

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE TATUÍ
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA
INFORMAÇÃO

LUCAS SILVEIRA SBIZERA
VINÍCIUS BERNARDO

GESTÃO DE AGENDAMENTOS DE UMA BARBEARIA

Tatuí/SP
2º Semestre/2019

LUCAS SILVEIRA SBIZERA

VINÍCIUS BERNARDO

GESTÃO DE AGENDAMENTOS DE UMA BARBEARIA

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Tatuí, como exigência parcial para obtenção do grau de tecnólogo em gestão da tecnologia da informação sob orientação do Prof. Marcelo Akira Yamamoto.

Tatuí/SP

2º Semestre/2019

LUCAS SILVEIRA SBIZERA

VINÍCIUS BERNARDO

GESTÃO DE AGENDAMENTOS DE UMA BARBEARIA

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Tatuí, como exigência parcial para obtenção do grau de tecnólogo em gestão da tecnologia da informação sob orientação do Prof. Marcelo Akira Yamamoto.

() APROVADO () REPROVADO

Com média:.....

Tatuí, de 20.....

Prof.

FATEC – Tatuí

Prof.

FATEC – Tatuí

Prof.

FATEC – Tatuí

Tatuí/SP

2º Semestre/2019

Dedico este trabalho a minha família, que sempre esteve ao meu lado em todos os momentos, pelo estímulo e compreensão que me deram, e acima de tudo, pelo reconhecimento do meu trabalho. Dedico este trabalho a minha família, que sempre esteve ao meu lado em todos os momentos, pelo estímulo e compreensão que me deram, e acima de tudo, pelo reconhecimento do meu trabalho.

AGRADECIMENTOS

Não podíamos deixar de citar as pessoas que se fizeram tão presentes no decorrer da nossa vida acadêmica e principalmente na elaboração deste trabalho:

Ao Centro Paula Souza, pela oportunidade;

À Faculdade de Tecnologia de Tatuí, pela formação acadêmica;

Ao professor Marcelo Akira Yamamoto, pela sabedoria e paciência que soube dar forma às nossas ideias;

Aos nossos pais, pelo apoio, carinho e presença sempre que necessárias;

"Quando as pessoas estão determinadas, elas conseguem alcançar qualquer coisa."
(Nelson Mandela)

RESUMO

O presente trabalho tem como principal objetivo desenvolver um Sistema de Informações com o intuito de gerenciar o cadastro de clientes e serviços prestados por um salão de Barbearia por meio de aplicativo. Com o desenvolvimento deste trabalho foi desenvolvido um sistema de informações onde é possível realizar cadastros de clientes e serviços, e principalmente o agendamento para os serviços oferecidos. Dentre as funcionalidades do sistema, o usuário também poderá verificar os clientes e serviços cadastrados, os agendamentos e também estar gerando relatórios dos atendimentos. O sistema tem um layout bem simples de fácil compreensão e acesso com opções bem definidas e escolhida, sendo que é de fácil utilização pelo usuário. Após o desenvolvimento do sistema, a proprietária da Barbearia terá um melhor controle e gerenciamento dos horários dos serviços, bem como, os dados e informações importantes para o estabelecimento serão mantidas e armazenadas de modo seguro. O sistema foi desenvolvido na linguagem de programação java junto com o banco de dados MySQL.

Palavras-chaves: aplicativos; sistemas de informações; Barbearia; programação.

ABSTRACT

The present work has as main objective to develop an Information System in order to manage the registration of clients and services provided by a Barber Shop through application. With the development of this work was developed an information system where it is possible to perform customer and service registrations, and especially the scheduling for the services offered. Among the functionalities of the system, the user will also be able to check the registered customers and services, the schedules and also be generating reports of the calls. The system has a very simple layout that is easy to understand and access with well defined and chosen options, which is user friendly. After the development of the system, the owner of Barber Shop will have better control and management of service times, as well as important data and information for the establishment will be kept and stored securely. The system was developed in the java programming language along with the MySQL database.

Key-words: applications; information systems; barber shop; programming.

LISTA DE ILUSTRAÇÕES

Figura 1: Modelo de Caso de uso do Ator Barbeiro.....	30
Figura 2: Modelo de Caso de uso do Ator Cliente.....	31
Figura 3: Diagrama de sequência do relatório da demanda ativa.....	33
Figura 4: Diagrama de Classes (UML).....	37
Figura 5: Diagrama entidade relacionamento de sistema imobiliário.....	46
Figura 6: Diagrama entidade relacionamento	47
Figura 7: Diagrama entidade relacionamento (Variação).....	47
Figura 8: Atributos apresentados como elipses.....	48
Figura 9: Diagrama com atributos nas entidades.....	48
Figura 10: Diagrama de entidade-relacionamento (DER).....	49
Figura 11: Diagrama de entidade-relacionamento (Workbench).....	52
Figura 12: Cadastro do cliente	61
Figura 13: Cadastro do usuário	62
Figura 14: Cadastro da concessionária	64
Figura 15: Cadastro do dispositivo de medição	65
Figura 16: Cadastro de tarifas	66
Figura 17: Cadastro do dispositivo de medição.....	67
Figura 18: Cadastro do dispositivo de medição	68
Figura 19: Cadastro do dispositivo de medição.....	69
Figura 20: Cadastro do dispositivo de medição	70
Figura 21: Cadastro do dispositivo de medição.....	71
Figura 22: Cadastro do cliente.....	72
Figura 23: Cadastro do dispositivo de medição.....	73
Figura 24: Cadastro de tarifas.....	74
Figura 25: Cadastro do dispositivo de medição.....	75
Figura 26: Cadastro do dispositivo de medição.....	76
Figura 27: Consulta do Fator de Potência.....	77
Figura 28: Consulta do Fator de Carga.....	78
Figura 29: Consulta da Demanda de Consumo ativa.....	79
Figura 30: Procedimento de Contagem de Pontos de Função.....	82

LISTA DE TABELAS

Quadro 1: Caso de uso do ator usuário master.....	29
Quadro 2: Caso de uso do ator usuário operador.....	29
Quadro 3: Simbologia de relacionamentos das classes	38
Quadro 4: Objetos do diagrama de entidade-relacionamento.....	44
Quadro 5: Dicionário de dados da entidade Barbeiro.....	50
Quadro 6: Dicionário de dados da entidade Cliente	50
Quadro 7: Dicionário de dados da entidade Agendamento	50
Quadro 8: Dicionário de dados da entidade Serviço.....	50
Quadro 9: Dicionário de dados da entidade RealizaAgendamento	50
Quadro 10: Base de cálculo para complexidade de ALI e AIE.....	55
Quadro 11: Tabela para cálculo de complexidade de ALI e AIE.....	56
Quadro 12 - Arquivo lógico interno de Pedido.....	57
Tabela 13 - Arquivo lógico interno de Serviço.....	57
Tabela 14 - Arquivo lógico interno da realização de agendamento.....	58
Tabela 15 - Arquivo lógico interno de Produto.....	59
Tabela 16 - Arquivo lógico interno de Cliente.....	59
Tabela 17: Tabela de referência para DER e ALR para telas do programa.....	60
Tabela 18: Tabela de cálculo para aquisição dos pontos por função.....	83
Quadro 19: Valor de complexidade para respostas das questões.....	84
Tabela 25: Questionário.....	84
Tabela 26: Métricas de pontuação.....	87
Tabela 27: Resultados finais de pontuação.....	87
Tabela 28: Requisitos de hardware.....	88
Tabela 29: Requisitos de software.....	89

LISTA DE ABREVIATURAS E SIGLAS

MIT – Instituto de tecnologia de Massachusetts.

App - application, traduzindo para português significa aplicativo.

Vintage - Estilo antigo de design.

Feedback - Retorno ou resposta.

Sebrae - Serviço Brasileiro de Apoio às Micro e Pequenas Empresas

Sumário

1. INTRODUÇÃO	14
1.1 JUSTIFICATIVA	15
1.2 OBJETIVOS	16
1.2.1 Objetivo Geral	16
1.2.2 Objetivos Específicos	16
2. REFERENCIAL TEÓRICO	17
2.1 BELEZA E BARBEARIA	17
2.2 IMPORTÂNCIA DA GESTÃO NOS NEGÓCIOS	18
2.3 GESTÃO DE AGENDAMENTOS	18
3. MATERIAL E MÉTODOS	20
3.1 TECNOLOGIAS E FERRAMENTAS UTILIZADAS	20
3.2 ENGENHARIA DE SOFTWARE	23
3.3 REQUISITOS FUNCIONAIS DO SISTEMA	23
3.3.1 Cadastro de usuários barbeiros	23
3.3.2 Cadastro de clientes	24
3.3.3 Agendamentos	24
3.4 REQUISITOS NÃO-FUNCIONAIS DO SISTEMA	25
3.4.1 Confiabilidade	25
3.4.2 Tolerância a falhas	25
3.4.3 Portabilidade	25
3.4.4 Hardware	26
3.5 VISÃO DE CASO DE USO	26
3.5.1 Conceito de Caso de Uso	26
3.5.2 Operação do Sistema	27
3.5.3 Definição de Atores	27
3.5.4 Tabelas de Caso de Uso	28
3.5.5 Modelos de Caso de Uso	29
3.6 MODELOS DE SEQUÊNCIA	32
3.6.1 Diagrama de Sequência	33
3.7 VISÃO LÓGICA: NÍVEL DE ANÁLISE	35
3.8 RELACIONAMENTOS PRINCIPAIS AS CLASSES	36
3.9 DIAGRAMA DE CLASSES DO PROJETO (UML)	37
3.10 SIMBOLOGIA DOS RELACIONAMENTOS ENTRE CLASSES	38

3.11	MODELO ENTIDADE RELACIONAMENTO (MER)	39
3.11.1	Entidades	40
3.11.2	Relacionamentos	41
3.11.3	Atributos	42
3.12	DIAGRAMAS DE ENTIDADE E RELACIONAMENTO (DER)	43
3.13	RELACIONAMENTO ENTRE ENTIDADES	45
3.14	DIAGRAMA DE ENTIDADE-RELACIONAMENTO DO PROJETO	49
3.15	DICIONÁRIOS DE DADOS	50
3.16	DIAGRAMA DE ENTIDADE-RELACIONAMENTO (WORKBENCH)	53
3.17	VISÃO GERENCIAL: GERENCIAMENTO DE PROJETO	53
3.18	MÉTRICAS DE SOFTWARE	54
3.19	PONTOS POR FUNÇÃO	55
3.19.1	Arquivos Lógicos Internos (ALI)	55
3.19.2	Arquivos de Interface Externa (AIE)	59
3.19.3	Entradas Externas (EE)	60
3.19.4	Consultas Externas (CE)	76
3.20	PLANEJAMENTO POR DECOMPOSIÇÃO	79
3.20.1	Tabela de contagem	81
3.20.2	Questões de avaliação de complexidade de software	82
3.20.3	Cálculo do Fator de Ajuste	85
3.20.4	Produtividade, qualidade, preço e documentação	85
3.21	VISÃO SOBRE DISTRIBUIÇÃO E IMPLEMENTAÇÃO (DEPLOYMENT)	88
4.	CONSIDERAÇÕES FINAIS	89
	REFERÊNCIAS	92

1. INTRODUÇÃO

A importância da gestão para pequenos negócios se deve ao fato de a capacidade do microempresário dar conta de sua demanda e ser competitivo ao mesmo tempo. Um dos principais fatores que auxiliam nesse processo é automatização comercial que vem crescendo a passos largos no mercado atual.

