****

**INSTITUTO POLITÉCNICO INDUSTRIAL DE LUANDA**

**CURSO TÉCNICO DE GESTÃO DE SISTEMAS INFORMÁTICOS**

**CRIAÇÃO DE UM SISTEMA WEB DE MARCAÇÃO DE CONSULTAS PARA O HOSPITAL JOSINA MACHEL - SISTMARC**

Autores:

Paulo Barros Franco

Rossana Patrícia Pereira Pinto

Turma: 13ª GSI

Orientador: Arnaldo Bonga

**LUANDA**

**2023**

**CRIAÇÃO DE UM SISTEMA WEB DE MARCAÇÃO DE CONSULTAS PARA O HOSPITAL JOSINA MACHEL - SISTMARC**

Trabalho de fim do curso apresentado à coordenação do Instituto Politécnico Industrial de Luanda, para a obtenção do título de Técnico Médio no curso de Gestão de Sistemas Informático, realizado sob a orientação do Prof. Eng.º Arnaldo Bonga.

Prof.Engº. Arnaldo Bonga

(Prof. Orientador)

**LUANDA**

**2023**

**CRIAÇÃO DE UM SISTEMA WEB DE MARCAÇÃO DE CONSULTAS PARA O HOSPITAL JOSINA MACHEL - SISTMARC**

Trabalho de fim do curso apresentado à coordenação do Instituto Politécnico Industrial de Luanda, para a obtenção do título de Técnico Médio no curso de Gestão de Sistemas Informático, realizado sob a orientação do Prof. Eng.º Arnaldo Bonga.

**Aprovado em\_\_\_\_de\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_de\_\_\_\_\_\_\_.**

**Nota:\_\_\_\_\_.**

**BANCA EXAMINADORA**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prof.Engº. Arnaldo Bonga**

**(Prof. Orientador)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(Oponente I)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(Oponente II)**

**LUANDA**

**2023**

Sumário

[**1. INTRODUÇÃO 5**](#_Toc126050535)

[**1.1 Hipóteses 5**](#_Toc126050536)

[**1.2 Objectivos 5**](#_Toc126050537)

[**1.2.1 Objectivo Geral 5**](#_Toc126050538)

[**1.2.2 Objectivos Específicos 5**](#_Toc126050539)

[**1.3 Justificativa 6**](#_Toc126050540)

[**1.4 Quadro Teórico 6**](#_Toc126050541)

[**CAPÍTULO 2: FUNDAMENTAÇÃO TEÓRICA 8**](#_Toc126050542)

[**2.1 Origem dos Sistemas de Informação 8**](#_Toc126050543)

[**2.1 Sistema 9**](#_Toc126050544)

[**2.2 Sistema Web 9**](#_Toc126050545)

[**2.3 Sistema Web para marcação de consultas 10**](#_Toc126050546)

[**2.4 Benefícios de um Sistema Web para marcação de consultas 10**](#_Toc126050547)

[**3 METODOLOGIA 11**](#_Toc126050548)

[**3.1 Metodologia de investigação científica 11**](#_Toc126050549)

[**3.2 Etapas da Pesquisa Erro! Indicador não definido.**](#_Toc126050550)

# 1. INTRODUÇÃO

Acessar aos serviços de saúde em Luanda tem apresentado certa dificuldade, desde o atendimento ao tratamento mais complexo. Quando nos dirigimos a uma unidade hospitalar, há sempre dificuldades em marcar uma consulta na área em que precisamos ser consultados e tem sido um processo muito demorado. Muitas vezes temos de sair muito cedo das nossas casas para marcar uma simples consulta, o que tem causado alguns transtornos devido a enchente, transportes e as nossas actividades laborais.

A situação de quem espera por atendimento médico, em alguns hospitais públicos, é**preocupante**. Os pacientes são submetidos a longas filas para fazerem o agendamento, esta situação em alguns casos resulta no agravamento da saúde ou mesmo a morte de quem aguarda pelo atendimento.

Essa situação é igual em certas clínicas privadas. Em Luanda verifica-se a mesma situação no Hospital Público Josina Machel que tem preocupado cada vez mais as autoridades sanitárias por ser uma das unidades mais aderidas pela população em Luanda.

Diante deste facto que permanece é evidente que se precisa pensar em: quais são os factores que dificultam o atendimento dos pacientes no hospital Josina Machel.

## **1.1 Hipóteses**

1. A falta de um sistema web de marcação de consulta;
2. Quantidades insuficientes de consultórios;

## **1.2 Objectivos**

### **1.2.1 Objectivo Geral**

O objectivo deste trabalho é desenvolver um software para auxiliar a marcação de consultas.

### **1.2.2 Objectivos Específicos**

* Criar uma base de dados para armazenar os dados e informações necessárias para o procedimento de marcação de consultas;
* Desenvolver páginas HTML visualmente interativas para os pacientes;
* Programar códigos php para manter a conectividade da base de dados com as páginas web;
* Desenvolver um painel administrativo para gerir e manter as marcações de consultas;

## **1.3 Justificativa**

O uso de novas tecnologias tem facilitado a realização de diversas tarefas por meio um único mecanismo, citando como exemplo o “Excel”, o uso desta tecnologia em empresas tem permitido que um único funcionário consiga verificar produtos disponíveis, o preçário por produto, as vendas realizdas e entre muitas outras tarefas possíveis, os benefícios da mesma tem ajudadoemoresas a pouparem mais em recurso humanos e financeiros.

De modo a melhorar a organização e a realizacão do trabalho muitas unidades hospitalares têm feito o uso de tecnologias auxiliares como a “TELEMEDICINA“, actualmente a marcação de consulta online tem sido uma boa opção, sendo beneciável tanto para as unidades como para os pacientes dando a autonomia de realizar o agendamento de consulta de um modo mais rápido seja aonde estiver.

Essas Tecnologias tem ganhado grande destaque nos últimos dias por terem uma interface interativa e comoda tem atraído as pessoas a utilizarem, deste modo o portal do governo facilita a marcação de consulta no hospitail David Bernardino (Hossi, 2023). Destaca-se, porém que o SISTMARC por apresentar esta automação de processos manuais, proporciona uma melhoria, como no caso de gerenciamento das agendas, horários, disponibilidade, assim como praticidade e comodidade para o paciente. A utilização do mesmo vai permitir que haja uma diminuição considerável das enchentes verificadas no Hospital Josina Machel de modo a melhorar a qualidade de agendamento e sua consistência.

## **1.4 Quadro Teórico**

A fundamentação teórica garante a confiabilidade e a qualidade técnica e ciéntica do trabalho. Em outras palavras, dá a certificação de que o trabalho possui um embasamento teórico e científico firme.

A área de saúde vem crescentemente aderindo à internet e fazendo uso da mesma em diversas maneiras, desde processos administrativos até a assistência ao paciente, como é o caso da telemedicina (COSTA, 2001, p. 30).

Com o aumento de novas tecnologiase e a medida que os protagonistas do mercado de saúde se conectam, ocorrerá gradativamente um aumento no número de usuários desses serviços, levando à redução de custos e maior eficiência do trabalho em saúde (DARVES, 2006, p.10).

Os estudos já existentes relacionados com o impacto da internet na saúde, apesar de não se focarem em falar de marcação de consultas online, permitiu de facto obter certo aprofundamento do problema encontrado nesta pesquisa.

## **1.5 Estrutura do trabalho**

Este trabalho é apresentado por 6 capítulos:

**Capítulo 1 -** Introdução

Este capítulo espelha o contexto

**Capítulo 2-** Fundamentação teórica;

**Capítulo 3 -** Metodologia;

**Capítulo 4 –** Implementação do projecto**;**

**Capítulo 5 –** Resultados**;**

**Capítulo 6 –** Considerações finais**.**

# CAPÍTULO 2: FUNDAMENTAÇÃO TEÓRICA

## **2.1 Origem dos Sistemas de Informação**

O Século XX é considerado aquele do advento da Era da Informação. A partir de então, a informação começou a fluir com velocidade maior que a dos corpos físicos. Desde a invenção do telégrafo elétrico em 1837, passando pelos meios de comunicação de massa, e até mais recentemente, o surgimento da grande rede de comunicação de dados que é a Internet, o ser humano tem de conviver e lidar com um crescimento exponencial do volume de dados disponíveis. Os sistemas de informação surgiram antes mesmo da informática. Antes da popularização dos computadores, os sistemas de informação nas organizações se baseavam em técnicas de arquivamento e recuperação de informações de grandes arquivos. Geralmente existia a figura do “arquivador”, que era a pessoa responsável em organizar os dados, registrá-los, catalogá-los e recuperá-los quando necessário. Esse método, apesar de simples, exigia um grande esforço para manter os dados actualizados bem como recuperá-los. As informações em papéis também não possibilitavam a facilidade de cruzamento e análises dos dados. Por exemplo o inventário de stock de uma empresa não era uma tarefa trivial nessa época, sendo que a actualização dos dados não era uma tarefa prática e quase sempre envolvia muitas pessoas, aumentando a probabilidade de ocorrerm erros. Com isso, percebe-se que a pré-história dos sistemas de informação foi marcada pela simplicidade dos dados, informações, métodos e técnicas assim como pela limitação do sistema e pela sua ineficiência. (Portogente.com. br. 01 de Janeiro de 2016).

