sequência: Agendar consulta

Fonte: Elaboração própria, 2025

Na figura 9 é representado a sequência de processos para que se realize o caso de uso notificar o médico na existência de um novo agendamento.

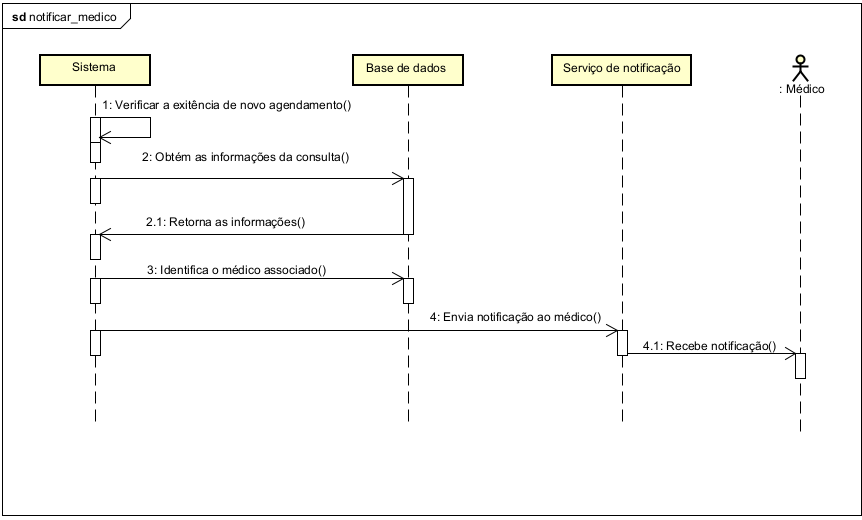


Figura 9 - Diagrama de sequência: Notificar Médico

Fonte: Elaboração própria, 2025

## **Diagramas de Classe**

O Diagrama de Classes fornece uma visão estática da estrutura do sistema, mostrando classes, atributos, métodos e relacionamentos. Ele é essencial na análise e design orientado a objectos, facilitando a comunicação entre analistas e programadores, além de servir como base para a geração de código e assegurar a consistência entre o modelo conceitual e o sistema implementado (Rumbaugh et al., 1991; Blaha & Rumbaugh, 2005).

Na figura XXXXXXXXXX é representado o diagrama de classe onde podemos mostrar se uma classe herda de outra ou se contém uma referência para outra e também vimos a relação existente entre as tabelas.

## **Diagramas de Objectos**

Diagrama de Objectos representa instâncias específicas das classes, ilustrando exemplos concretos de como os objectos se relaccionam em tempo de execução. Ele complementa o diagrama de classes, oferecendo uma visão prática e detalhada da estrutura e das relações do sistema (Miles & Hamilton, 2006).

## **Protótipos e Interface do Sistema**

## **Cronograma de Desenvolvimento**

CAPÍTULO III – APRESENTAÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS OBTIDOS

3.1. Introdução

O presente capítulo tem como propósito apresentar os resultados obtidos no desenvolvimento do Sistema de Agendamento de Consultas para o HGL, no qual são descritas as principais funcionalidades implementadas, bem como os objectivos alcançados com a sua aplicação.

Neste capítulo serão apresentados o protótipo do sistema, por meio de capturas de telas, além da análise e discussão dos resultados obtidos durante a fase de desenvolvimento e utilização do sistema, evidenciando o seu funcionamento e a sua aplicabilidade no contexto hospitalar.

A avaliação do sistema será realizada com base na comparação entre os objectivos definidos no Capítulo I e os resultados efectivamente alcançados, considerando o cumprimento dos requisitos funcionais, o desempenho do sistema, a usabilidade das interfaces e os benefícios proporcionados em relação ao processo tradicional de agendamento de consultas.

3.2. Visão Geral do Sistema Desenvolvido

O sistema desenvolvido denomina-se SAC-HGL: Appointment System, uma plataforma de agendamento remoto de consultas, cujo objectivo principal é facilitar o processo de marcação de consultas no HGL, proporcionando maior comodidade, eficiência e organização tanto para os pacientes quanto para a administração hospitalar.