De acordo com o Sebrae (2015), métodos utilizados para automatizar os processos (compras, vendas, controle de estoques, cadastro de clientes, consultas, etc.) fazem parte da automação de serviços, fazendo com que esses processos manuais se tornem mais mecanizados, rápidos e objetivos. Um software de automação comercial, ou aplicativo nesse caso, integra processos comerciais que vão desde procedimentos operacionais de gestão de estoque e aquisições até a venda e geração de notas fiscais, com geração de informações financeiras que serão essenciais para a gestão da empresa.

Por isso, a automatização comercial é indispensável para qualquer negócio, tradicional ou virtual. Isso é válido também para barbearias. As barbearias estão se tornando ambientes cada vez mais modernos e completos, um estilo mais clássico com bebidas em frigobares digitais transmite um conforto maior ao consumidor. Esse conforto e preferência também exigiria uma demanda maior para atender, o que automaticamente exige sistemas de gestão e de automação comercial mais ágeis para essa demanda ser atendida de forma eficaz.

No momento nesse mercado de salão de barbearia são poucos que trabalham com essa tecnologia e automatizam os seus processos, já em muitos salões de beleza ou barbearia os processos de gerenciamento é manual utilizando agenda de papel ou bloco de notas, podemos ver isso em nossa cidade. Os meios de comunicação são falhos pois acaba não tendo um feedback positivo além da falta de informação de horários disponíveis para atendimento e do horário de funcionamento do salão, organização do agendamento muito ruim que pode até diminuir os números de atendimento. Alguns profissionais tentam adaptar meios de comunicação como telefone ou aplicativos como WhatsApp, Messenger ou Instagram para auxiliar essa carência.

Atualmente na barbearia que vai ser desenvolvido o aplicativo a gestão é feita manualmente e exige alguns processos realizados pelo barbeiro para o atendimento acontecer. O primeiro passo é realizado pelo cliente, o agendamento manual feito por WhatsApp ou Telefone, o segundo passo é feito pelo barbeiro, checar a disponibilidade de horário e confirmar se pode ou não realizar o atendimento, o terceiro passo é o feedback do cliente e a confirmação de horário e disponibilidade para comparecer ao estabelecimento por ambas as partes, e muitas vezes ocorrem falhas de comunicação que atrapalham o processo operacional do barbeiro, como por exemplo, um imprevisto acontece e, como consequência, o cliente não comparece na hora agendada e aquele horário fica vazio, diminuindo o número de atendimentos e o valor do tempo do barbeiro.

Portanto, esse Projeto de Trabalho de Graduação pretende desenvolver um aplicativo de gestão de agendamentos de uma barbearia a fim de otimizar o processo, bem como garantir uma melhor interação com os clientes.

1.1 JUSTIFICATIVA

O mercado consumidor de uma barbearia é composto por homens, por mulheres e por crianças, isso nos leva a conclusão que toda a população de uma república pode ser considerada como potencial cliente. Nos últimos anos houve um grande aumento no número de estabelecimentos de barbearia no mercado. Segundo Sebrae (2019), De acordo com o Caderno de Tendências da Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos, para 2019, a expectativa é de que o mercado brasileiro masculino de cuidados pessoais passe de US\$ 6,7 bilhões. Com isso, o país deve se tornar o maior mercado do setor no mundo e esse pode ser um bom sinal para investir em uma carreira voltada para esse público.

Com este mercado em franco crescimento, observou-se a necessidade de melhorar o processo de agendamento de horário para serviços de barbearia. Grande parte dos salões de barbearias agendam os horários de forma manual, atendendo o

cliente pelo telefone e informando a disponibilidade de horário para o serviço solicitado. Em salões de pequeno porte que ainda não possuem um sistema informatizado, os agendamentos são registrados em agendas ou blocos de anotação de papel, podendo facilmente ser extraviados, gerando um transtorno tanto para o salão quanto para o cliente, uma caderneta e uma caneta não oferecem total confiabilidade nesse tipo de operação.

O sistema, portanto, irá melhorar a gestão, além de facilitar o controle por parte do proprietário, ou seja, uma facilidade que demonstrará a organização do salão e a qualidade de serviço, gerando para o cliente um olhar satisfatório e comprometido do estabelecimento. Pretende-se também que seja uma vantagem competitiva com outras barbearias que ainda fazem esse trabalho manualmente e que apresentam constantes erros operacionais por isso.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral deste Projeto de Trabalho de Graduação é desenvolver um aplicativo em que o cliente possa consultar os horários disponíveis e agendar um serviço em uma barbearia de forma rápida e simples. Essa ferramenta será de grande ajuda para retomar e aumentar a alta demanda que a barbearia tem, atendendo mais pessoas de modo mais informatizado.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, serão estabelecidos os seguintes objetivos específicos:

1. Utilizar os conhecimentos adquiridos nas disciplinas de Engenharia de Software, no planejamento e no desenvolvimento do aplicativo.

2. Desenvolver um aplicativo a onde o barbeiro possa gerenciar sua agenda, seus serviços e disponibilizá-los no aplicativo para seus clientes.
3. Utilizar a ferramenta app inventor para desenvolver o aplicativo para o gerenciamento do agendamento de uma barbearia.
4. Testar e avaliar o aplicativo, analisando as consequências positivas da implantação.

2. REFERENCIAL TEÓRICO

2.1 BELEZA E BARBEARIA

Não há muitos relatos a respeito de como surgiram as barbearias, porém sabe-se que foi há muito tempo, e os homens cada vez mais estão preocupados com a sua aparência e cuidado de seus cabelos e barba. Segundo Sebrae (2019), o mercado da beleza vem passando por transformações em todo o mundo e uma das que mais se destacam é o interesse do público masculino por esse tipo de atendimento. "É um mercado que apresenta uma grande maturidade, porém, ainda oferece enorme espaço para profissionalização e melhoria da qualidade, tanto dos serviços prestados quanto do atendimento pessoal ao cliente. Existe uma grande quantidade de barbearias que trabalha de forma precária, em instalações inadequadas, com profissionais mal preparados no mister em si e sem conhecimentos mínimos sobre a melhor forma de administrar um pequeno negócio"(SEBRAE, 2008).

Segundo o Sebrae (2019), cerca de 342 mil salões de beleza são formalmente registrados no Brasil e destes, 286 mil são microempreendedores individuais - MEI. Conforme o levantamento feito pelo Sebrae (2019), a partir de dados da Pnad - Pesquisa Nacional por Amostra de Domicílio, do IBGE, de um total de 6,6 milhões de empreendedores, mais de 1 milhão é de empreendedores da beleza. E o nosso intuito é documentar como uma barbearia pode melhorar a gestão

de gerenciamento de agendamento utilizando ferramentas inovadoras. Pois a grande demanda de ferramentas e soluções gera muita competição e aquele que adquire a tecnologia na prestação de serviços ou entrega um serviço com qualidade e usando a tecnologia ao seu favor com ferramenta que auxilia, otimiza ou até mesmo automatiza serviço de forma prática e ágil, possibilitando um aumento no número de clientes. Pois como a população está cada vez mais conectada à internet e aplicativos, buscando cada vez mais a facilidade na sua rotina com a automatização dos serviços. “Com toda a competitividade do mercado de trabalho presente na vida atual, cada detalhe importa para se destacar, então a preocupação com a autoimagem aumenta e se torna uma forma de promover um bom marketing pessoal. Além disso, de acordo com levantamento de 2016 do Instituto Qualibest para ABIHPEC, mais de oito entre dez homens afirmam que “foi o tempo em que só as mulheres se preocupavam com a beleza” (SEBRAE, 2019).

2.2 IMPORTÂNCIA DA GESTÃO NOS NEGÓCIOS

De acordo com o Sebrae (2008), um Software no estabelecimento, será de grande utilidade para a automatização dos serviços, mesmo que o negócio seja simples, os dispositivos aprimoram o atendimento e a experiência do cliente com seus serviços, além de ser uma ferramenta de divulgação de propaganda para o estabelecimento podendo mostrar cortes de cabelos, barbas feitas e outros serviços disponíveis para os clientes, gerando um contato com o ambiente onde será efetuado o atendimento podendo também demonstrar a localização do salão. Acaba sendo muito interessante e útil utilizar um aplicativo que além dos clientes agendarem sozinho no seu celular de forma rápida, o barbeiro cuidar de sua gestão de gerenciamento de agendamento para atendimento de forma inteligente e organizada. Se preocupando e focando mais com o serviço prestado e seu cliente, pois ficaria mais simples acompanhar a lista de clientes agendado para ser atendido por ele no dia sem precisar interromper um atendimento para marcar um agendamento aumentando a capacidade de atender mais clientes.

2.3 GESTÃO DE AGENDAMENTOS

A gestão de agendamentos, basicamente é a capacidade de organizar a sua agenda de serviços e procedimentos de forma clara e objetiva, visando a otimização dos processos de agendamento, economia de tempo e integridade da informação, focalizando na qualidade do serviço e/ou produto e na satisfação total do cliente. Nos pequenos negócios, a gestão de agendamentos passou de um diferencial para algo essencial em qualquer negócio que envolva atendimento ao público, as pessoas necessitam do serviço mais rápido, com a entrega mais rápida e com o preço mais rentável, por isso, os empreendedores necessitam se adaptar com essas mudanças para se manterem competitivos no mercado, a grande alternativa atual são sistemas de agendamentos, que automatizam a famosa agenda, a caneta e o trabalho manual, além de garantirem integridade total da informação e deixando o primeiro atendimento ao cliente de forma automática, eliminando trabalho manual desnecessário.

De acordo com o Sebrae (2015), métodos utilizados para automatizar os processos (compras, vendas, controle de estoques, cadastro de clientes, consultas.etc.) fazem parte da automação de serviços, fazendo com que esses processos manuais se tornem mais mecanizados, rápidos e objetivos. Um *software* de automação comercial, ou aplicativo nesse caso, integra processos comerciais que vão desde procedimentos operacionais de gestão de estoque e aquisições até a venda e geração de notas fiscais, com geração de informações financeiras que serão essenciais para a gestão da empresa.

Atualmente na barbearia que vai ser desenvolvido o aplicativo, a gestão é feita manualmente e exige alguns processos realizados pelo barbeiro para o atendimento acontecer. O primeiro passo é realizado pelo cliente, o agendamento manual feito por WhatsApp ou Telefone, o segundo passo é feito pelo barbeiro, checar a disponibilidade de horário e confirmar se pode ou não realizar o atendimento, o terceiro passo é aguardar o *feedback* do cliente e a confirmação de horário e disponibilidade para comparecer ao estabelecimento por ambas as partes. Muitas vezes, ocorrem falhas de comunicação que atrapalham o processo

operacional do barbeiro, como por exemplo, um imprevisto acontece e, como consequência, o cliente não comparece na hora agendada e aquele horário fica vazio, diminuindo o número de atendimentos e o valor do tempo do barbeiro, o que consequentemente, diminui a agregação de valor ao serviço final. O aplicativo também deve auxiliar nas situações que geram improdutividade e perda de lucro, como por exemplo, a situação citada acima. Não é possível, por exemplo, prever os imprevistos do cliente, mas é possível fazer uma ferramenta que possibilite um cancelamento em tempo real, transmitindo essa informação para ambas as partes e impedindo inconvenientes.

3. MATERIAL E MÉTODOS

Neste capítulo, apresenta-se o material utilizado e os procedimentos realizados no desenvolvimento da pesquisa. O material é listado abaixo e os procedimentos são descritos nos itens seguintes.

A fase de levantamento de dados envolveu coleta de material bibliográfico e de dados industriais, uma fonte em destaque, foi o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e fontes do Google Acadêmico, que exploram muito bem a área da automatização comercial e desenvolvimento tecnológico de pequenos negócios.