Actualmente com a implementação de novas tecnologias os sistemas de informação ganharam diferentes requisitos funcionais concordando, no entanto que este é um sistema cujo principal objectivo é gerenciar e prover o ascesso a um banco de dados de informações. As questões em sistemas de informação incluem usabilidade, privacidade e manutenção da integridade dos dados (Sommerville, 2011).

**(NOTA:Resto do conteúdo dependendo de informações de livros e TCC´s)**

## **2.1 Sistema**

A palavra sistema vem do grego synístanai, que significa colocar junto “ao mesmo tempo” e expressa união de elementos de forma coordenada por isso normalmente entendemos que um siStema é um conjunto de elementos que se relacionam entre si afim de manter interação (Maria Emília Rodrigues Regina. 11 de agosto de 2021).

Em informática o termo sistema pode se referir a softwares, programas, ou aplicativos; que normalmente referem-se a mesma coisa, levando em conta situações em que o uso de determinado termo serve para indicar certa particularidade no funciomamento.

Se olharmos pra o sistema como software eles são classificados por:

* Software Aplicativo
* Software de Programação
* Software de Sistema

Entretanto quando olhamos como aplicativo são classificados como:

* Aplicativo móvel ou App
* Apliicativo desktop
* Aplicativo Web

## **2.2 Sistema Web**

Os sites são uma colecção de páginas web organizadas e localizadas num servidor com conteúdo expositivo e estático, quando olhamos pra um sistema (aplicativo) web apesar de ser um website notamos a adesão de novas tecnologias que a deixam capaz de manipular e armazenar determinados tipos de dados, tornando o website num software hospedado na internet onde qualquer usuário cadastrado pode acessá-lo através de um navegador.

Sua acessibilidade torna-a uma plataforma independe de acesso, ou seja, não depende das configuraçõe de um computador para ser acessada, o que facilita aos usuários do sistema manter a interação.

## **2.3 Sistema Web para marcação de consultas**

Este tipo de sistema é uma plataforma digital que aproxima pacientes ás instituições de saúde. Ela permite que o próprio paciente, onde e quando estiver, agende suas consultas nos horários disponíveis.

É uma forma de optimizar uma etapa tradicionalmente manual, agilizando o processo de marcação, confirmação e também cancelamento de consultas. Depois de a consulta ser confirmada, a agenda do médico é automaticamente atualizada. Desse modo, a data marcada é automaticamente sincronizada com a agenda do profissional e da unidade hospitalar

## **2.4 Benefícios de um Sistema Web para marcação de consultas**

O grande volume de informações e procedimentos tornam num desafio para as unidades hospitalares proporcionarem rapidez no atendimento às pessoas. Deste modo, um **sistema de marcação de consulta** auxília para que esse objectivo seja alcançado. Por permitir uma maior organização de horários ele melhora a rotina de médicos e pacientes. Seu funcionamento facilita e torna digital as interações mais básicas entre eles, e as unidades hospitalares.

A utilização deste sistema permitirá:

* Melhoria na experiência do paciente - Por ter a consulta agendada ele poderá Contar com um acompanhamento médico mais cuidado e atensioso;
* Reduzir a fila de espera - a possibilidade do agendamento digital facilita a vida dos pacientes que buscam rapidez e praticidade na marcação;
* Um canal a mais de agendamento - m disso, é possível aumentar o número de atendimentos sem a necessidade de investir no crescimento da equipa;
* Redução de erros - Por ser automatizado, garante qualidade e precisão nos dados fornecidos, conta com maior padronização no sistema e na organização evitando aspectos como rasuras e marcações sobrepostas, evita também que sejam marcados procedimentos para dias em que a equipe não estará no local;
* Preenchimento de horários vagos com agilidade - Quando não há um planejamento estratégico para lidar com eventualidades, os horários ficam ociosos e há impacto no faturamento daquele período Portanto, quando um paciente faz o cancelamento da sua marcação, rapidamente o horário deve estar novamente disponível para ser pree3nchido por outra demanda;
* Liberdade de escolha e mobilidade - A partir do momento em que o paciente pode visualizar em uma tela os horários disponíveis, ele rapidamente seleciona aquele que mais se encaixa à sua necessidade. Isso confere ao paciente maior liberdade no momento da marcação;

# 

# CAPÍTULO 3: METODOLOGIA

## **3.1 Metodologia de investigação científica**

A metodologia é a aplicação de procedimentos e técnicas que devem ser observadas para a construção de um conhecimento, com o propósito de comprovar a sua validade e utilização nos diversos ámbitos da sociedade, tem uma grande importância para os estudos académicos por permitir aos estudantes métodos para construir conhecimentos científicos. A finalidade de pesquisa é resolver problemas e solucionar dúvidas, mediante a utilização de procedimentos ciéntificos (BARROS; LEHFELD, 2000a, p.14).

Segundo Tumelero (data) a metodologia de investigaçãociêntifica é a descrição do processo de pesquisa do trabalho, a definição de quais serão os procedimentos para a coleta e análise dos dados que se adapte ao objectivo do trabalho.

Deste modo entende-se que “o planejamento de uma pesquisa depende tanto do problema a ser estudado, da sua natureza e situação espaço-temporal em que se encontra, quanto da natureza e nível de conhecimento do pesquisador.” (KÖCHE, 2007, p. 122).

As pesquisas podem ser classificadas Quanto:

* A Natureza
* Aos Objectivos
* Aos procedimentos técnicos
* A sua forma de abordagem

A Imagem 1 demonstra com mais detalhe a classificação dos tipos de pesquisa:

**(Pretendemos colocar uma imagem , será que todas imagens vão para os anexos)**

Do ponto de vista de sua natureza esta pesquisa identifica-se como aplicada. Verificando que seu objectivo é criar um sistema web executável para marcações de consultas online. De acordo com a visão defendida por Prodanov e Freitas em que definem pesquisa aplicada como sendo objectiva a gerar conhecimentos para aplicação prática dirigida à solução de problemas específicos.

Para construirmos a mesma foi necessário fazer uma “pesquisa exploratória que tem como finalidade proporcionar mais informações sobre o assunto que vamos investigar, possibilitando a delimitação do tema, ela possui planejamento flexíveis, o que permite o estudo do tema sob diversos ângulos e aspectos” (PRODANOV e FREITAS, 2013). Afim de compreendermos melhor os problemas a serem investigados, nesta etapa realizamos várias entrevistas a Direção administrativa, com alguns médicos, enfermeiros e pacientes do hospital Josina Machel; deste modo foi necessário realizar-se também uma revisão bibliográfica que definiu o procedimento técnico a se usar, fizemos uma busca de materiais autênticos de autores que já se referiram ao assunto em estudo, o mesmo permitiu coletar informações sobre o processo de marcação de consultas, os princípios de conectividade, aplicativos móveis e web para a saúde, com ênfase no sistema web para marcação de consulta.

## **3.2 Metodologia de desenvolvimento**

Segundo Sommerville (2011) um dos pontos importantes em engenharia de software é perceber que um Software não é apenas um programa ou programas; ele inclui também a documentação; destaca ainda que os atributos principais de um produto de software são a manutenção, confiaça, proteção, eficiência e aceitabilidade.

Deste modo ressaltamos a importância em usarmos uma metodologia, ou seja, processos e métodos já criados para desenvolvermos um software.

Um processo de software é um conjunto de actividades relacionadas que levam a produção de um produto de software. Essas actividades podem envolver o desenvolvimento de software em uma linguagem padrão de programação como Java ou C (SOMMERVILLE, 2011).