O sistema funciona de forma semelhante ao modelo tradicional de agendamento de consultas, porém realizado gestão integralmente de forma remota, permitindo que os usuários realizem a marcação sem a necessidade de deslocar-se ao hospital. Dessa forma, contribui para a redução de longas filas de espera, optimiza o tempo dos pacientes e melhora o acesso às informações relacionadas às consultas.

O público-alvo do sistema é constituído por pacientes do HGL, bem como por administradores e profissionais responsáveis pela das consultas, que passam a dispor de uma ferramenta informatizada para o controlo e organização dos atendimentos.

O SAC-HGL: Appointment System é composto pelos seguintes módulos principais:

1. Módulo de Autenticação: Responsável pelo processo de agendamento de consulta e acesso a partes restritas da plataforma web, permitindo que os usuários cadastrados e posteriormente logados possam fazer o agendamento da consulta de forma segura.
2. Módulo de Gestão de Usuários: Destinado à administração do sistema, possibilitando ao administrador o controlo dos usuários, bem como a definição e acompanhamento das suas respectivas funções.
3. Módulo de Agendamento de Consultas: Permite a marcação de consultas de forma prática e eficiente, possibilitando aos pacientes selecionar datas e horários disponíveis, de acordo com as especialidades oferecidas.

O sistema foi desenvolvido para execução em ambiente web, podendo ser acessado por meio de navegadores, o que garante maior flexibilidade e acessibilidade aos seus utilizadores.

3.3. Arquitetura do Sistema

O sistema SAC-HGL: Appointment System foi desenvolvido com base na arquitetura em camadas, também conhecida como arquitetura de três camadas (3-tier), a qual permite uma melhor organização, manutenção e escalabilidade do sistema. O diagrama abaixo representa a estrutura do sistema.

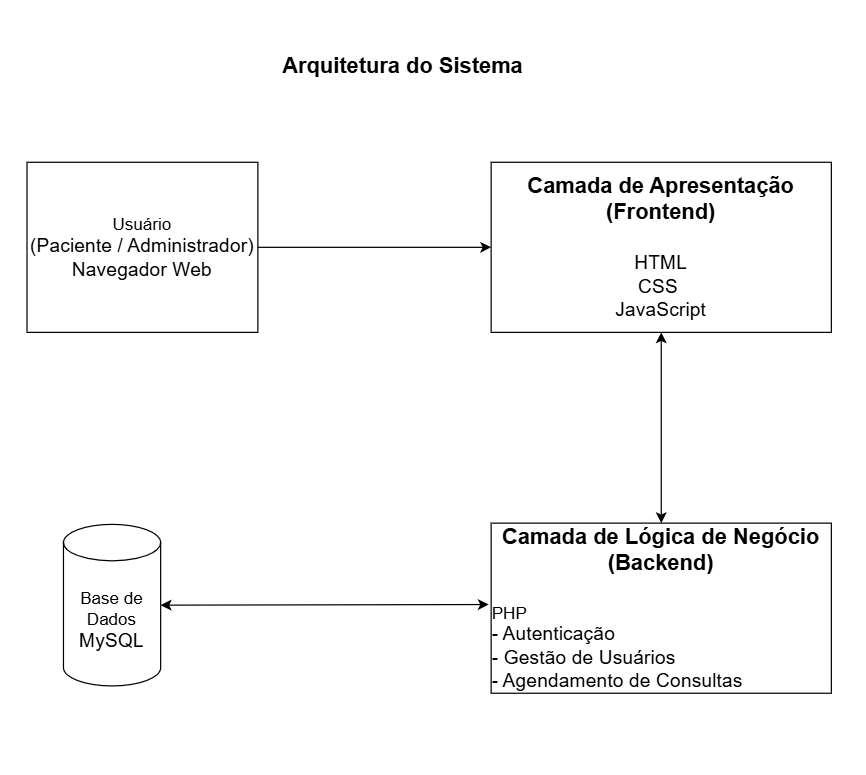


Figura1 – Arquitetura do sistema  
Fonte: Elaboração Própria, 2025

A Camada de Apresentação é responsável pela interação com o utilizador, sendo implementada com as tecnologias HTML, CSS e JavaScript, permitindo o acesso ao sistema por meio de um navegador web.