Em busca do conhecimento existente sobre o crescimento da automatização, foram consultados os trabalhos desenvolvidos anteriormente que envolveram a esse tema em sua composição. No Brasil, as fontes que exploram com riqueza de detalhes a área tecnológica, são escassas. Mesmo assim, foi possível levantar um considerável volume de informações, que serviu para fornecer uma visão geral do conhecimento existente sobre o assunto.

Para formatação e desenvolvimento do texto, utilizamos o Google Drive e o Word 2016, formatando conforme pede a Associação Brasileira de Normas Técnicas (ABNT).

3.1 TECNOLOGIAS E FERRAMENTAS UTILIZADAS

Para o desenvolvimento do trabalho, foi necessário a utilização de diversas ferramentas. Entre elas:

- A. AppInventor: O App Inventor é um ambiente de programação, desenvolvido no MIT. De acordo com MIT App Inventor (2018) a proposta dessa ferramenta é que permita às pessoas que tenham pouco conhecimento de técnico de programação, a desenvolver aplicativos para smartphones e tablet, permitindo que isso seja feito de forma online sem precisar fazer *download* ou instalar, apenas necessitando que o usuário realize um cadastro com uma conta pessoal. Essa ferramenta funciona por meio da programação em blocos, podendo incluir estruturas de controle, funções lógicas e matemáticas;
- B. Firebase: É uma plataforma para desenvolvimentos de aplicativos para dispositivos móveis e Web desenvolvida inicialmente pela Firebase Inc em 2011 e adquirido pela google em 2014.

Entre suas tecnologias, as principais:

- Realtime Database

O Firebase Realtime Database é um banco de dados NoSQL hospedado na nuvem. Com ele, você armazena e sincroniza dados entre os seus usuários em tempo real.

- Hosting

Com o Firebase hosting, você implanta facilmente um app da Web de uma única página, uma página de destino de aplicativos para dispositivos móveis ou um Progressive Web App.

- Cloud Firestore

O Cloud Firestore é um banco de dados de documentos NoSQL que permite armazenar, sincronizar e consultar dados facilmente para seus aplicativos móveis e da Web - em escala global.

- Cloud Functions

Crie funções acionadas por produtos do Firebase, como alterações de dados no Realtime Database, inscrições de novos usuários pelo Auth e eventos de conversão no Analytics.

- Monitoramento de desempenho

Receba insights sobre o desempenho do seu app do ponto de vista dos usuários, com rastreamento de desempenho automático e personalizado.

- Authentication

O objetivo do Firebase Authentication é facilitar o desenvolvimento de um sistema de autenticação seguro e melhorar a experiência de login e ambientação para os usuários finais. Ele oferece uma solução de identidade

completa, compatível com contas de e-mail/senha, autenticação por telefone, login do Google, Twitter, Facebook, GitHub e outros.

- Cloud Storage

O Cloud Storage foi criado para ajudar você a armazenar e oferecer conteúdo gerado pelo usuário, como fotos ou vídeos, com facilidade e rapidez.

- Google AdWords

Impulsione o número de instalações, receba insights aprofundados sobre conversões de anúncios e exiba campanhas de anúncios segmentados usando os públicos-alvo do Google Analytics for Firebase para interagir com a sua base de usuários.

- AdMob

A AdMob by Google é uma plataforma de publicidade móvel que pode ser usada para gerar receita com seu aplicativo. O uso da AdMob com o Google Analytics para Firebase fornece recursos adicionais de dados e recursos de análise de aplicativos. O Firebase integra-se à AdMob sem exigir alterações na configuração existente.

- Cloud Messaging

O Firebase Cloud Messaging (FCM) oferece uma conexão confiável e com baixo consumo de bateria entre servidor e dispositivos para enviar e receber mensagens e notificações no Android, no iOS e na Web sem custo nenhum.

- Google Analytics

O Google Analytics para Firebase oferece geração de relatórios gratuita e ilimitada para até 500 eventos diferentes. O SDK captura automaticamente determinados eventos-chave e propriedades do usuário, e você pode definir eventos personalizados e avaliar itens relevantes para o seu negócio.

De acordo com Tekzoom (2019), Kodular é uma plataforma de criação de aplicativos através do uso de blocos lógicos de programação. É uma evolução do MakeRoid, uma plataforma baseada no AppInventor, criada no MIT (Massachusetts Institute of Technology).

3.2 ENGENHARIA DE SOFTWARE

O mundo não poderia existir sem o software. Pois ele é utilizado em todas as áreas, sendo assim a engenharia de software é essencial para o funcionamento da sociedade. Como a criação de um software não tem limites para o potencial do software, surge a necessidade da Engenharia de software para compreensão do projeto porque pode se tornar extremamente complexo o modo de entender o desenvolvimento. Um software não é apenas o programa, mas também todos os dados de documentação e configuração associados, necessários para que o programa opere corretamente. (Sommerville, 2013, p, 2).

3.3 REQUISITOS FUNCIONAIS DO SISTEMA

3.3.1 Cadastro de usuários barbeiros

O cadastro de usuários Barbeiros do sistema contemplará as seguintes funções:

- a) deverá armazenar os dados no sistema, por meio de uma login e Senha do usuário;
- b) o código do usuário terá os seguintes atributos: nome, senha e perfil de acesso;
- c) nesse acesso do perfil, denominada Cod_Barb, o sistema deverá obter os seguintes métodos: adicionar, alterar e consultar. Serviços, horário e data de atendimento e será visualizado na tela as informações requeridas;
- d) restrições: o sistema libera opção da hora de atendimento assim que escolhida e for liberada para os clientes agendar, não permitirá a exclusão

da reserva solicitada pelo cliente e permanecerá, dessa forma, no banco de dados.

3.3.2 Cadastro de clientes

O cadastro de clientes do sistema contemplará as seguintes funções:

- a) Deverá cadastrar os clientes que desejam solicitar serviços coletando dados pessoais como o nome do barbeiro escolhido, serviço solicitado, data e horário, que utilizará no sistema, por meio das informações será liberado o agendamento do cliente;
- b) o código do cliente deverá ter os seguintes atributos: nome completo do cliente, telefone e observação;
- c) nessa mesma chave primária, denominado Cod_Clin, o sistema deverá obter os seguintes métodos: incluir, alterar, consultar e deletar, e será visualizado na tela as informações requeridas;
- d) restrições: o sistema não permitirá o cliente libera data e horário, só será liberado pelo barbeiro. Só terá a opção de escolher horários disponíveis pelo barbeiro.

3.3.3 Agendamentos

Os agendamentos serão feitos das seguintes formas:

- a) O sistema deverá consultar os horários e dias que foram disponibilizados e trazer a lista para que o cliente escolha o horário e o dia de sua preferência;
- b) Dentro deste procedimento utilizará os seguintes atributos: data, horário, serviço disponível;
- c) Os métodos utilizados serão os seguintes: consultar, agendar, alterar, cancelar;

- d) Os horários disponíveis serão liberados conforme o horário de funcionamento:

3.4 REQUISITOS NÃO-FUNCIONAIS DO SISTEMA

Os requisitos não-funcionais do sistema são relacionados às propriedades emergentes do sistema, tais como a confiabilidade, a segurança, a eficiência, a usabilidade, a portabilidade, o desempenho entre outros do software. (SOMMERVILLE, 2013, p, 216).

3.4.1 Confiabilidade

A confiabilidade de um sistema de gestão de agendamento será definida como a habilidade do sistema executar sua função dentro de limites e condições operacionais.

Os dados do agendamento, por meio de um aplicativo conectado a um *modem wi-fi* ou a dados móveis da operadora do aparelho para transmissão dos dados ao banco de dados.

3.4.2 Tolerância a falhas

O modelo de uma falha em um dispositivo pode mudar ao longo do seu ciclo de vida. Pode permanecer inalterado, diminuir ou mesmo aumentar.

- a) o sistema deverá ser capaz de salvar todos os dados enviados pelos clientes e o barbeiro;
- b) o sistema deverá prover atenção especial a respeito de datas e horários, resiliente a possíveis erros de configuração na data e hora do sistema, prevenindo que haja problemas com os dados.

3.4.3 Portabilidade

A portabilidade de um sistema é a capacidade de ser compilado ou executado em diferentes arquiteturas.

- a) o sistema será desenvolvido em blocos que usa a linguagem Java e XML;
- b) o sistema será desenvolvido com *software* livre de restrições e licenças comerciais;
- c) o banco de dados que deverá ser implementado será o Firebase;
- d) o sistema será disponibilizado por meio de acesso via aplicativo, assim, utilizaremos programação em Blocos para construção do aplicativo e poderá ser acessado por tablets e celulares, de forma responsiva.

3.4.4 Hardware

Os requisitos mínimos de hardware para o sistema são:

- a) Um aparelho celular ou tablet com no mínimo Android 4.4;
- b) um sistema de agendamento autônomo;
- c) um modem *wi-fi* ou dados moveis.

3.5 VISÃO DE CASO DE USO

Abaixo, temos algumas especificações relacionadas à engenharia de software

3.5.1 Conceito de Caso de Uso

Segundo Ribeiro (2018), o Diagrama de Casos de Uso serve para auxiliar a comunicação entre o desenvolvedor e o cliente, ou seja, ele serve para trazer informações dos processos e atividade para a modelagem e estruturação dos dados.

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em

outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz. (RIBEIRO, 2018).

É importante frisar que o Caso de Uso não descreve como o *software* deverá ser construído, mas sim como ele deverá se comportar quando estiver pronto.

3.5.2 Operação do Sistema

Os Clientes que estão insatisfeitos com sua aparência ou estética, vão em busca de procurar um profissional qualificado para fazer sua barba ou corte de cabelo, em barbearia que possa solucionar e agradar seus gostos e necessidades estéticas com sucesso. Estes clientes estão à procura de facilidade e agilidade para facilitar ao máximo o contato e o agendamento para efetuar o serviço. Intenção desse aplicativo é facilitar o agendamento que possa ser feito com antecedência por aplicativo, dando opções de cancelamento pelo cliente e a disponibilização de datas e horários pelo barbeiro para o agendamento do serviço, e requerido do cliente os seus dados no sistema como nome, telefone e observação para estar efetuando o agendamento. No início do sistema será exibido uma tela de menu aonde poderá efetuar o agendamento, consulta de agendamento e entrar no Facebook da Barbearia. Após a entrada no agendamento, abrir-se-á a tela do programa que disponibilizará ao cliente as opções de serviço e agendamento. Já entrando na consulta de agendamento, exibirá a tela para o serviço e o agendamento escolhido pelo cliente e poderá cancelar o serviço se for necessário pelo o usuário. Já para o dono o barbeiro a entrada na tela administrador aonde será necessário um login com o seu nome e senha do barbeiro. Vai abrir uma nova tela aonde será possível disponibilizar as datas e os horários de atendimento e consultar os horários e as datas agendadas e os nomes e contatos dos clientes, poderá adicionar serviços e os valores. Os dados serão guardados no banco de dados.

3.5.3 Definição de Atores

A definição de ator nos modelos de caso de uso é todo elemento que interage com o sistema, disparando uma ação. Estes elementos podem ser pessoas ou outros subsistemas que podem, ou não, fazer parte do sistema. Aquele que não está explicitamente sendo parte do sistema precisa estar inserido no contexto do processo.

Sendo assim a gestão de agendamentos de uma barbearia contem os seguintes atores:

a) Ator Barbeiro

O ator Barbeiro é aquele que executa o serviço, faz o corte de cabelo e barba do cliente, e disponibiliza conforme a sua necessidade o agendamento e o serviço as datas e os horários.

b) Ator Cliente

O ator cliente é a pessoa que se submete ao serviço; faz o agendamento na barbearia e a cancela o atendimento conforme a sua necessidade.