Sommerville (2011) assim como outros investigadores apoiam a ideia de que apesar de existir vários métodos de desenvolvimento ou seja processos de software todos eles 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 do software – O software deve ser produzido para atender as especificações.
3. Validação do software – O software deve ser validado para garantir que atenda às demandas do cliente.
4. Evolução de software – O software de evoluir para atender às necessidades de mudanças dos clientes.

Os processos de software, às vezes, são categorizados como dirigidos a planos ou processos ágeis. Processos dirigidos a planos são aqueles em que todas as atividades são planejadas com antecedência, e o progresso é avaliado por comparação com o planejamento inicial. Em processos ágeis, o planejamento é gradativo, e é mais fácil alterar o processo de maneira a reﬂetir as necessidades de mudança dos clientes. Conforme Boehm e Turner (2003), cada abordagem é apropriada para diferentes tipos de software. Geralmente, é necessário encontrar um equilíbrio entre os processos dirigidos a planos e os processos ágeis (SOMMERVILLE, 2011).

Existem diversos modelos para representamos de forma simplificada um processo de software. Assim sendo cada modelo representa uma perspectiva particular de um processo e, portanto, fornece informações parciais sobre ele. Por exemplo, um modelo de atividade do processo pode mostrar as atividades e sua sequência, mas não mostrar os papéis das pessoas envolvidas nelas (SOMMERVILLE, 2011).

**Modelo Em Cascata**

O primeiro modelo do processo de desenvolvimento de software a ser publicado foi derivado de processos mais gerais da engenharia de sistemas (ROYCE, 1970). 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 (SOMMERVILLE, 2011).

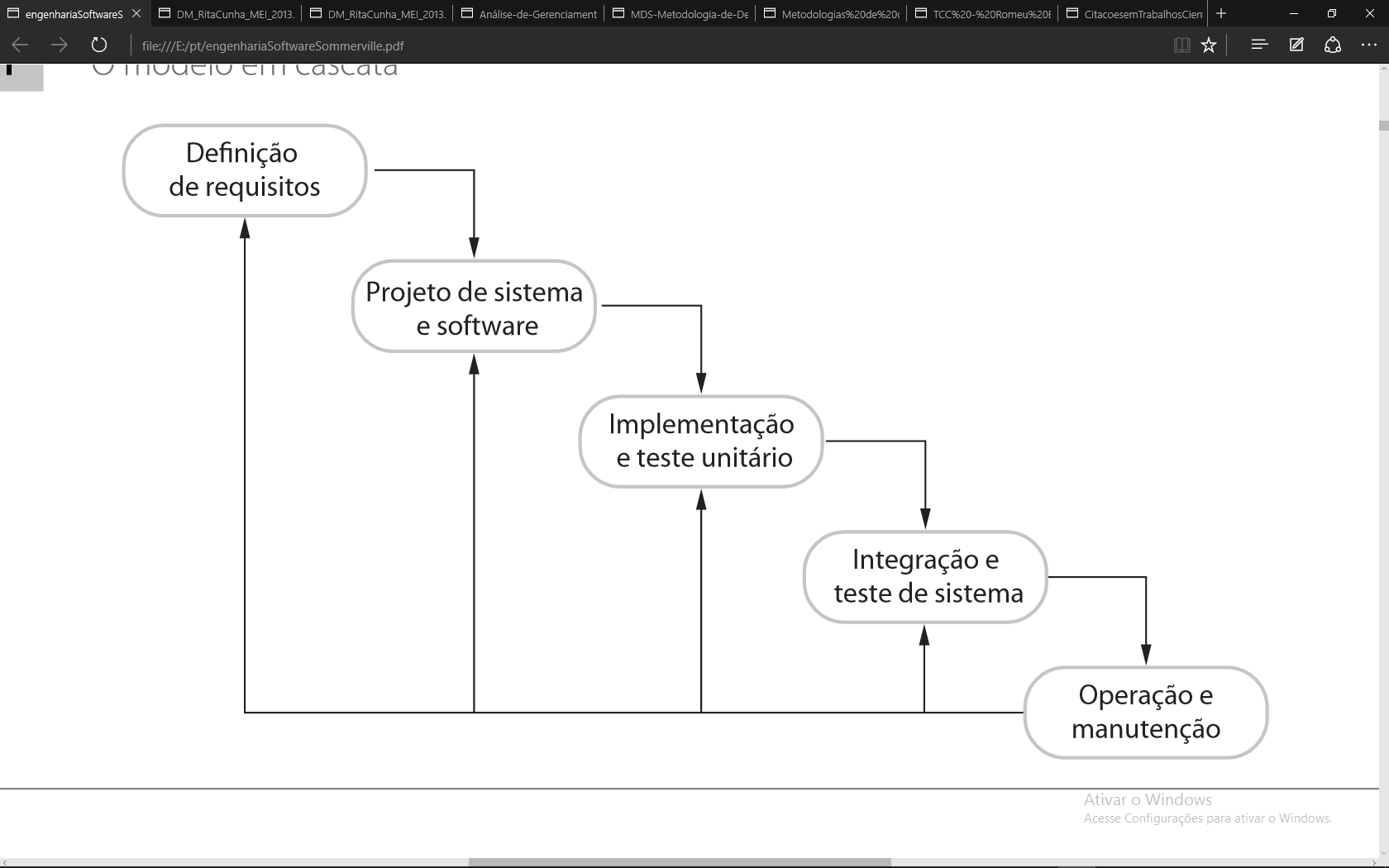
**Fases do modelo cascata**

Apoiando o que diz Sommerville (2011), o modelo em cascata é composto pelas seguintes fases:

1. **Análise e definição de requisitos**: Os serviços, restrições e metas do sistema são estabelecidos por meio de consulta aos usuários. Em seguida, são definidos em detalhes e funcionam como uma especificação do sistema.
2. **Projeto de sistema e software:** O processo de projeto de sistemas aloca os requisitos tanto para sistemas de hardware como para sistemas de software, por meio da definição de uma arquitetura geral do sistema. O projeto de software envolve identificação e descrição das abstrações fundamentais do sistema de software e seus relacionamentos.
3. **Implementação e teste unitário**: Durante esse estágio, o projeto do software é desenvolvido como um conjunto de programas ou unidades de programa. O teste unitário envolve a verificação de que cada unidade atenda a sua especificação.
4. **Integração e teste de sistema**: As unidades individuais do programa ou programas são integradas e testadas como um sistema completo para assegurar que os requisitos do software tenham sido atendidos. Após o teste, o sistema de software é entregue ao cliente.
5. **Operação e manutenção**: Normalmente (embora não necessariamente), essa é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em uso. A manutenção envolve a correção de erros que não foram descobertos em estágios iniciais do ciclo de vida, com melhora da implementação das unidades do sistema e ampliação de seus serviços em resposta às descobertas de novos requisitos.

Como é ilustrado na imagem neste modelo as actividades são desenvolvidas de forma sequencial

Figura 1 – Fases do Modelo Cascata

Fonte: SOMMERVILLE (2011)

**Desvantagens do Modelo Cascata**

De acordo com Moreira (2011) o modelo cascata Somente é permitida a participação do cliente no início e no final do projeto, e é muito difícil o cliente passar todas as necessidades do sistema logo no início, outro problema é que ele só terá acesso ao sistema no final do projeto e só aí os erros serão encontrados. (PRESSMAN, 2010). Moreira destaca ainda que nem sempre os prazos de entrega são cumpridos, pois segundo Pressman (2010) “Projetos reais raramente seguem o fluxo sequencial que o modelo propõe”. A estrutura sequencial não permite voltar atrás para corrigir falhas.