A Camada de Lógica de Negócio, desenvolvida em PHP, é responsável pelo processamento das requisições, aplicação das regras de negócio e controlo das funcionalidades principais do sistema, tais como autenticação, gestão de usuários e agendamento de consultas.

A Camada de Dados utiliza o sistema de gerenciamento de base de dados MySQL, sendo responsável pelo armazenamento e recuperação das informações do sistema.

A comunicação entre as camadas ocorre de forma sequencial, onde as requisições realizadas pelos utilizadores são processadas pelo backend e os resultados são apresentados novamente na interface do sistema.

3.4. Funcionalidades Implementadas

Esta secção apresenta as principais funcionalidades implementadas no sistema, organizadas de acordo com os módulos previamente definidos, demonstrando o seu funcionamento e a forma como atendem aos requisitos estabelecidos para o desenvolvimento do sistema.

* Módulo de Autenticação

O módulo de autenticação tem como principal função regular o acesso dos utilizadores ao sistema, permitindo a sua identificação e validação com base nas informações armazenadas na base de dados.

Inicialmente, o utilizador anônimo pode aceder a plataforma normalmente sem estar cadastrado e logado no sistema, mas quando ele quizer agendar uma consulta, deverá se cadastrar estar logado para fazer o agendamento da consulta. Para o cadastro do paciente, o sistema oferecerá duas opções, naqual o mesmo pode se deslocar para ser cadastrado pelo adminstrador ou a partir de casa, ele poderá fazer o cadastro.

A Figura 2 ilustra a interface do módulo de autenticação do sistema.

[Screenshot Picture]

*Figura 2 – Módulo de Autenticação*

* Módulo de Gestão de Utilizadores

O módulo de gestão de utilizadores é destinado exclusivamente ao administrador do sistema, permitindo-lhe exercer controlo sobre os utilizadores registados, bem como gerir os seus respectivos papéis e permissões.

Após efectuar o login, o administrador tem acesso às principais funcionalidades do sistema, incluindo a área de gestão de utilizadores, onde pode visualizar, adicionar, editar ou remover utilizadores de forma prática e intuitiva, graças à interface gráfica desenvolvida.

A Figura 3 apresenta a interface correspondente ao módulo de gestão de utilizadores.

[Screenshot Picture]

*Figura 3 – Módulo de Gestão de Utilizadores*

* Módulo de Agendamento de Consultas

O módulo de agendamento de consultas constitui o principal componente do sistema, sendo responsável pela marcação e gestão das consultas médicas de forma remota.

O usuário logado, tem acesso à funcionalidade de agendamento de consultas, onde pode selecionar o tipo de consulta pretendida, bem como o horário disponível apresentado pelo sistema. Ao confirmar o agendamento, os dados da consulta são registados na base de dados, garantindo que o horário selecionado fique indisponível para outros pacientes.

A Figura 4 ilustra a interface do módulo de agendamento de consultas.

[Screenshot Picture]

*Figura 4 – Módulo de Agendamento de Consultas*

Com base nos testes e análises realizados, verifica-se que o sistema contribui significativamente para a resolução de um dos principais problemas do agendamento tradicional, nomeadamente as longas filas de espera. No contexto do Hospital Geral de Luanda, é comum que os utentes permaneçam durante várias horas à espera de atendimento, o que gera insatisfação e desconforto. Com a implementação do sistema, os pacientes podem realizar a marcação das consultas em poucos minutos e dirigir-se ao hospital apenas no horário agendado, reduzindo significativamente o tempo de espera e melhorando a qualidade do atendimento.