3.5.4 Tabelas de Caso de Uso

As tabelas de caso de uso referenciam e explicam textualmente as ações que são representadas por terminologia verbal disparada pelo ator. O Gerenciamento de Consumo de Energia apresenta as seguintes tabelas:

Nº	CASO DE USO	DESCRIÇÃO
1	IncluirBarb	Método de classe que faz o cadastro do barbeiro no banco de dados.
2	ConsultarBarb	Método de classe que faz a consulta do barbeiro no banco de dados.

3	AlterarBarb	Método de classe que faz a alteração do barbeiro no banco de dados.
4	ExcluirBarb	Método de classe que faz a exclusão do barbeiro no banco de dados.
5	GerenciarAgend	O barbeiro é quem gerencia a agenda podendo incluir novos tipos de serviços e disponibilizar os dias e horários de atendimento.

Quadro 1: Caso de uso do ator usuário máster (Elaboração Própria).

Nº	CASO DE USO	DESCRIÇÃO
1	IncluirClient	Método de classe que faz o cadastro do cliente no banco de dados.
2	ConsultarClient	Método de classe que faz a consulta do cliente no banco de dados.
3	AlterarClient	Método de classe que faz a alteração do cliente no banco de dados.
4	ExcluirClient	Método de classe que faz a exclusão do cliente no banco de dados.
5	AgendarServic	O cliente faz o agendamento do serviço podendo também cancelar o agendamento se necessário.

Quadro 2: Caso de uso do ator usuário operador (Elaboração Própria).

3.5.5 Modelos de Caso de Uso

Modelos de caso de uso trazem a visualização gráfica das ações que são chamadas pelos atores envolvidos no processo. O Gerenciamento Clientes e serviços apresenta os seguintes atores:

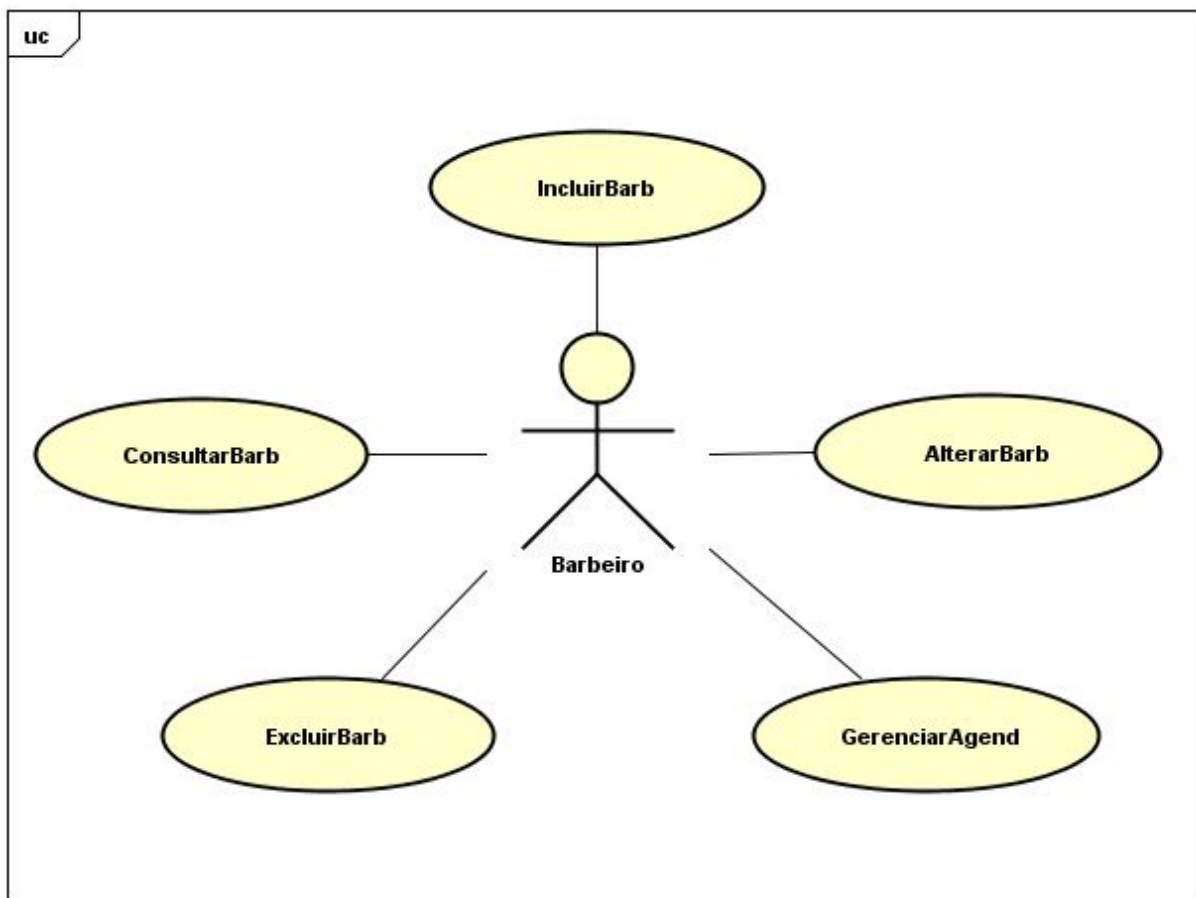


Figura 1: Modelo de Caso de uso do Ator Barbeiro (Elaboração Própria).

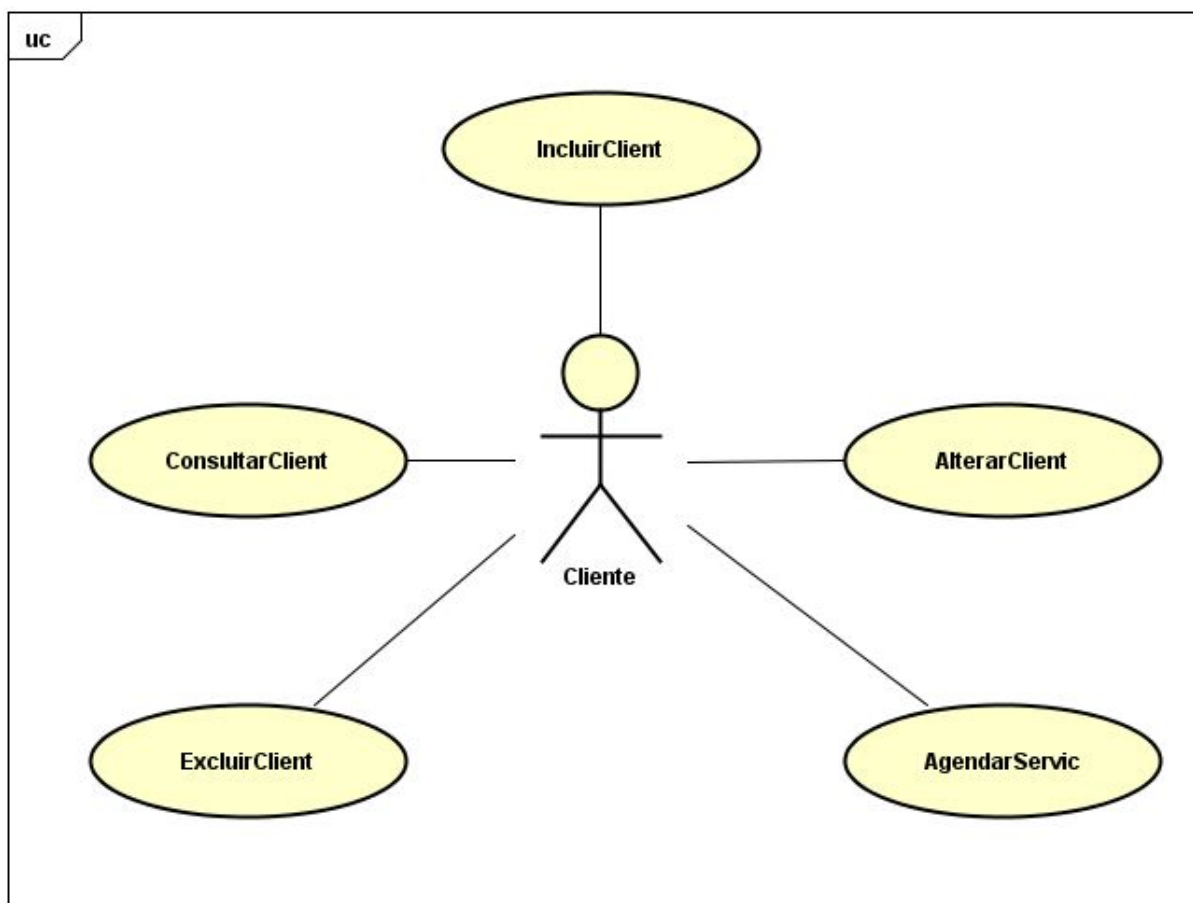


Figura 2: Modelo de Caso de uso do Ator Cliente (Elaboração Própria).

3.6 MODELOS DE SEQUÊNCIA

O diagrama de sequência demonstra de forma clara e objetiva a interação entre as classes em sua linha de vida", ou seja, na execução do processo ou no dia a dia dos usuários descrevendo a ordem exatas em que os atores disparam as ações. É possível também entender as possibilidades que podem ocorrer dentro do sistema fazendo a previsão de possíveis erros além de ilustrar as interações dos objetos que pertencem ao caso de uso porém este diagrama não demonstra a relações das classes é voltado mais para a parte de execução do sistema ou seja a parte pratica do software.

O Diagrama de Sequências não surge do nada e sim de uma modelagem feita a partir dos casos de uso, com o auxílio das classes já identificadas num modelo de classes. No caso de uso, estabelecemos a ordem das funcionalidades, sem nos preocuparmos com a implementação. Ao modelarmos o diagrama de sequências, representaremos por mensagens cada item descrito nos cenários principal e alternativos. (CRISTINA, 2009).

Tudo começa com o ator que faz o disparo da ação então é enviado uma mensagem para um receptor que envia uma resposta de volta ou envia uma outra solicitação para outro receptor e assim segue em sequência até se encerrar aquela determinada atividade. As mensagens são a representação de como o sistema vai se comunicar através de solicitações e respostas, vale salientar que os atores e receptores podem ser pessoas ou parte do sistema que age de forma automática conforme a solicitação, porém, quando o responsável por disparar a ação não é uma pessoa este é chamado de modulo . Um bom exemplo são os aplicativos de troca de mensagem onde você envia a informação para a o banco de dados do aplicativo então o sistema envia para pessoa que você destinou e em alguns casos quando a pessoa recebe a mensagem é enviado uma resposta que a mensagem foi entregue ou até mesmo lida. Cada ação disparada deve possuir um índice incluindo as respostas como subtítulos deste índice, após a representação deste fluxo do sistema

deve-se estudar as possibilidades de erros nesta comunicação sequencial listando-as e colocando índices que serão o subcapítulo referentes a ação onde há possibilidade do erro evidenciado. Em outras palavras o diagrama de sequência é como o roteiro de uma peça de teatro onde é necessário estudar o desfecho da história para se garantir uma coerência no resultado final.