**Modelo Incremental**

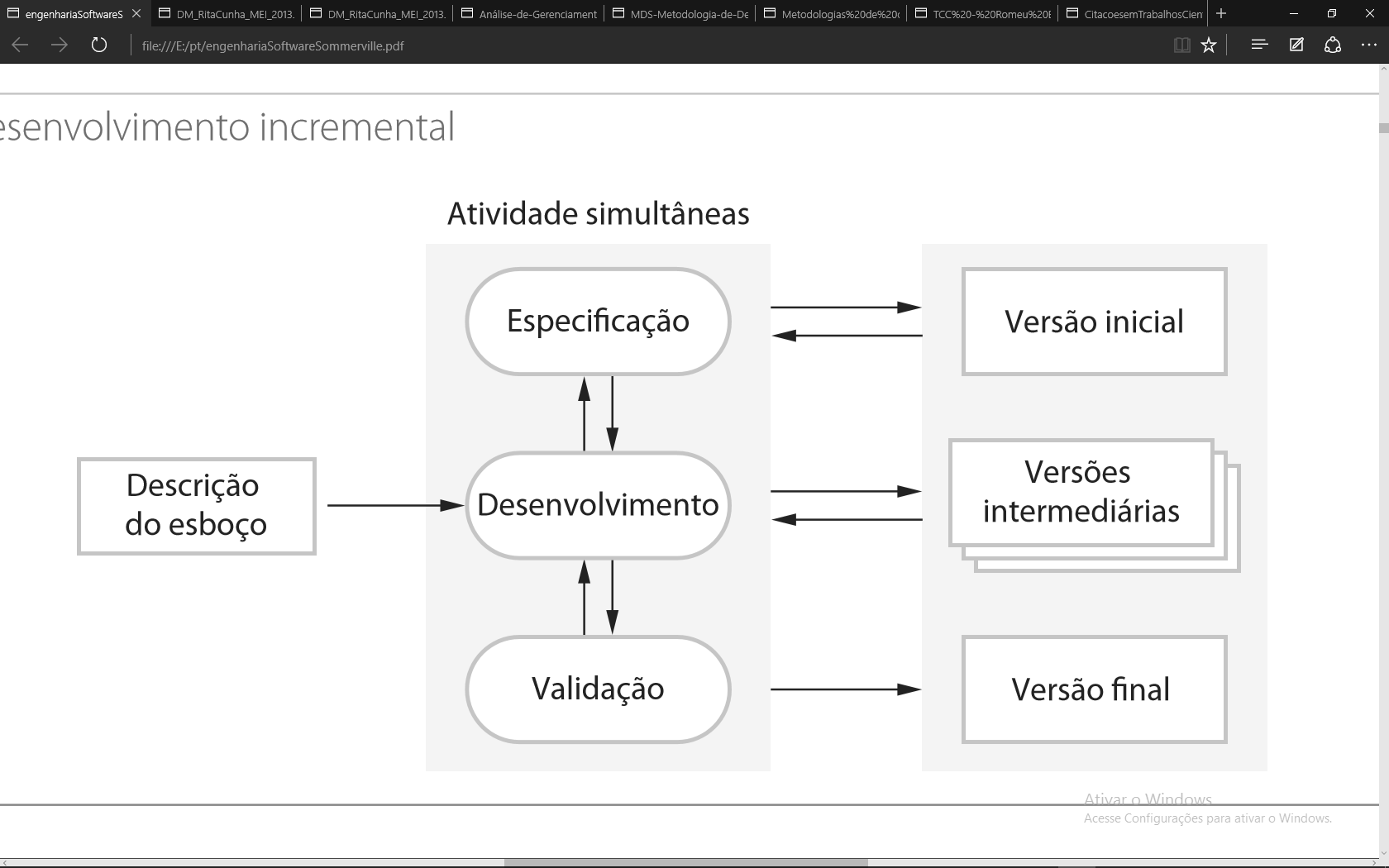
O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido. Atividades de especificação, desenvolvimento e validação são intercaladas, e não separadas, com rápido feedback entre todas as atividades (SOMMERVILLE, 2011).

Este tipo de desenvolvimento de software, que é uma parte fundamental das abordagens ágeis, é melhor do que uma abordagem em cascata para a maioria dos sistemas de negócios, e-commerce e sistemas pessoais. O mesmo reflete a maneira como resolvemos os problemas. Raramente elaboramos uma completa solução do problema com antecedência; geralmente movemo-nos passo a passo em direção a uma solução, recuando quando percebemos que cometemos um erro. Ao desenvolver um software de forma incremental, é mais barato e mais fácil fazer mudanças no software durante seu desenvolvimento. Cada incremento ou versão do sistema incorpora alguma funcionalidade necessária para o cliente. Frequentemente, os incrementos iniciais incluem a funcionalidade mais importante ou mais urgente. Isso significa que o cliente pode avaliar o sistema em um estágio relativamente inicial do desenvolvimento para ver se ele oferece o que foi requisitado; em caso negativo, só o incremento que estiver em denvolvimento no momento precisará ser alterado e, possivelmente, nova funcionalidade deverá ser definida para incrementos posteriores (SOMMERVILLE, 2011).

Novamente apoiando o que diz Sommerville (2011) o desenvolvimento incremental tem três vantagens importantes quando comparado ao modelo em cascata:

1. **O custo de acomodar as mudanças nos requisitos do cliente é reduzido**: A quantidade de análise e documentação a ser refeita é muito menor do que o necessário no modelo em cascata.
2. **É mais fácil obter feedback dos clientes sobre o desenvolvimento que foi feito**: Os clientes podem fazer comentários sobre as demonstrações do software e ver o quanto foi implementado. Os clientes têm dificuldade em avaliar a evolução por meio de documentos de projeto de software.
3. **É possível obter entrega e implementação rápida de um software útil ao cliente, mesmo se toda a funcionalidade não for incluída**: Os clientes podem usar e obtiver ganhos a partir do software inicial antes do que é possível com um processo em cascata. O desenvolvimento incremental, atualmente, é a abordagem mais comum para o desenvolvimento de sistemas aplicativos.

Figura 1.1: Fases do Modelo Incremental

Fonte: SOMMERVILLE (2011)

**Desvantagens do modelo incremental**

Quando comparada com a metodologia em cascata o modelo incremental apresenta algumas vantagens, principalmente por permitir ao desenvolvedor uma concepção rápida do software, apesar disto o autor (SOMMERVILLE, 2011) define que do ponto de vista do gerenciamento, a abordagem incremental tem dois problemas:

1. **O processo não é visível:** Os gerentes precisam de entregas regulares para mensurar o progresso. Se os sistemas são desenvolvidos com rapidez, não é economicamente viável produzir documentos que reflitam cada uma das versões do sistema.
2. **A estrutura do sistema tende a se degradar com a adição dos novos incrementos:** A menos que tempo e dinheiro sejam dispendidos em refatoração para melhoria do software, as constantes mudanças tendem a corromper sua estrutura. Incorporar futuras mudanças do software torna-se cada vez mais difícil e oneroso. Os problemas do desenvolvimento incremental são particularmente críticos para os sistemas de vida-longa, grandes e complexos, nos quais várias equipes desenvolvem diferentes partes do sistema. Sistemas de grande porte necessitam de um framework ou arquitetura estável, e as responsabilidades das diferentes equipes de trabalho do sistema precisam ser claramente definidas, respeitando essa arquitetura. Isso deve ser planejado com antecedência, e não desenvolvido de forma incremental.

**Modelo RUP**

O Rational Unified Process — RUP (KRUTCHEN, 2003) é um exemplo de modelo de processo moderno, derivado de trabalhos sobre a UML e o Unified Software Development Process associado (RUMBAUGH, et al., 1999; ARLOW e NEUSTADT, 2005). Incluí uma descrição aqui, pois é um bom exemplo de processo híbrido. Ele reúne elementos de todos os modelos de processo genéricos, ilustra boas práticas na especificação e no projeto e apoia a prototipação e a entrega incremental (SOMMERVILLE, 2011).