3.5. Interface do Sistema

# **REFERÊNCIAS BIBLIOGRÁFICAS**

Ala, A., & Chen, F. (2022). *Appointment Scheduling Problem in Complexity Systems of the Healthcare Services: A Comprehensive Review.* Obtido de PubMed Central: https://pmc.ncbi.nlm.nih.gov/articles/PMC8913063

Alam, M., Rahman, A., & Hossain, M. (2021). *A study on Laravel framework and its architeture.* Obtido de ResearchGate: https://www.researchgate.net/publication/351776747\_A\_Study\_on\_Laravel\_Framework\_and\_Its\_Architecture

Alura. (2024). *Banco de Dados: o que é, principais tipos e um guia para iniciar*. Obtido de Alura: https://www.alura.com.br/artigos/banco-de-dados

Anderson, K. (16 de Outubro de 2025). *What Is Cross-Platform Software Development?* Obtido de Cognativ: https://www.cognativ.com/blogs/post/what-is-cross-platform-software-development/168

Battisti, M. (12 de Feveireiro de 2021). *Introdução ao React - Aula 1 [Vídeo]*. Obtido de YouTube: https://youtu.be/FXqX7oof0I4?si=UtMaSbS1wyQHU8x

Biblioteca Trabalhista. (2024). *O que é marcação de consultas.* Obtido de Biblioteca Trabalhista: https://bibliotecatrabalhista.com.br/glossario/o-que-e-marcacao-de-consulta-2

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2nd ed.). Addison-Wesley.

Boyatzis, R., Calvetti, D., & Meckes, E. (17 de Outubro de 2023). *What is Accessibility?* Obtido de https://case.edu/accessibility/what-accessibility

Caldeira, R. P. (2023). *Banco de dados, database, SGBD: você sabe o que é isso?* Obtido de Idicium Academy: https://academy.indicium.tech/blog/banco-de-dados-database-sgbd-voce-sabe-o-que-e-isso

Damas, L. (2005). *SQL* (7ª ed.). Lisboa: Editora de Informática, Lda FCA e Lidel e Edições Técnicas, Lda.

Dos Santos, F. R. (2009). *UML - Linguagem de Modelagem Unificada.*

Duckett, J. (2011). *HTML & CSS: Design and build websites.* Wiley.

Eich, B. (1995). *JavaScript definitive guide.* Netscape Communications.

Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Boston, Massachusetts: Pearson Education.

Estrella, C. (7 de Maio de 2025). *Como conectar PHP a um banco de dados MySQL*. Obtido de Hostinger: https://www.hostinger.com/br/tutoriais/como-conectar-php-com-mysql

Ferreira, A. (2007). *Sistema de Informação: A Base do Conhecimento.*

Frankl, V. E. (1945). *Man’s Search for Meaning.* Viena: Franz Deuticke Verlag.

Fullstack.io. (2020). *30 days of React: Learn React in 30 days by building 30 projects.* Obtido de Fullstack.io: https://github.com/fullstackreact/30-days-of-react

Grigoryan, A. (2024). *O que são sistemas de informações*. Obtido de Latenode: https://latenode.com/pt-br/blog/what-is-information-systems

Higher School of Network. (13 de Abril de 2023). *Database Management Systems – DBMS: a guide with the most relevant information about this technology.* Obtido de Escola Superior de Redes: https://esr.rnp.br/desenvolvimento-de-sistemas/sistemas-gerenciadores-de-banco-de-dados

Juliano. (2014). *Gerenciamento de Banco de Dados: Análise Comparativa de SGBD’S*. Obtido de DevMedia: https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd-s/30788

Júnior, M. F. (Outubro de 2022). *Sistema de Marcação de Consultas.* Obtido de Monografia: https://monografias.ufop.br/bitstream/35400000/5300/6/MONOGRAFIA\_SistemaMarcaçãoAgendamento.pdf

Kruchten, P. (2000). *The Rational Unified Process* (2ª ed.). Addison Wesley.

Larman, C. (2012). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (3rd ed.). Prentice Hall.

Lerdorf, R. (2016). *Introdução à programação web com PHP.*

Lima, V. (2019). *Node.js - Curso e explicações práticas sobre o ambiente JavaScript.*

McDowell, D. L., & Panchal, J. H. (2010). *Integrated Design of Multiscale Multifunctional Material and Products.* Butterworth-Heinemann.