3.6.1 Diagrama de Sequência

a) Agenda

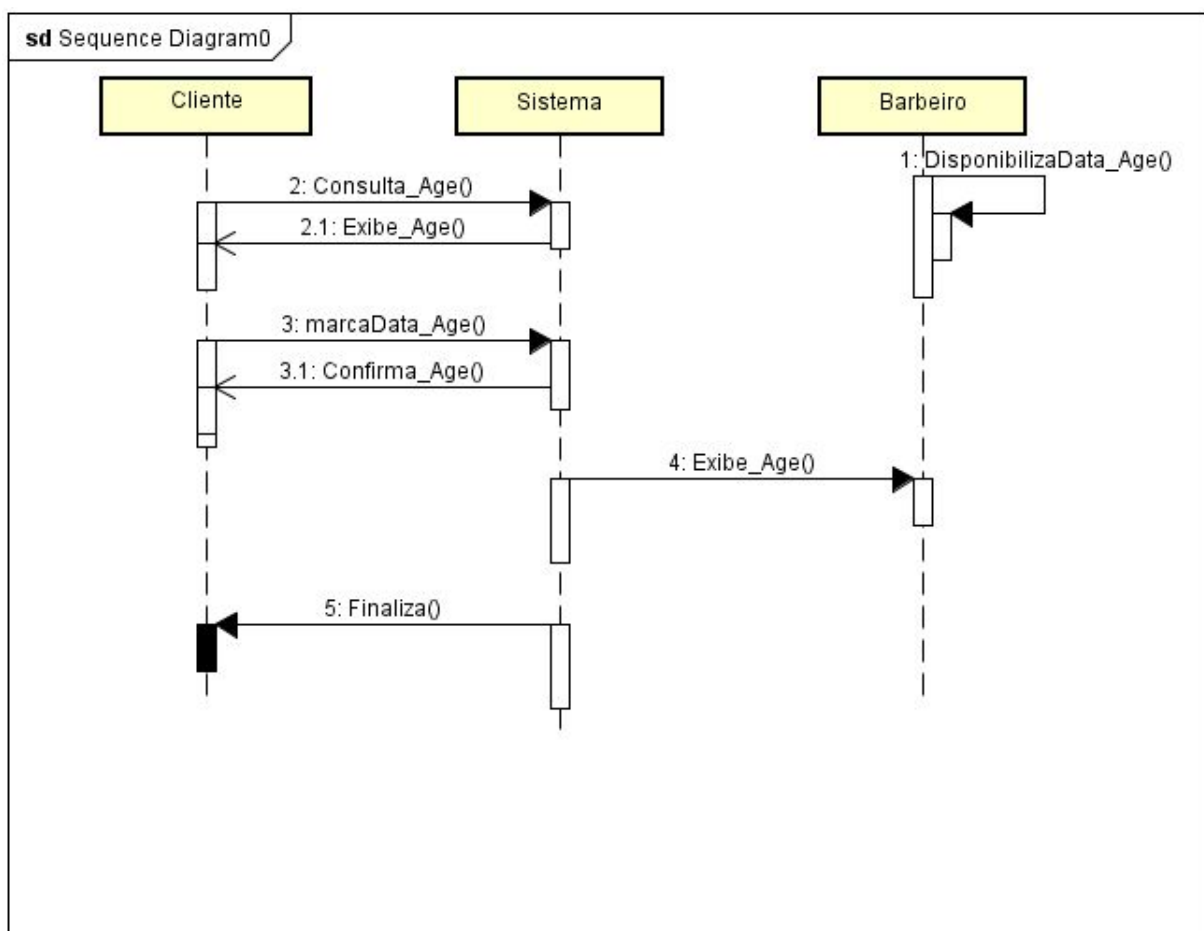


Figura 3: Diagrama de sequência do relatório da demanda ativa (Elaboração Própria).

Curso Normal

- 1 O barbeiro disponibiliza data.

- 2 O cliente solicita a agenda.
- 2.1 O sistema exibe agenda.
- 3 O cliente marca a data.
- 3.1 O cliente confirma a data escolhida.
- 4 O sistema exibe a agenda para barbeiro.
- 5 O sistema finaliza.

Curso Alternativo

- 1.1 Sem conexão com a internet.

3.7 VISÃO LÓGICA: NÍVEL DE ANÁLISE

É muito importante para o processo de produção de *software* ter sua visão lógica de negócios estipulada de forma clara e concisa, para que o melhor entendimento das interoperações venha a solucionar dúvidas que pairam sobre a modelagem dos processos dentro do ambiente de uso do aplicativo.

Diferentemente dos modelos anteriores abordados, o diagrama de classes é um diagrama que inclui melhores detalhes individuais de cada objeto no sistema e nos remetem a uma visão de modelo de análise bastante compreensível. As classes são dispostas no diagrama em forma de retângulos divididos em 3 seções e são interligados por linhas que identificam seu relacionamento correspondente, uns com os outros.

Com base nesses em tais diagramas, também podemos identificar vários “erros” cometidos no processo de elaboração do *software*, pois muitas vezes deixamos de entender a visão de negócios da forma como ela deve ser entendida. Motivo este pelo qual são muito bem pagos os profissionais da área de modelagem na engenharia de *software*: não é fácil identificar a visão de negócios.

Conforme ratificado por Sommerville (2003), identificar os objetos do sistema e as relações entre eles, é a área mais difícil da análise e projeto na orientação a objetos.

Apesar de ser reconhecido como a parte que mais dificulta o trabalho para o profissional, podemos considerar que:

A comunicação da mensagem (*messaging*) é importante para a implementação de um sistema orientado a objeto, mas ela não precisa ser considerada detalhadamente durante a análise de requisitos. De fato, nossa única preocupação nessa fase é usar o conceito para ajudar a determinar as possíveis operações para um objeto específico. (PRESSMAN, 1995, p. 330).

Os tipos de mensagens trocadas entre as classes denotam uma ação e um tipo de relacionamento, bem como sua cardinalidade.

3.8 RELACIONAMENTOS PRINCIPAIS AS CLASSES

Relacionamentos estruturais entre classes precisam ser criteriosamente definidos durante o projeto do software. São obtidos a partir da análise dos diagramas de colaboração, e é um dos principais diagramas da UML, que define o esqueleto do sistema.

Há cinco tipos de relacionamentos entre classes:

1. associação: Relacionamentos estruturais na visão de negócios, determinando como cada classe afeta o fluxo do sistema, sem necessariamente afetar um ao outro.
2. herança: Informa que o relacionamento dos objetos possui características comuns a ponto de uns herdarem os atributos das classes pai.
3. dependência: Relacionamento em que um objeto depende da existência de outro objeto, que têm suas especificações alteradas caso o objeto ao qual depende for alterado. É considerada uma classe bastante fraca e difícil de figurar um projeto – normalmente esta classe se encontra muitas vezes eliminada por um atributo ou método de outra classe.
4. composição: Relacionamento em que certas classes são compostas por outras, existindo estas quando se verificar a existência de outra classe que compõe o objeto. Quando aquela classe for “destruída”, imediatamente se destruirá as classes que ela compõe.
5. agregação: Relacionamento em que define que uma classe só terá valor se não interagir sozinha, mas unicamente junto com outra classe. Este é o melhor método para diferenciar a agregação da composição, que são facilmente confundidas nos projetos. Quando uma classe não tem utilidade sozinha, dizemos então que aquela classe deve ser agregada à outra.

Junto à simbologia dos relacionamentos, é especificado o termo de ação entre

as classes. Deve-se observar também, no diagrama de classes, que quando um método está explicitamente ligando duas classes como uma ação, e é listado originalmente na classe, este método desaparecerá da classe daquele objeto, deixando apenas que a ação permaneça na ligação de relacionamento entre as duas classes.

A cardinalidade denota o número de objetos que se relacionam aos objetos que estão ligados a ele: a notação de (1) significa que apenas 1 classe interage do lado em que está posicionado o número; de forma que, no lugar que tivermos (1..*) significará que 1 classe ou mais objetos desta classe interagirão com outras classes.

Por exemplo, a classe Máster temos a designação (1) pois neste contexto somente pode-se ter um único usuário Máster, e não dois ou mais; porém, um único usuário Máster cadastra 1 ou mais usuários Administrador (1..*) sendo o asterisco a representação de um número indeterminado.

3.9 DIAGRAMA DE CLASSES DO PROJETO (UML)

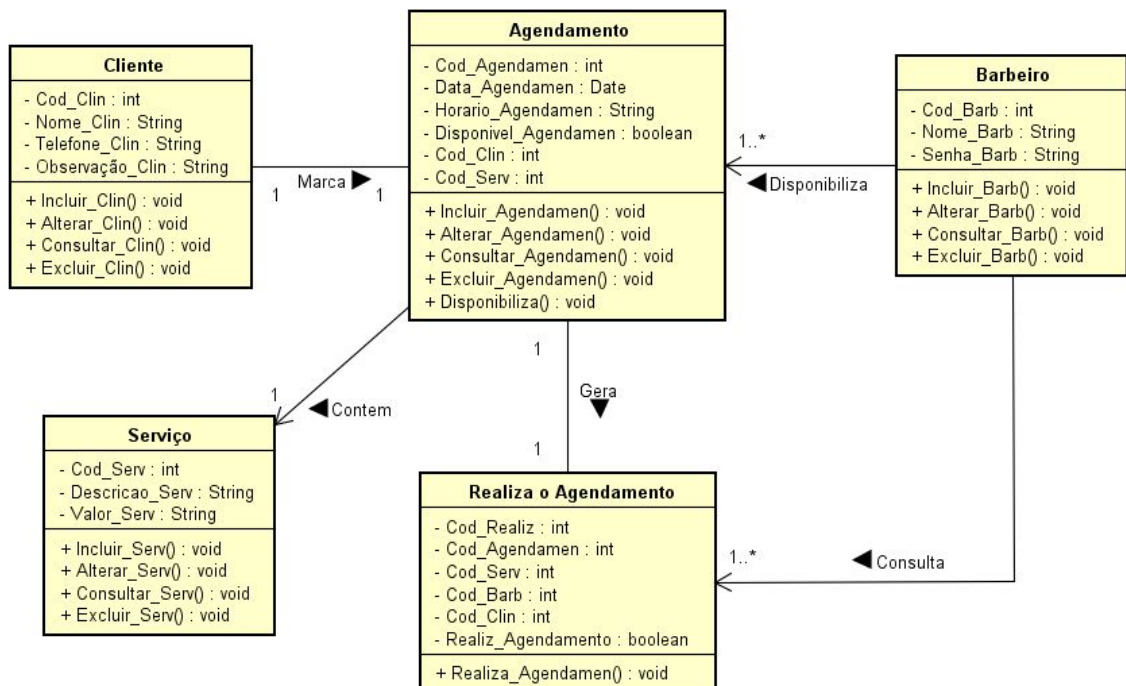







Figura 4: Diagrama de Classes (Elaboração Própria).

3.10 SIMBOLOGIA DOS RELACIONAMENTOS ENTRE CLASSES

SIMBOLOGIA DE RELACIONAMENTOS DAS CLASSES	
	Associação
	Herança
	Dependência
	Composição
	Agregação

Quadro 5: Simbologia de relacionamentos das classes (Elaboração Própria)

3.11 MODELO ENTIDADE RELACIONAMENTO (MER)

Quando se inicia o desenvolvimento de um novo sistema, ou mesmo de uma nova funcionalidade para um sistema existente, um dos primeiros passos a ser executado é o estudo e levantamento dos requisitos necessários para a construção do produto. Durante essa análise, identifica-se as principais partes e objetos envolvidos, suas possíveis ações e responsabilidades, suas características e como elas interagem entre si.

A partir das informações obtidas, pode-se desenvolver um modelo conceitual que será utilizado para orientar o desenvolvimento propriamente dito, fornecendo informações sobre os aspectos relacionados sobre domínio do projeto em questão.