No entanto, o mesmo autor frisa ainda que este modelo é normalmente descrito em três perspectivas:

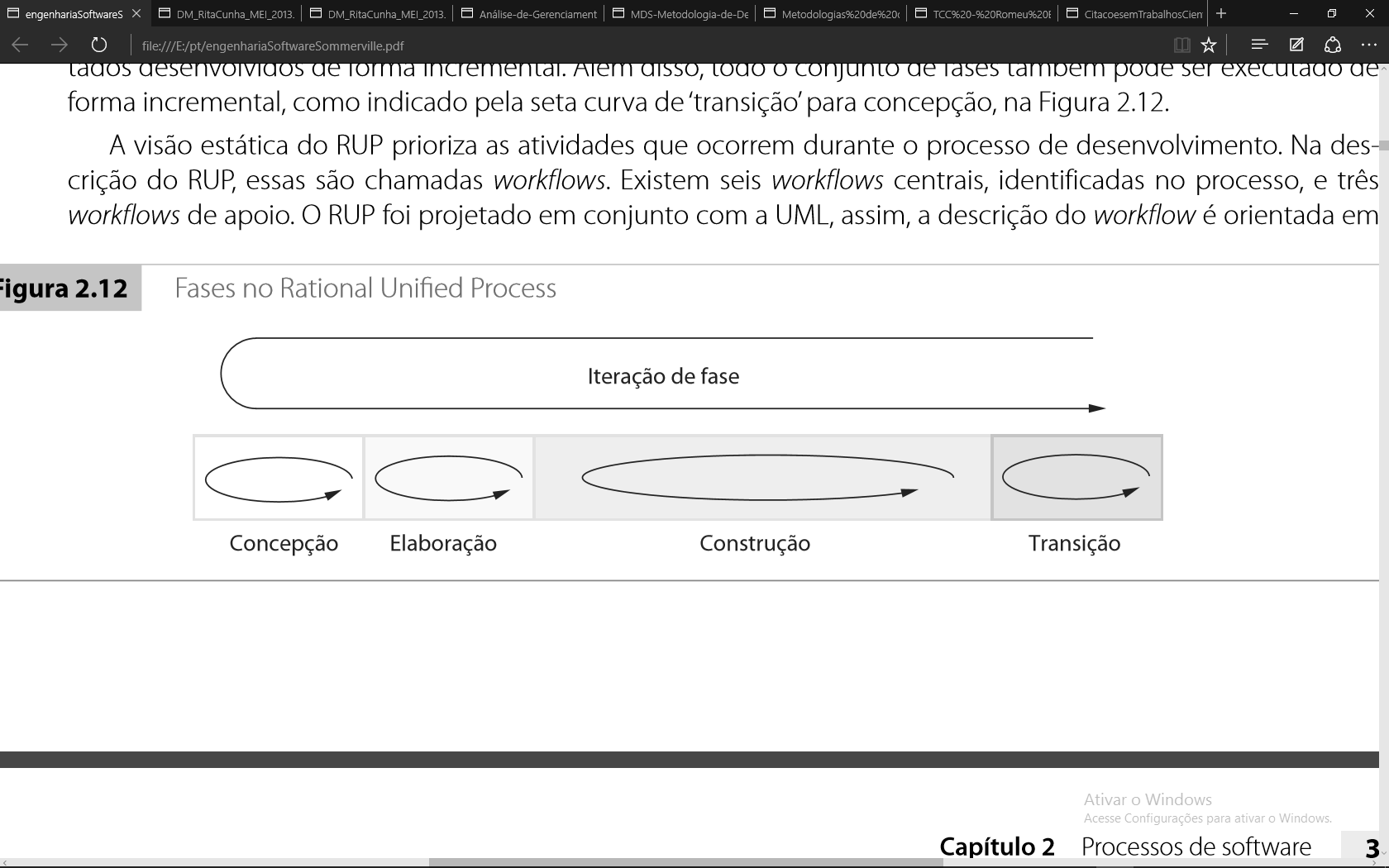
1. Uma perspectiva dinâmica, que mostra as fases do modelo ao longo do tempo.
2. Uma perspectiva estática, que mostra as atividades relizadas no processo.
3. Uma perspectiva prática, que sugere boas práticas a serem usadas durante o processo.

É notável que em vantagem á outros modelos tradicionais a maioria das descrições do RUP tenta combinar as perspectivas estática e dinâmica em um único diagrama (KRUTCHEN, 2003). No entanto, ao contrário do modelo em cascata, no qual as fases são equalizadas com as atividades do processo, as fases do RUP são estreitamente relacionadas ao negócio, e não a assuntos técnicos (SOMMERVILLE, 2011).

As fases do modelo RUP São:

1. **Concepção**: O objetivo da fase de concepção é estabelecer um business case para o sistema. Você deve identificar todas as entidades externas (pessoas e sistemas) que vão interagir com o sistema e definir as interações. Então, você deve usar essas informações para avaliar a contribuição do sistema para o negócio. Se essa contribuição for pequena, então o projeto poderá ser cancelado depois dessa fase.
2. **Elaboração**: As metas da fase de elaboração são desenvolver uma compreensão do problema dominante, estabelecer um framework da arquitetura para o sistema, desenvolver o plano do projeto e identificar os maiores riscos do projeto. No fim dessa fase, você deve ter um modelo de requisitos para o sistema, que pode ser um conjunto de casos de uso da UML, uma descrição da arquitetura ou um plano de desenvolvimento do software.
3. **Construção**: A fase de construção envolve projeto, programação e testes do sistema. Durante essa fase, as partes do sistema são desenvolvidas em paralelo e integradas. Na conclusão dessa fase, você deve ter um sistema de software já funcionando, bem como a documentação associada pronta para ser entregue aos usuários.
4. **Transição**: A fase final do RUP implica transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários e em seu funcionamento em um ambiente real. Isso é ignorado na maioria dos modelos de processo de software, mas é, de fato, uma atividade cara e, às vezes, problemática. Na conclusão dessa fase, você deve ter um sistema de software documentado e funcionando corretamente em seu ambiente operacional.

Figura 1.2: Fases do Modelo RUP



Fonte: SOMMERVILLE (2011)

No RUP, a iteração é apoiada de duas maneiras. Cada fase pode ser executada de forma iterativa com os resultados desenvolvidos de forma incremental. Além disso, todo o conjunto de fases também pode ser executado de forma incremental, como indicado pela seta curva de ‘transição’ para concepção, na Figura 2.12. A visão estática do RUP prioriza as atividades que ocorrem durante o processo de desenvolvimento. Na descrição do RUP, essas são chamadas workflows. Existem seis workflows centrais, identificadas no processo, e três workflows de apoio (SOMMERVILLE, 2011).

Sommerville (2011) ressalta que neste modulo a vantagem de proporcionar visões estáticas e dinâmicas é que as fases do processo de desenvolvimento não estão associadas a workflows específicos. Ao menos em princípio, todos os workflows do RUP podem estar ativos em todas as fases do processo. Nas fases iniciais, provavelmente, maiores esforços serão empenhados em workflows, como modelagem de negócios e requisitos, e, nas fases posteriores, no teste e na implantação. A perspectiva prática sobre o RUP descreve as boas práticas da engenharia de software que são recomendadas para uso no desenvolvimento de sistemas. Seis boas práticas fundamentais são recomendadas:

1. Desenvolver software iterativamente. Planejar os incrementos do sistema com base nas prioridades do cliente e desenvolver os recursos de alta prioridade no início do processo de desenvolvimento.
2. Gerenciar os requisitos. Documentar explicitamente os requisitos do cliente e acompanhar suas mudanças. Analisar o impacto das mudanças no sistema antes de aceitá-las.
3. Usar arquiteturas baseadas em componentes. Estruturar a arquitetura do sistema em componentes, conforme discutido anteriormente neste capítulo.
4. Modelar o software visualmente. Usar modelos gráficos da UML para apresentar visões estáticas e dinâmicas do software.
5. Verificar a qualidade do software. Assegurar que o software atenda aos padrões de qualidade organizacional.
6. Controlar as mudanças do software. Gerenciar as mudanças do software, usando um sistema de gerenciamento de mudanças e procedimentos e ferramentas de gerenciamento de configuração.

**Desvantagens do modelo RUP**

O RUP foi projetado em conjunto com a UML, que por sua vez permite ao desenvolvedor uma melhor interpretação dos requisitos do sistema a ser desenvolvido, porém apesar desta e outras vantagens o RUP não é um processo adequado para todos os tipos de desenvolvimento, como, por exemplo, desenvolvimento de software embutido assim como citou (SOMMERVILLE, 2011).

**Desenvolvimento ágil de software**

O desenvolvimento ágil tornou-se bastante popular na indústria do software. Valores como rapidez, flexibilidade e a capacidade de resposta são as principais razões desta popularidade. Ao contrário dos métodos de desenvolvimento tradicionais, cujo foco é nos processos e ferramentas, esta abordagem dá especial ênfase às equipas, ao software funcional, à colaboração com o cliente e à resposta às mudanças sendo estes os quatro valores principais do manifesto proposto em fevereiro de 2001, por um grupo de 17 peritos em desenvolvimento de software que definiram um conjunto de boas práticas de desenvolvimento de software, tornando-se uma mais-valia face a outras abordagens (CUNHA, 2013).