Mills , C., & Vadim, B.-b. (2025). *HTML, CSS and JavaScript documantation.* Mozilla Foundation.

Obermeier, A. (8 de Outubro de 2025). *8 Database Management System Selection Guide: 8 Critical Factors for Business Success*. Obtido de Paessler: The Monitoring Experts: https://blog.paessler.com/key-considerations-when-choosing-a-dbms

O'Brien, J. A. (2000). *Introduction to Information Systems: Essentials for the Internetworked Enterprise* (10. ed.). New York: McGraw-Hill/Irwin.

Oliveira, J. F., & Martins, L. C. (2019). A informatização dos serviços de saúde: Desafios e benefícios dos sistemas de agendamento eletrônico. *Cadernos de Administração Pública, 25(2)*, pp. 77-90.

Pandey, D., Aron , M., & Ramachandran, A. (2023). *O que é um banco de dados?* Obtido de Glossário Nutanix: https://www.nutanix.com/pt\_br/info/database

Pearson Education. (2016). *Concepts of programing languages* (12ª ed.). Pearson.

Pereira, B. (19 de Janeiro de 2023). Obtido de Brainly: https://brainly.com.br/tarefa/54842334

Pereira, J. L. (1998). *Tecnologia de Bases de Dados* (3ª ed.). Lisboa: FCA.

Pressman, R. S. (2016). *Software Engineering: A Practitioner’s Approach* (8th ed.). McGraw-Hill Education.

Rattighieri, B. C., & Cracco, V. A. (2023). *Protótipo de Sistema de Agendamento para Consultório Médico.* Obtido de https://www.fateclins.edu.br/web/arquivos/tg/egressos/vROZ3xeYwMFpTPguNpAU9XfdcZRGNxr6.pdf#:~:text=3.1.3%20Diagrama%20de%20Caso%20de%20Uso%20Consultas,de%20ele%20desmarcar%20a%20consulta%20já%20agendada.&text=Fonte:%20Elaborada%20pelos%20autores%2C%202023

Ravanello, F. (2025). *Sistemas de agendamento de consultas*. Obtido de Gestão DS: https://www.gestaods.com.br/sistema-de-marcacao-de-consulta

React. (2023). *React Documentation.* Obtido de https://react.dev/learn

Severino, A. J. (2007). *Metodologia do trabalho científico.* São Paulo: Cortez.

Shah, D. (8 de Julho de 2025). *Advantages and Disadvantages of Developing Web Applications*. Obtido de Solvios Technology: https://solvios.technology/advantages-and-disadvantages-of-web-applications/

Sommerville, I. (2011). *Engenharia de Software* (9ª ed.). São Paulo: Pearson Prentice Hall.

Sommerville, I. (2019). *Engenharia de Software* (10ª ed.). Pearson.

Souza, R. P., Lima, A. M., & Ferreira, T. S. (2020). Tecnologia da informação na gestão da saúde: O impacto dos sistemas de agendamento online. *Revista Brasileira de Informática em Saúde, 16(1)*, pp. 45–52.

Stauffer, M. (2019). *Laravel: Up & running: A framework for building modern PHP apps* (2ª ed.). O'REILLY Media.

Susnjara, S., & Smalley, I. (17 de Outubro de 2025). *What is cloud storage?* Obtido de IBM: https://www.ibm.com/think/topics/cloud-storage

The PHP Group. (2025). *O que é o PHP e o que ele pode fazer?* Obtido de Manual do PHP: https://www.php.net/manual/pt\_BR/introduction.php

W3C. (2023). *HTML and CSS Overview.* Obtido de https://www.w3.org/standards/webdesign/htmlcss

Workana. (2023). *O que é PHP?* Obtido de Workana: https://i.workana.com/guias/o-que-e-php

Zac, R. (2024). *O que é agendamento de consultas*. Obtido de Glossário Renata Zac: https://renatazac.com.br/glossario/o-que-e-agendamento-de-consultas.