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais.

Observação: Nem sempre criaremos modelos para um sistema completo, pois isso poderia resultar em um modelo muito extenso e difícil de interpretar. Dependendo da magnitude do que estaremos desenvolvendo, podemos criar modelos apenas para uma parte do sistema, um módulo, ou mesmo uma funcionalidade. Imagine, por exemplo, um sistema ERP de grande porte que contemple vendas, finanças, recursos humanos, etc.

Várias entidades estão presentes em mais de uma parte do sistema, mas não seria muito interessante, e provavelmente nem mesmo necessário, criar um único modelo para todo o sistema, por isso pode-se dividir a modelagem em várias partes menores.

3.11.1 Entidades

Os objetos ou partes envolvidas um domínio, também chamados de entidades, podem ser classificados como físicos ou lógicos, de acordo com sua existência no mundo real. Entidades físicas: são aquelas realmente tangíveis, existentes e visíveis no mundo real, como um cliente (uma pessoa, uma empresa) ou um produto (um carro, um computador, uma roupa). Já as entidades lógicas são aquelas que existem geralmente em decorrência da interação entre ou com entidades físicas, que fazem sentido dentro de um certo domínio de negócios, mas que no mundo externo/real não são objetos físicos (que ocupam lugar no espaço). São exemplos disso uma venda ou uma classificação de um objeto (modelo, espécie, função de um usuário do sistema).

As entidades são nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio. Exemplos práticos de entidades comuns em vários sistemas são Cliente, Produto, Venda, Turma, Função, entre outros.

Podemos classificar as entidades segundo o motivo de sua existência:

- a) entidades fortes: são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir. Em um sistema de vendas, a entidade produto, por exemplo, independe de quaisquer outras para existir;
- b) entidades fracas: ao contrário das entidades fortes, as fracas são aquelas que dependem de outras entidades para existirem, pois individualmente elas não fazem sentido. Mantendo o exemplo, a entidade venda depende da entidade produto, pois uma venda sem itens não tem sentido;
- c) entidades associativas: é uma entidade que originalmente era um relacionamento e que posteriormente foi transformada em entidade associativa pelo motivo de existir um relacionamento que é gerado a partir do relacionamento original, e como não é possível ter um relacionamento

ligado a outro relacionamento, então surge a entidade associativa. Exemplo: Entidade Médico - Relacionamento Consulta - Entidade Paciente. Originalmente consulta é um relacionamento. Então surge a necessidade de criar uma entidade chamada Receita Médica, e esta entidade também seria ligada ao relacionamento Consulta. Mas percebe-se que a Receita Médica só faz sentido se for relacionada a Consulta, pois não pode existir receita médica somente relacionada ao paciente ou ao médico (um relacionamento ternário permite esta interpretação). Como relacionar uma entidade a um relacionamento? Seria uma entidade com uma relação ligada a outra relação. Então transforma-se Consulta em Entidade Associativa. Ela continua sendo uma relação para Médico e Paciente, mas para Receita Médica ela se comporta como Entidade. Esta Entidade tem um relacionamento com Receita médica, deixando claro que a receita médica se relaciona com a consulta, e para a consulta existir se faz necessário um relacionamento entre médico e paciente.

3.11.2 Relacionamentos

Uma vez que as entidades são identificadas, deve-se então definir como se dá o relacionamento entre elas. De acordo com a quantidade de objetos envolvidos em cada lado do relacionamento, podemos classifica-los de três formas:

- a) relacionamento 1..1 (um para um): cada uma das duas entidades envolvidas referencia obrigatoriamente apenas uma unidade da outra. Por exemplo, em um banco de dados de currículos, cada usuário cadastrado pode possuir apenas um currículo na base, ao mesmo tempo em que cada currículo só pertence a um único usuário cadastrado;
- b) relacionamento 1..n ou 1..* (um para muitos): uma das entidades envolvidas pode referenciar várias unidades da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada uma unidade da outra entidade. Por exemplo, em um sistema de plano de saúde, um usuário pode ter vários dependentes, mas cada dependente só

pode estar ligado a um usuário principal. Note que temos apenas duas entidades envolvidas: usuário e dependente. O que muda é a quantidade de unidades/exemplares envolvidas de cada lado;

- c) relacionamento $n..n$ ou $*..*$ (muitos para muitos): neste tipo de relacionamento cada entidade, de ambos os lados, pode referenciar múltiplas unidades da outra. Por exemplo, em um sistema de biblioteca, um título pode ser escrito por vários autores, ao mesmo tempo em que um autor pode escrever vários títulos. Assim, um objeto do tipo autor pode referenciar múltiplos objetos do tipo título, e vice-versa.

Os relacionamentos em geral são nomeados com verbos ou expressões que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra. Essa nomenclatura pode variar de acordo com a direção em que se lê o relacionamento. Por exemplo: um autor escreve vários livros, enquanto um livro é escrito por vários autores.

3.11.3 Atributos

Atributos são as características que descrevem cada entidade dentro do domínio. Por exemplo, um cliente possui nome, endereço e telefone. Durante a análise de requisitos, são identificados os atributos relevantes de cada entidade naquele contexto, de forma a manter o modelo o mais simples possível e consequentemente armazenar apenas as informações que serão úteis futuramente. Uma pessoa possui atributos pessoais como cor dos olhos, altura e peso, mas para um sistema que funcionará em um supermercado, por exemplo, estas informações dificilmente serão relevantes.

Os atributos podem ser classificados quanto à sua função da seguinte forma:

- a) descritivos: representam característica intrínsecas de uma entidade, tais como nome ou cor;
- b) nominativos: além de serem também descritivos, estes têm a função de definir e identificar um objeto. Nome, código, número são exemplos de atributos nominativos;

- c) referenciais: representam a ligação de uma entidade com outra em um relacionamento. Por exemplo, uma venda possui o CPF do cliente, que a relaciona com a entidade cliente.

Quanto à sua estrutura, podemos ainda classificá-los como:

- a) simples: um único atributo define uma característica da entidade.
Exemplos: nome, peso;
- b) compostos: para definir uma informação da entidade, são usados vários atributos. Por exemplo, o endereço pode ser composto por rua, número, bairro, etc.


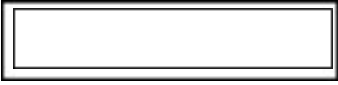
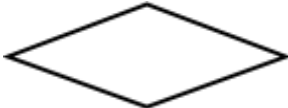





Alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir. Em um cadastro de clientes, por exemplo, esse atributo poderia ser o CPF. A este podemos chamar de Chave Primária.

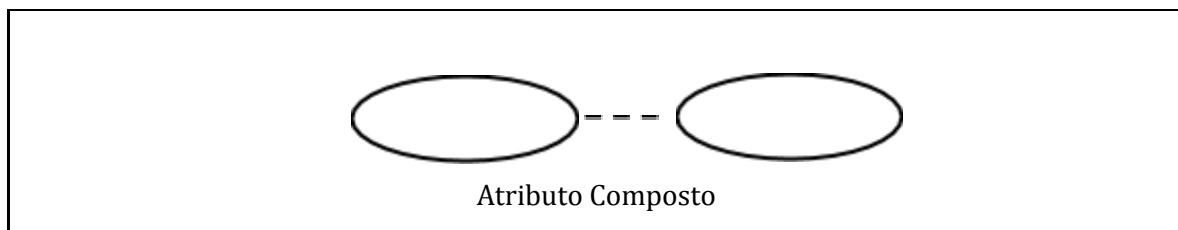
Já os atributos referenciais são chamados de Chave Estrangeira e geralmente estão ligados à chave primária da outra entidade. Estes termos são bastante comuns no contexto de bancos de dados. Mantendo o exemplo anterior, a entidade cliente tem como chave primária seu CPF, assim, a venda possui também um campo “CPF do cliente” que se relaciona com o campo CPF da entidade cliente.

3.12 DIAGRAMAS DE ENTIDADE E RELACIONAMENTO (DER)

O diagrama de entidade e relacionamento (também chamado de “DER”) é uma espécie de espelho do diagrama de classes e caracterizam apenas os atributos das classes e seus relacionamentos, tal como sua cardinalidade; assim como ocorre no diagrama de classes. Utilizamos tais diagramas principalmente nas modelagens de banco de dados, que são imprescindíveis para o entendimento da estrutura das tabelas a serem criadas e popularizadas (BARROS,2012).

Segundo Alves (2009), utilizamos as seguintes simbologias e terminologias para determinar os objetos do diagrama de entidade e relacionamento:

OBJETOS DO DIAGRAMA DE ENTIDADE-RELACIONAMENTO	
	Entidade
	Entidade Fraca
	Relacionamento
	Relacionamento Identificador
	Atributo
	Atributo Chave
	Atributo Multivalorado
	Atributo Derivado



Quadro 4: Objetos do diagrama de entidade-relacionamento (Elaboração Própria).

3.13 RELACIONAMENTO ENTRE ENTIDADES

Todas as entidades no diagrama apresentam atributos próprios, os quais ficam armazenados num banco de dados. Cada tipo de entidade é identificado pelo seu atributo que determina sua propriedade. Os atributos também podem receber certas marcações, como a chave primária que distingue a identificação da identidade de forma única. As ações existentes entre as entidades são simbolizadas por um losango que contém uma ação escrita. A cardinalidade já comentada, especifica como é realizada a forma do relacionamento da visão de negócios, de entidade para entidade (BARROS, 2012).

Enquanto o MER é um modelo conceitual, o Diagrama Entidade Relacionamento (Diagrama ER ou ainda DER) é a sua representação gráfica e principal ferramenta. Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema. Dessa forma, quando se está modelando um domínio, o mais comum é já criar sua representação gráfica, seguindo algumas regras.

O diagrama facilita ainda a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado.

Em sua notação original, proposta por Peter Chen (idealizador do modelo e do diagrama), as entidades deveriam ser representadas por retângulos, seus atributos por elipses e os relacionamentos por losangos, ligados às entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n). Porém, notações mais modernas abandonaram o uso de elipses para atributos e passaram a utilizar o formato mais utilizado na UML, em que os atributos já aparecem listados na própria entidade. Essa forma torna o diagrama mais limpo e fácil de ser lido.

Observe na Figura 5 um exemplo simples de um diagrama para um sistema de imobiliárias.