Os métodos ágeis de desenvolvimento de softwares surgiram com a necessidade de tornar o desenvolvimento de software mais leve, flexível a mudanças, sem o aumento exponencial dos custos em contrapartida dos métodos tradicionais que desperdiçavam muito tempo em análise e planejamento (SAVOINE,).

Actualmente em engenharia de software as metodologias ágeis XP e Scrum são as destacadas como principais.

O Manifesto Ágil define explicitamente quatro valores (acima apresentados) que devem ser considerados ao longo do desenvolvimento ágil de software. Contudo, os quatro valores são suportados por doze princípios. Na Tabela 1 encontra-se um resumo dos princípios ágeis comuns a todas as metodologias ágeis.

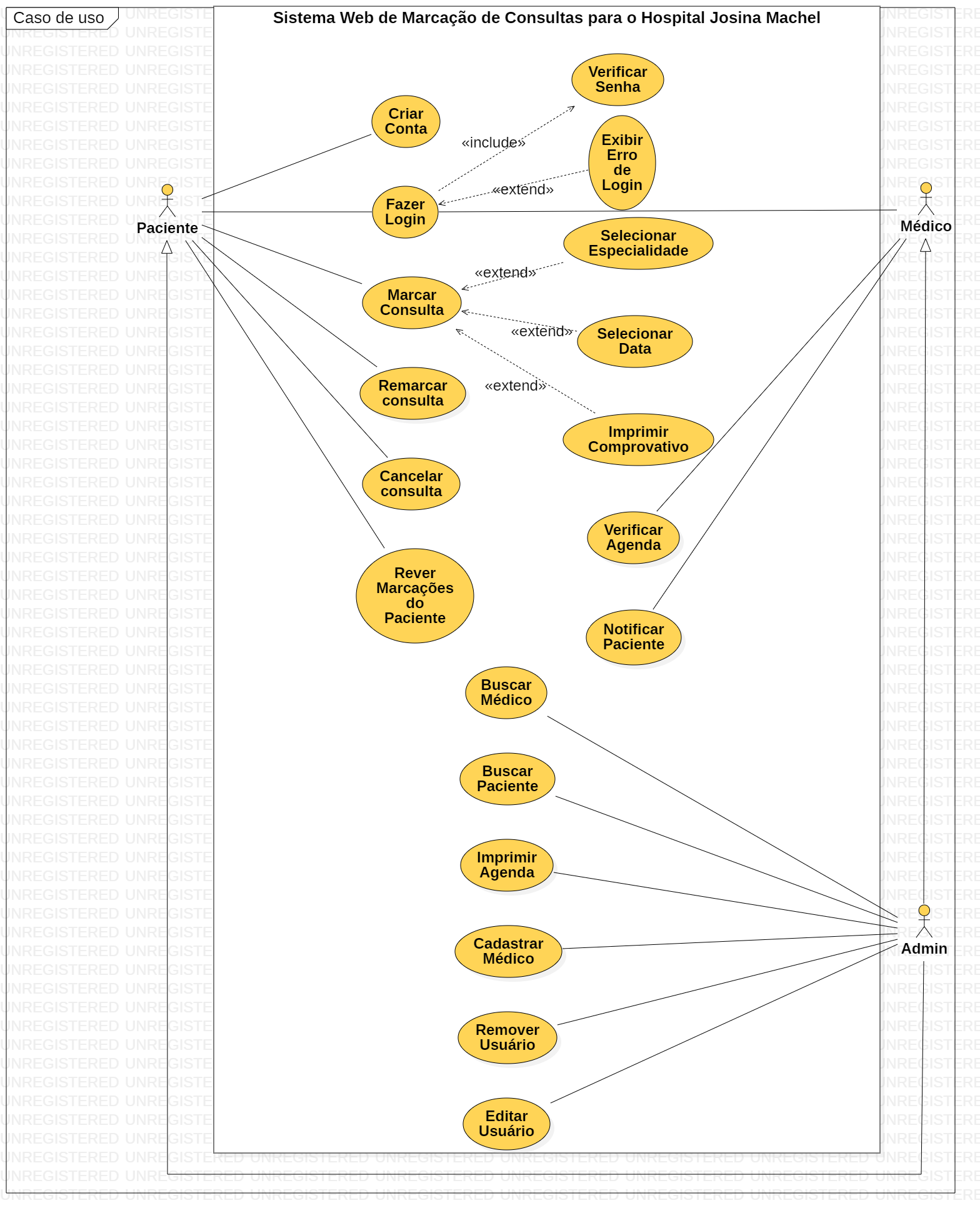
**Diagrama De Casos De Uso**

Este elemento é responsável em documentar o que o sistema faz do ponto de vista do usuário, ele descreve as principais funcionalidades do sistema e a interação destas com o usuário.

Um caso de uso é um recurso para documentar com máximo de detalhes uma funcionalidade do sistema. Na documentação é descrita quem irá utilizar as funcionalidade que são os atores; fluxos para utilização; pré-condições e qual será o resultado final da funcionalidade.

Essa etapa foi dividida em duas partes o diagrama, responsável por representar as funções que cada usuário pode utilizar no sistema; e parte dos casos de uso, que descreve cada uma dessas funções. Assim no final desta obtivemos uma visão prévia de como será o sistema ().

**(NOTA:Resto do conteúdo dependendo de informações de livros e TCC´s, afim de dar mais constência a informação)**

****

**Figura 01.** Diagrama de Casos de Uso

**ESPECIFICAÇÕES DE CASOS DE USO**

**Use Case – Cadastrar Médico**

|  |  |
| --- | --- |
| **Nome da Use Case** | Cadastar Médico |
| **Descrição** | Este caso de uso serve para realizar o cadastro do médico no sistema |
| **Ator** | Administrador do sistema |
| **Pré-condições** | Este caso de uso só inicia se:   * O médico não estiver cadastrado no sistema |
| **Pós-condições** | Quando termina o caso de uso o sistema é responsável por:   * Guardar o registro e todos os dados com consistência |
| **Fluxo de Eventos Principais** | 1. “O Administrador seleciona a opção “Cadastrar” no painel Administrativo”. 2. O Administrador clica na opção Cadastar Médico. 3. O Sistema exibe a tela de Cadastro para médicos. 4. O Adminstrador preenche todos os campos necessários. 5. O Administrador clica na opção finalizar. 6. O sistema verifica se os campos preenchidos estão correctos. 7. O sistema guarda o registro no banco de dados. 8. O sistema emite um alert com a mensagem “Médico cadastrado com sucesso”. 9. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Se pretender cancelar a operação: 2. O administrador seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial. |
| **Fluxos de Exceção** | 1. Campos não preenchidos 2. O sistema exibe a mensagem “Todos os campos devem ser preenchidos!”. 3. O caso de uso retorna ao passo 2 do fluxo principal. |
| **Regras de Negócio** | 1. Todos os campos devem ser preenchidos. |

**Use Case – Buscar Médico**

|  |  |
| --- | --- |
| **Nome da Use Case** | Buscar Médico |
| **Descrição** | Este caso de uso serve para realizar a pesquisa de médicos cadastrados no sistema. |
| **Ator** | Administrador do sistema. |
| **Pré-condições** | Este caso de uso só inicia se:   * O médico estiver cadastrado no sistema. |
| **Pós-condições** | Quando termina o caso de uso o sistema é responsável por:   * Apresentar uma tela com os resultados da pesquisa. |
| **Fluxo de Eventos Principais** | 1. “O Administrador seleciona a opção “Cadastro” no painel Administrativo”. 2. O Administrador clica na opção Cadastro de Médico. 3. O Sistema exibe na tela a lista de médicos cadastrados e uma barra de pesquisa. 4. O Adminstrador preenche os campos de pesquisa. 5. O Administrador clica na opção pesquisar. 6. Redireciona uma busca no bsnco de dados. 7. O sistema exibe uma lista de resultado com base os critérios da pesquisa. 8. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Usar a barra de pesquisa inicial: 2. O administrador preenche os campos de busca. 3. O administrador seleciona a opção Pesquisar. 4. O sistema exibe uma lista de médicos com base os critérios da pesquisa. |
| **Fluxos de Exceção** | 1. Campos não preenchidos 2. O sistema exibe a mensagem “Digite o médico na Barra de Pesquisa”.   **E2** Médico não encontrado   1. O sistema exibe a mensagem “Médico não encontrado”. |
| **Regras de Negócio** | 1. É importante saber os dados do médico. |