**GLOSSÁRIO**

API (Application Programming Interface) - Interface de Programação de Aplicações; conjunto de definições e protocolos que permite a comunicação entre diferentes sistemas de software.

AWS (Amazon Web Services) - Plataforma de serviços de computação em nuvem oferecida pela Amazon.

Backup - Cópia de segurança de dados armazenados, realizada para prevenir perda de informações em caso de falhas ou incidentes.

Base de Dados (BD) - Conjunto organizado de dados relacionados, armazenados electronicamente e geridos por um sistema específico.

CSS (Cascading Style Sheets) - Linguagem de folhas de estilo utilizada para definir a apresentação visual de documentos HTML.

Framework - Estrutura de suporte que oferece componentes e bibliotecas pré-definidas para facilitar o desenvolvimento de software.

GPS (Global Positioning System) - Sistema de Posicionamento Global; tecnologia de navegação por satélite.

HTML (HyperText Markup Language) - Linguagem de Marcação de Hipertexto; linguagem padrão para criação de páginas web.

HTTP/HTTPS - Protocolo de Transferência de Hipertexto (Seguro); protocolo de comunicação utilizado para transferência de dados na web.

Inteligência Artificial (IA) - Área da ciência da computação dedicada ao desenvolvimento de sistemas capazes de realizar tarefas que normalmente requerem inteligência humana.

JIT (Just-In-Time) - Técnica de compilação que melhora o desempenho de programas durante a execução.

JavaScript (JS) - Linguagem de programação utilizada principalmente para adicionar interactividade a páginas web.

Laravel - Framework PHP moderno baseado na arquitectura MVC, utilizado para desenvolvimento web.

MVC (Model-View-Controller) - Padrão arquitectural que separa a aplicação em três componentes: modelo (dados), visão (interface) e controlador (lógica).

MySQL - Sistema de gestão de base de dados relacional open source amplamente utilizado. NoSQL (Not Only SQL) - Tipo de base de dados não relacional, adequada para grandes volumes de dados não estruturados.

OMG (Object Management Group) - Grupo de Gestão de Objectos; consórcio internacional que define padrões para tecnologia orientada a objectos.

PDO (PHP Data Objects) - Extensão PHP que fornece uma interface consistente para acesso a bases de dados.

PHP (Hypertext Preprocessor) - Linguagem de script do lado do servidor, amplamente utilizada para desenvolvimento web.

RAM (Random Access Memory) - Memória de Acesso Aleatório; tipo de memória volátil usada para armazenamento temporário de dados.

React.js - Biblioteca JavaScript para construção de interfaces de utilizador interactivas e componentes reutilizáveis.

RDBMS (Relational Database Management System) - Sistema de Gestão de Base de Dados Relacional.

Requisitos Funcionais - Especificações que descrevem as funcionalidades e comportamentos que um sistema deve executar.

Requisitos Não Funcionais - Especificações que definem atributos de qualidade do sistema, como desempenho, segurança e usabilidade.

Scrum - Framework ágil para gestão de projectos, baseado em ciclos iterativos chamados sprints.

SGBD (Sistema de Gerenciamento de Base de Dados) - Software responsável por criar, gerir e manipular bases de dados.

Sistema de Informação (SI) - Conjunto integrado de componentes (pessoas, hardware, software, dados e processos) para colectar, processar e distribuir informações.

SMS (Short Message Service) - Serviço de Mensagens Curtas; sistema de envio de mensagens de texto.

SPA (Single Page Application) - Aplicação de Página Única; aplicação web que carrega uma única página HTML e actualiza dinamicamente o conteúdo.

SQL (Structured Query Language) - Linguagem de Consulta Estruturada; linguagem padrão para manipulação de bases de dados relacionais.

UML (Unified Modelling Language) - Linguagem de Modelagem Unificada; notação padrão para modelagem de sistemas orientados a objectos.

XAMPP - Pacote de software que integra servidor Apache, MySQL/MariaDB, PHP e Perl para criar ambiente de desenvolvimento local.