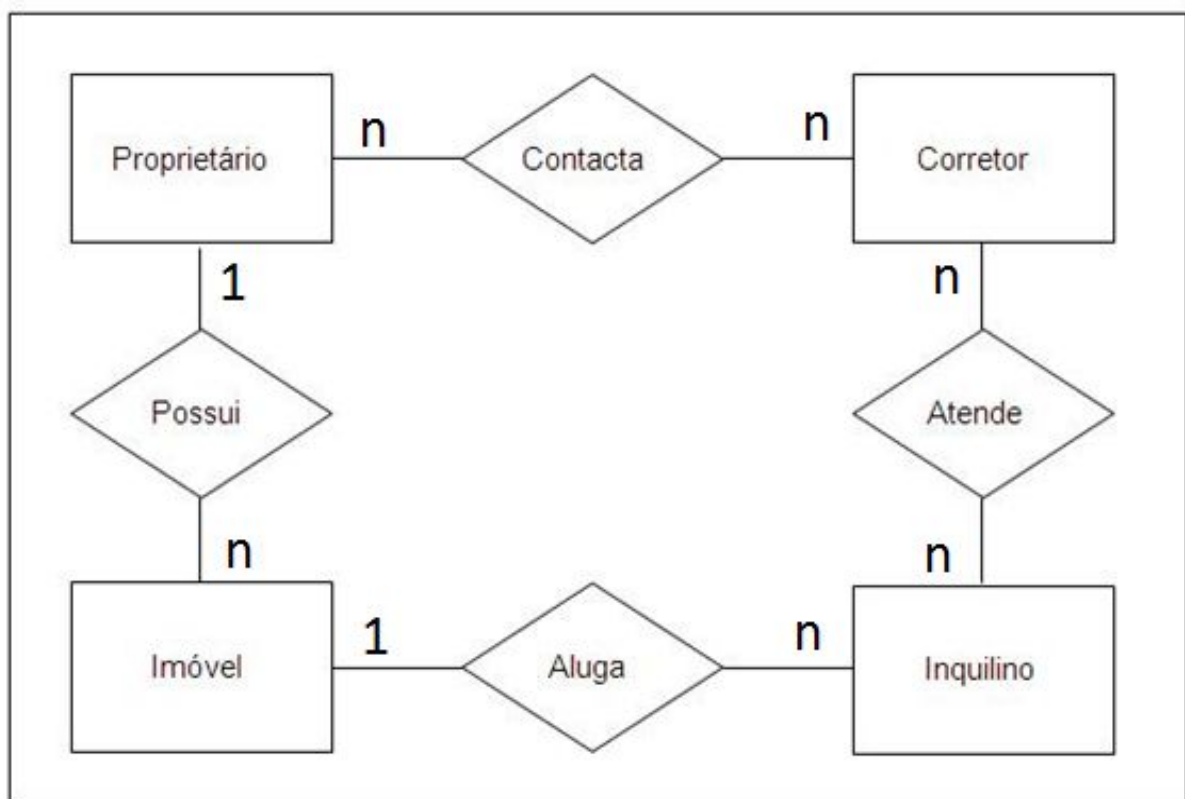


Figura 5: Diagrama entidade relacionamento de sistema imobiliário (Elaboração Própria).

No domínio representado pelo diagrama acima temos as seguintes entidades e relacionamentos:

- a) Proprietário *contacta* Corretor (um proprietário pode contatar vários corretores e um corretor pode ser contatado por vários proprietários).

- b) Corretor *atende* Inquilino (um corretor pode atender vários inquilinos e um inquilino pode ser atendido por vários corretores).
- c) Inquilino aluga Imóvel (um inquilino aluga um imóvel e um imóvel pode ser alugado por vários inquilinos).
- d) Proprietário possui Imóvel (um proprietário possui vários imóveis e um imóvel pertence a apenas um proprietário).

Uma variante da Figura 5 pode ser vista na Figura 6, onde a cardinalidade do relacionamento é exibida junto do losango.

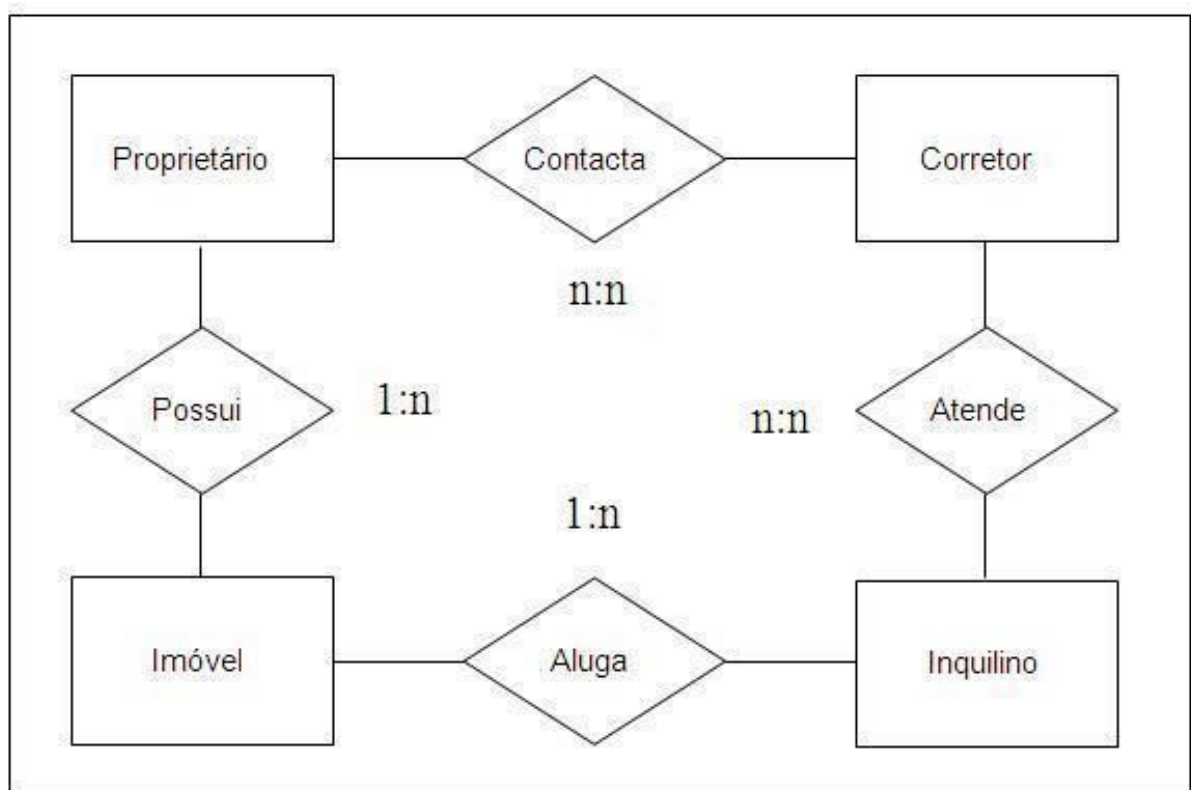


Figura 6: Diagrama entidade relacionamento (Elaboração Própria).

Uma outra variação já mostra a cardinalidade de uma forma mais completa, deixando claro as possibilidades de números de objetos envolvidos em cada relacionamento. Nesse modelo, em cada lado do relacionamento os números aparecem no formato (X, Y) ao invés de um único número como vemos nas figuras anteriores. A Figura 7 ilustra um exemplo desse tipo.



Figura 7: Diagrama entidade relacionamento (Elaboração Própria).

Neste diagrama, temos os relacionamentos da seguinte forma:

- 1 ou 1 grupo possui 0 ou muitos produtos. Como de um lado temos “1 ou 1”, isso equivale a apenas “1”, pois não temos várias possibilidades. Já do lado do produto, indicamos que um grupo pode possuir nenhum produto, mas também pode possuir vários;
- 0 ou várias vendas contém 1 ou muitos produtos. Ou seja, um produto pode nunca ser vendido (0 vendas) como também pode ser vendido várias vezes (n vendas). Já uma venda deve conter 1 ou vários produtos, pois uma venda não pode estar vazia (0 produtos).

Os atributos, como já foi dito, podem aparecer no diagrama na forma de elipses ligadas às entidades. Essa foi a notação original proposta, mas como podemos ver na Figura 8, ela deixa o diagrama com muitos itens e pode atrapalhar um pouco a organização destes.

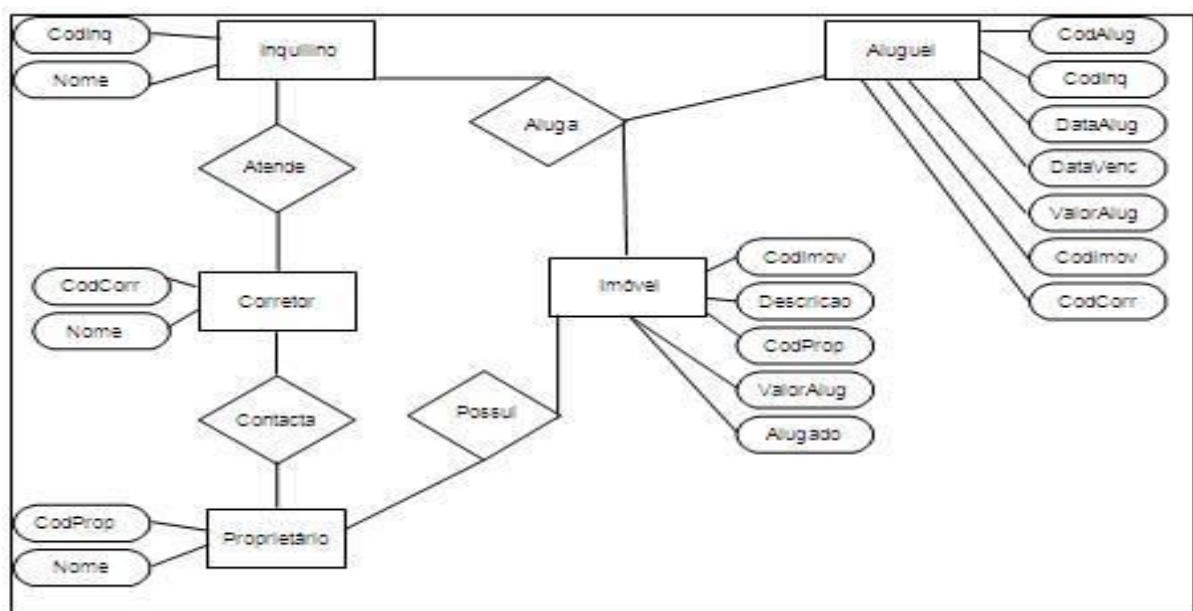


Figura 8: Atributos apresentados como elipses (Elaboração Própria).

Em uma notação mais atual, comumente utilizada na UML, os atributos aparecem listados dentro do próprio retângulo da entidade, enquanto o nome da entidade aparece no topo na forma de título. Na Figura 9 temos um exemplo.



Figura 9: Diagrama com atributos nas entidades (Elaboração Própria).

O Modelo Entidade Relacionamento (e principalmente o diagrama) é uma importante ferramenta durante o desenvolvimento de sistemas, principalmente aqueles mais complexos e difíceis de visualizar sem uma análise mais aprofundada.

Portanto a correta modelagem auxilia no correto desenvolvimento da base de dados e evita que várias alterações sejam necessárias para corrigir erros de concepção provenientes de falhas durante a análise.