**Use Case – Buscar Paciente**

|  |  |
| --- | --- |
| **Nome da Use Case** | Buscar Paciente |
| **Descrição** | Este caso de uso serve para realizar a pesquisa de pacientescadastrados no sistema. |
| **Ator** | Administrador do sistema. |
| **Pré-condições** | Este caso de uso só inicia se:  O paciente estiver cadastrado no sistema. |
| **Pós-condições** | Quando termina o caso de uso o sistema é responsável por:  Apresentar uma tela com os resultados da pesquisa. |
| **Fluxo de Eventos Principais** | 1. “O Administrador seleciona a opção “Cadastro” no painel Administrativo”. 2. O Administrador clica na opção Cadastro de Paciente. 3. O Sistema exibe na tela a lista de pacientes cadastrados e uma barra de pesquisa. 4. O Adminstrador preenche os campos de pesquisa. 5. O Administrador clica na opção pesquisar. 6. Redireciona uma busca no banco de dados. 7. O sistema exibe uma lista de resultado com base os critérios da pesquisa. 8. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Usar a barra de pesquisa inicial: 2. O administrador preenche os campos de busca. 3. O administrador seleciona a opção Pesquisar. 4. O sistema exibe uma lista de pacientes com base os critérios da pesquisa. |
| **Fluxos de Exceção** | 1. Campos não preenchidos 2. O sistema exibe a mensagem “Digite o usuário na Barra de Pesquisa”. 3. Usuário não encontrado 4. O sistema exibe a mensagem “Paciente não encontrado”. |
| **Regras de Negócio** | 1. É importante saber os dados do paciente. |

**Use Case – Remover Usuário**

|  |  |
| --- | --- |
| **Nome da Use Case** | Remover Usuário |
| **Descrição** | Este caso de uso serve para o administrador remover algum usuário do sistema. |
| **Ator** | Administrador do sistema. |
| **Pré-condições** | Este caso de uso só inicia se:   * Houver algum usuário cadastrado no sistema. |
| **Pós-condições** | Quando termina o caso de uso o sistema é responsável por:   * Remover com consistência o usuário selecionado. |
| **Fluxo de Eventos Principais** | 1. “O Administrador seleciona a opção “Cadastro” no painel Administrativo”. 2. O Administrador seleciona a opção “Cadastro de Paciente” ou “Cadastro de Médico” de acordo ao objectivo. 3. O Sistema exibe na tela a lista de usuários de acordo a opção selecionada e uma barra de pesquisa. 4. O Administrador preenche o campo na barra de pesquisa. 5. O Administrador clica na opção pesquisar. 6. O sistema Redireciona uma busca no banco de dados. 7. O sistema exibe uma lista de resultado com base os critérios da pesquisa. 8. O Administrador seleciona o usuário. 9. O Administrador clica na opção “Remover”. 10. O Administrador clica na opção “Continuar”. 11. O sistema Remove o usuário do Banco de dados. 12. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Usar a barra de pesquisa inicial: 2. O administrador preenche os campos de busca. 3. O administrador seleciona a opção Pesquisar. 4. O sistema exibe uma lista de resultados com base os critérios da pesquisa. 5. O Administrador seleciona o usuário. 6. O Administrador clica na opção “Remover”. 7. O Administrador clica na opção “Continuar”. 8. O sistema Remove o usuário do Banco de dados. 9. O caso de uso termina. 10. O Administrador seleciona a opção “Cadastro” no painel Administrativo. 11. O Administrador seleciona a opção “Cadastro de Paciente” ou “Cadastro de Médico” de acordo ao objectivo. 12. O Sistema exibe na tela a lista de usuários de acordo a opção selecionada e uma barra de pesquisa. 13. O Administrador seleciona o usuário. 14. O Administrador segue a opção   “Remover”.   1. O Aministrador segue a opção “Continuar”. 2. O sistema Remove o usuário do Banco de dados. 3. O caso de uso termina. |
| **Fluxos de Exceção** | 1. Se pretender cancelar a operação: 2. O administrador seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial 4. Campos não preenchidos na barra de pesquisa. 5. O sistema exibe a mensagem “Digite o usuário na Barra de Pesquisa”. 6. Usuário não encontrado 7. O sistema exibe a mensagem “Usuário não encontrado”. |
| **Regras de Negócio** | 1. É importante saber os dados do usuário. |

**Use Case – Editar Usuário**

|  |  |
| --- | --- |
| **Nome da Use Case** | Editar Usuário |
| **Descrição** | Este caso de uso serve para o administrador Editar algum usuário do sistema. |
| **Ator** | Administrador do sistema. |
| **Pré-condições** | Este caso de uso só inicia se:   * Houver algum usuário cadastrado no sistema. |
| **Pós-condições** | Quando termina o caso de uso o sistema é responsável por:   * Editar com consistência o usuário selecionado. |
| **Fluxo de Eventos Principais** | 1. “O Administrador seleciona a opção “Cadastro” no painel Administrativo”. 2. O Administrador seleciona a opção “Cadastro de Paciente” ou “Cadastro de Médico” de acordo ao objectivo. 3. O Sistema exibe na tela a lista de usuários de acordo a opção selecionada e uma barra de pesquisa. 4. O Administrador preenche o campo na barra de pesquisa. 5. O Administrador clica na opção pesquisar. 6. O sistema Redireciona uma busca no banco de dados. 7. O sistema exibe uma lista de resultado com base os critérios da pesquisa. 8. O Administrador seleciona o usuário. 9. O Administrador clica na opção “Editar”. 10. O Administrador clica na opção “Continuar”. 11. O sistema actualiza as informações do usuário no Banco de dados. 12. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Usar a barra de pesquisa inicial: 2. O administrador preenche os campos de busca. 3. O administrador seleciona a opção Pesquisar. 4. O sistema exibe uma lista de resultados com base os critérios da pesquisa. 5. O Administrador seleciona o usuário. 6. O Administrador clica na opção “Editar”. 7. O Administrador clica na opção “Continuar”. 8. O sistema actualiza as informações do usuário no Banco de dados. 9. O caso de uso termina. 10. O Administrador seleciona a opção “Cadastro” no painel Administrativo. 11. O Administrador seleciona a opção “Cadastro de Paciente” ou “Cadastro de Médico” de acordo ao objectivo. 12. O Sistema exibe na tela a lista de usuários de acordo a opção selecionada e uma barra de pesquisa. 13. O Administrador seleciona o usuário. 14. O Administrador segue a opção   “Editar”.   1. O Aministrador segue a opção “Continuar”. 2. O sistema actualiza as informações do usuário no Banco de dados. 3. O caso de uso termina. |
| **Fluxos de Exceção** | 1. Se pretender cancelar a operação: 2. O administrador seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial 4. Campos não preenchidos na barra de pesquisa. 5. O sistema exibe a mensagem “Digite o usuário na Barra de Pesquisa”. 6. Usuário não encontrado 7. O sistema exibe a mensagem “Usuário não encontrado”. |
| **Regras de Negócio** | 1. É importante saber os dados do usuário. |

**Use Case – Criar Conta**

|  |  |
| --- | --- |
| **Nome da Use Case** | Criar Conta |
| **Descrição** | Este caso de uso serve para criar a conta do usuário. |
| **Ator** | Usuário |
| **Pré-condições** | Este caso de uso só inicia se:   * O usuário não estiver cadastrado no sistema. |
| **Pré-condições** | Quando termina o caso de uso o sistema é responsável por:   * Guardar o registro e todos os dados com consistência. |
| **Fluxo de Eventos Principais** | 1. “O Usuário seleciona a opção “Criar Conta” na página inicial”. 2. O Sistema exibe na tela o formulário de Cadastro para usuários. 3. O usuário preenche todos os campos necessários do formulário. 4. O usuário clica na opção finalizar. 5. O sistema verifica se os campos preenchidos estão correctos. 6. O sistema guarda o registro no banco de dados. 7. O sistema emite um alert com a mensagem “Usuário cadastrado com sucesso”. 8. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Se pretender cancelar a operação: 2. O administrador seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial. |
| **Fluxos de Exceção** | 1. Campos não preenchidos 2. O sistema exibe a mensagem “Todos os campos devem ser preenchidos!”. 3. O caso de uso retorna ao passo 2 do fluxo principal. |
| **Regras de Negócio** | * Todos os campos devem ser preenchidos. |

**Use Case – Fazer Login**

|  |  |
| --- | --- |
| **Nome da Use Case** | Fazer Login |
| **Descrição** | Este caso de uso serve para o usuário fazer o login de sua conta. |
| **Ator** | Usuário |
| **Pré-condições** | Este caso de uso só inicia se:   * O usuário estiver cadastrado no sistema. * A senha e o nome de usuário for correspondente. |
| **Pré-condições** | Quando termina o caso de uso o sistema é responsável por:   * Iniciar sessão da conta do usuário. |
| **Fluxo de Eventos Principais** | 1. “O Usuário seleciona a opção “Fazer Login” na página inicial”. 2. O Sistema exibe na tela o formulário de login para usuários. 3. O usuário preenche todos os campos necessários do formulário. 4. O usuário clica na opção continuar. 5. O sistema verifica se os campos preenchidos estão correctos. 6. O sistema guarda o registro no banco de dados. 7. O sistema emite um alert com a mensagem “Sessão iniciada com sucesso”. 8. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Se pretender cancelar a operação: 2. O administrador seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial. |
| **Fluxos de Exceção** | 1. Campos não preenchidos 2. O sistema exibe a mensagem “Todos os campos devem ser preenchidos!”. 3. O caso de uso retorna ao passo 2 do fluxo principal. 4. Dados inseridos incorrectos 5. O sistema exibe a mensagem “A senha ou o usuário está errado !”. |
| **Regras de Negócio** | * Todos os campos devem ser preenchidos. |

**Use Case – Marcar Consulta**

|  |  |
| --- | --- |
| **Nome da Use Case** | Marcar Consulta |
| **Descrição** | Este caso de uso serve para fazer o agendamento de consultas médicas. |
| **Ator** | Usuário |
| **Pré-condições** | Este caso de uso só inicia se:   * O usuário realizar o login de sua conta no sistema. |
| **Pré-condições** | Quando termina o caso de uso o sistema é responsável por:   * Guadar a marcação na agenda do médico selecionado(automaticamente) com todas as informações e detalhes destacado pelo usuário. |
| **Fluxo de Eventos Principais** | 1. “O Usuário seleciona a opção “Agendar Consulta” na página inicial. 2. O Sistema exibe na tela o formulário para a maração de consulta. 3. O usuário preenche todos os campos necessários do formulário. 4. O usuário clica na opção continuar. 5. O sistema verifica se os campos preenchidos estão correctos. 6. O sistema guarda o registro no banco de dados. 7. O sistema emite um alert com a mensagem “Agendamento realizado com sucesso”. 8. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Se pretender cancelar a operação: 2. O administrador seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial. |
| **Fluxos de Exceção** | 1. Campos não preenchidos 2. O sistema exibe a mensagem “Todos os campos obrigatórios devem ser preenchidos!”. 3. O caso de uso retorna ao passo 2 do fluxo principal. 4. Dados inseridos incorrectos 5. O sistema exibe a mensagem “A senha ou o usuário está errado !”. |
| **Regras de Negócio** | * Todos os campos obrigatórios devem ser preenchidos. |

**Use Case – Remarcar Consulta**

|  |  |
| --- | --- |
| **Nome da Use Case** | Remarcar Consulta |
| **Descrição** | Este caso de uso serve para Remarcar a consulta agendada. |
| **Ator** | Usuário |
| **Pré-condições** | Este caso de uso só inicia se:   * O usuário fazer o login no sitema * O usuário marcar alguma consulta. |
| **Pré-condições** | Quando termina o caso de uso o sistema é responsável por:   * Actualizar na agenda da consulta as informações e detalhes alterados pelo usuário. |
| **Fluxo de Eventos Principais** | 1. “O Usuário seleciona a opção “Consultas” na página inicial, barrra de menu. 2. O Sistema exibe as consultas marcadas(a agenda) do usuário. 3. O usuário seleciona a consulta á se remarcar. 4. O usuário escolhe a opção remarcar. 5. O usuário seleciona outra data disponível. 6. O usuário clica na opção continuar. 7. O sistema actualiza o campo no banco de dados. 8. O sistema emite um alert com a mensagem “Consulta remarcada com sucesso”. 9. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Se pretender cancelar a operação: 2. O usário seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial. |
| **Fluxos de Exceção** | 1. Mesma data selecionada 2. O sistema exibe a mensagem “Selecione outra data para remarcar a consulta!”. 3. O caso de uso retorna ao passo 2 do fluxo principal. 4. Dados inseridos incorrectos 5. O sistema exibe a mensagem “A senha ou o usuário está errado !”. |
| **Regras de Negócio** | * Todos os campos obrigatórios devem ser preenchidos. |

**Use Case – Cancelar Consulta**

|  |  |
| --- | --- |
| **Nome da Use Case** | Cancelar Consulta |
| **Descrição** | Este caso de uso serve para Cancelar a consulta agendada. |
| **Ator** | Usuário |
| **Pré-condições** | Este caso de uso só inicia se:   * O usuário fazer o login no sitema * O usuário marcar alguma consulta. |
| **Pré-condições** | Quando termina o caso de uso o sistema é responsável por:   * Eliminar na agenda do usuário a consulta selecionada. |
| **Fluxo de Eventos Principais** | 1. O Usuário seleciona a opção “Consultas” na página inicial, barrra de menu. 2. O Sistema exibe as consultas marcadas(a agenda) do usuário. 3. O usuário seleciona a consulta á se cancelar. 4. O usuário escolhe a opção cancelar consulta. 5. O usuário clica na opção continuar. 6. O sistema elimina o registro no banco de dados. 7. O sistema emite um alert com a mensagem “Consulta Cancelada com sucesso”. 8. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** | 1. Se pretender cancelar a operação: 2. O usário seleciona a opção “Voltar”. 3. O sistema cancela e retorna a tela inicial. |
| **Fluxos de Exceção** |  |
| **Regras de Negócio** | 1. Pelo menos uma consulta marcada. |

**Use Case – Rever Marcações**

|  |  |
| --- | --- |
| **Nome da Use Case** | Rever Marcações |
| **Descrição** | Este caso de uso serve para rever as consultas marcadas. |
| **Ator** | Usuário |
| **Pré-condições** | Este caso de uso só inicia se:   * O usuário fazer o login no sistema. * Tiver alguma consulta marcada. |
| **Pré-condições** | Quando termina o caso de uso o sistema é responsável por:   * Exibir as consultas marcadas(a agenda) do usuário. |
| **Fluxo de Eventos Principais** | 1. O Usuário seleciona a opção “Consultas” na página inicial, barrra de menu. 2. O Sistema exibe as consultas marcadas(a agenda) do usuário. 3. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** |  |
| **Fluxos de Exceção** |  |
| **Regras de Negócio** | 1. Pelo menos uma consulta marcada. |

**Use Case – Verificar Consultas Marcadas**

|  |  |
| --- | --- |
| **Nome da Use Case** | Verificar Consultas Marcadas |
| **Descrição** | Este caso de uso serve para verificar Consultas Marcadas rever as consultas marcadas. |
| **Ator** | Usuário |
| **Pré-condições** | Este caso de uso só inicia se:   * O usuário fazer o login no sistema. * Tiver alguma consulta marcada. |
| **Pré-condições** | Quando termina o caso de uso o sistema é responsável por:   * Exibir as consultas marcadas(a agenda) do usuário. |
| **Fluxo de Eventos Principais** | 1. O Usuário seleciona a opção “Consultas” na página inicial, barrra de menu. 2. O Sistema exibe as consultas marcadas(a agenda) do usuário. 3. O caso de uso termina. |
| **Fluxo de Eventos Alternativo** |  |
| **Fluxos de Exceção** |  |
| **Regras de Negócio** | 1. Pelo menos uma consulta marcada. |

**Atores**

Identificando os processos do negócio é possível determinar os atores e trabalhadores que intervêm em ditos processos. Ator de negócio representa um papel desempenhado em relação ao negócio por alguém ou algo no ambiente do negócio. Eis abaixo uma descrição sobre o ator do negócio.

Geralmente um trabalhador de negócio representa um papel desenvolvido por alguém ou algo dentro do negócio que realiza alguma atividade dentro do mesmo (negócio). Os trabalhadores de negócios interagem entre si e manipulam entidades de negócios enquanto participam de realizações de casos de uso de negócio. De acordo com a definição apresentada anteriormente a cerca do trabalhador do negócio, eis abaixo uma descrição sobre trabalhadores do negócio.