3.14 DIAGRAMA DE ENTIDADE-RELACIONAMENTO DO PROJETO

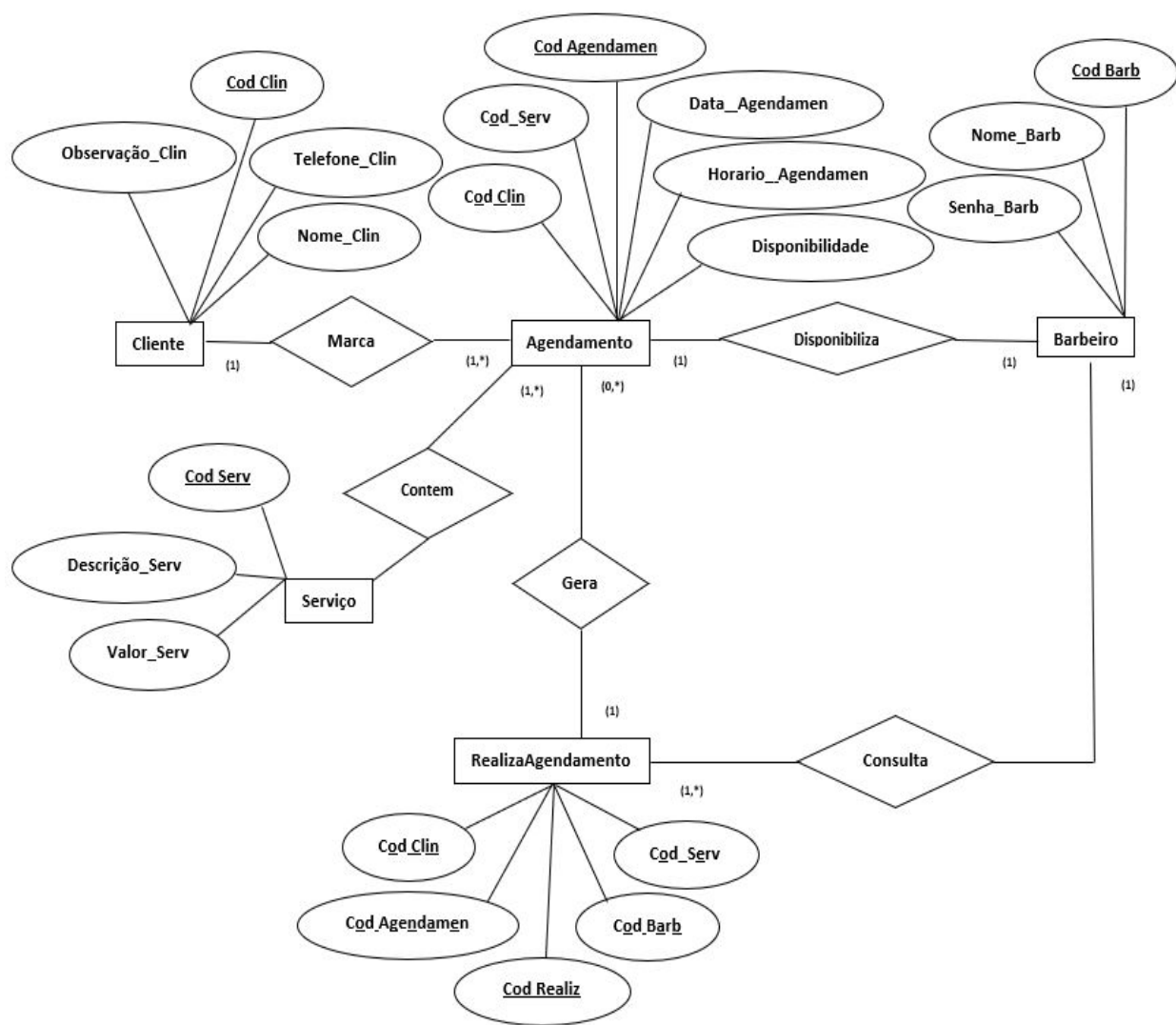


Figura 10: Diagrama de entidade-relacionamento (Elaboração Própria).

3.15 DICIONÁRIOS DE DADOS

Quadro 5: Dicionário da entidade barbeiro.

Barbeiro				
Nome do Campo	Tipo	Tamanho	Chave	Observações
Cod_Barb	Numérico		Primária	Auto Incremento

Nome_Barb	Alfanumérico	100		Não nulo
Senha_Barb	Alfanumérico	8		Não nulo

Quadro 5: Dicionário de dados da entidade Barbeiro (Elaboração Própria).

Quadro 6: Dicionário da entidade cliente.

Cliente				
Nome do Campo	Tipo	Tamanho	Chave	Observações
Cod_Clin	Numérico		Primária	Auto Incremento
Nome_Clin	Alfanumérico	100		Não Nulo
Telefone_Clin	Alfanumérico	11		Não Nulo
Observação_Clin	Alfanumérico	200		Não Nulo

Quadro 6: Dicionário de dados da entidade Cliente (Elaboração Própria).

Quadro 7: Dicionário da entidade agendamento.

Agendamento				
Nome do Campo	Tipo	Tamanho	Chave	Observações
Cod_Agendamen	Numérico		Primária	Auto Incremento
Data_Agendamen	Data			
Horario_Agendamen	Alfanumérico			Não nulo
Disponibilidade	Binário			Não nulo
Cod_Clin	Numérico		Estrangeira	Não nulo

Cod_Serv	Numérico		Estrangeira	Não nulo
----------	----------	--	-------------	----------

Quadro 7: Dicionário de dados da entidade Agendamento (Elaboração Própria).

Quadro 8: Dicionário da entidade serviço.

Serviço				
Nome do Campo	Tipo	Tamanho	Chave	Observações
Cod_Serv	Numérico		Primária	Auto Incremento
Descrição_Serv	Alfanumérico			Não Nulo
Valor_Serv	Numérico			Não Nulo

Quadro 8: Dicionário de dados da entidade Serviço (Elaboração Própria).

Quadro 9: Dicionário da entidade Realiza Agendamento.

RealizaAgendamento				
Nome do Campo	Tipo	Tamanho	Chave	Observações
Cod_Realiz	Numérico		Primária	Auto Incremento
Cod_Barb	Numérico		Estrangeira	Não Nulo
Cod_Clin	Numérico		Estrangeira	Não Nulo
Cod_Agendamen	Numérico		Estrangeira	Não nulo
Cod_Serv	Numérico		Estrangeira	Não nulo

Quadro 9: Dicionário de dados da entidade RealizaAgendamento (Elaboração Própria).

3.16 DIAGRAMA DE ENTIDADE-RELACIONAMENTO (WORKBENCH)

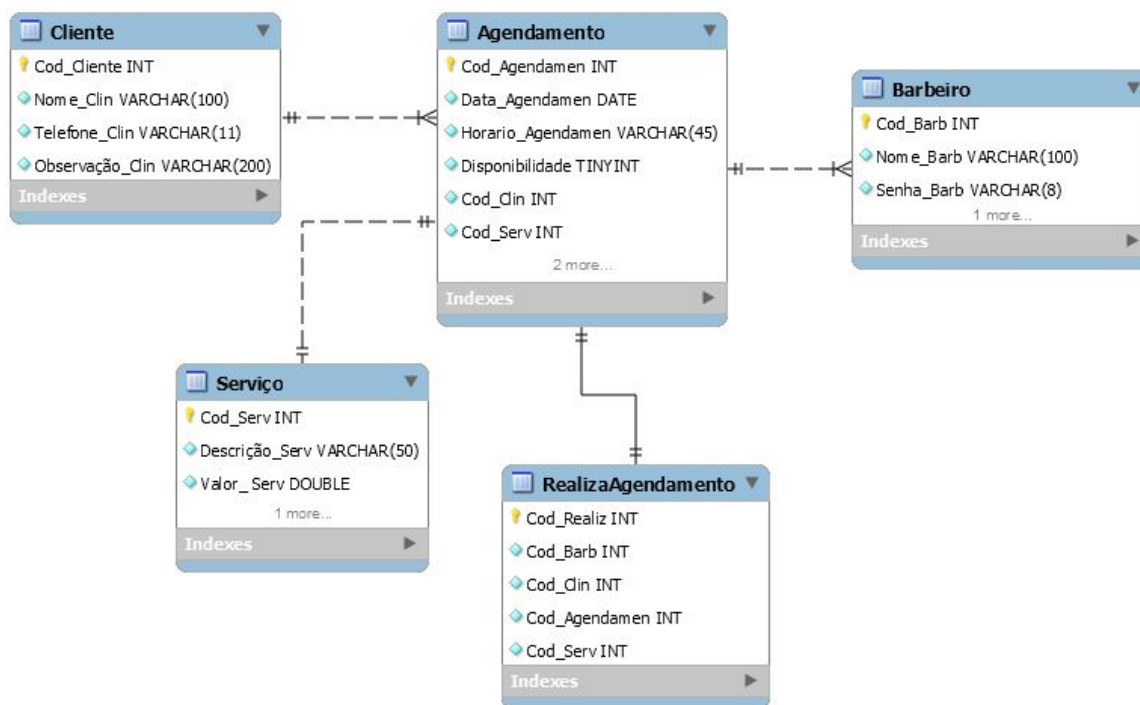


Figura 11: Diagrama de entidade-relacionamento (Workbench) (Elaboração Própria).

3.17 VISÃO GERENCIAL: GERENCIAMENTO DE PROJETO

O planejamento de *software* é algo fundamental que infelizmente não ocorre em muitas empresas que produzem *software*. Existem empresas que entregam ao cliente um *software* complexo, caro e que não passou por um planejamento minucioso e tais projetos terminam por causar muitos problemas entre empresas e clientes. Existem empresas que suprimem até mesmo a fase de teste do *software*, estipulando como política de ação, a resolução de problemas no computador do cliente; após o *software* se encontrar instalado e rodando nas máquinas de produção (BARROS, 2012).

Para um bom planejamento é preciso que realizemos algumas métricas de *software* que nos ajudam a perceber a tangibilidade da qualidade e produtividade que foi arquitetada para se definir melhores objetivos na execução completa do projeto, desde as análises iniciais até a sua implementação. Segundo Pressman (1995), são medidas diretas: as linhas de código (LOC) escritas no programa, a sua velocidade de execução, seu espaço armazenado em memória e seus defeitos registrados num período de tempo; as medidas indiretas são a qualidade, complexidade, eficiência, facilidade de uso, confiabilidade e grau de manutenção.

Medir o *software* é algo que passa despercebido por vários engenheiros até que estes percebam o caos instalado, após pular certas fases e não realizar medições dos processos. Isto torna a medição do software, uma etapa extremamente importante pois será com esta ferramenta que obteremos um panorama do sucesso do projeto. Conforme descrito por Sommerville (2003), as medições de *software* são trabalhosas e consomem grande tempo, atrasando a conclusão de um projeto; outra grande razão que é utilizada para evitar a medição é que existe apenas um apoio limitado de ferramentas para coleta e análise de dados, e não um padrão consistente em si. No entanto, gerenciar um projeto de *software* consiste em aplicar as métricas existentes para que se assegure um mínimo de qualidade e o menor número de erros possíveis na conclusão do projeto, e é de extrema importância que tais métricas sejam baseadas em padrões. Em suma, as métricas disponíveis, apesar de não serem regras que ditam universalmente todas as circunstâncias, são extremamente valiosas pois ajudam a identificar componentes com anomalias e possibilitar que estes componentes ou partes do software sejam fragmentadas e minuciosamente analisadas, com fim de eliminar a falha.

3.18 MÉTRICAS DE SOFTWARE

Abordando dois aspectos com relação à medição de software, podemos citar dois: a que se utiliza na medição de produtividade, utilizando-se linhas de código (LOC). Esta é controversa entre os autores pois a simplicidade ou complexidade da linguagem de programação pode “condenar” projetos curtos, eficientes e bem

elaborados, por terem menos linhas de código que outros programas mais extensos, mas que talvez não mais eficientes que àqueles.

Allan Albrecht propôs a medição de pontos por função (PF) que avalia a produtividade pelo exame de cinco aspectos do *software*:

- a) número de entradas do usuário;
- b) número de saídas do usuário;
- c) número de consultas do usuário;
- d) número de arquivos;
- e) número de *interfaces* externas.

As métricas de software existem pois são essenciais para avaliar se o projeto de software está progredindo ou enfrenta alguma estagnação; ou seja, não melhora. Se quisermos estabelecer uma padronização de gerenciamento de controle de qualidade e produtividade, é necessário que apliquemos durante a fase de planejamento, as métricas padronizadas para a estimativa do esforço e dos custos envolvidos, bem como o resultado final a ser apreciado.

A mensuração usando-se pontos por função nos permite obter um amplo contexto de visão de projeto, num modelo padronizado que enseja melhorias sempre que possível. Utilizamos as seguintes definições para abstrair os pontos por função: arquivos lógicos internos; arquivos de *interface* externa; entradas externas; consultas externas e saídas externas – Sendo todas estas definições são respectivamente denominadas pelas siglas ALI, AIE, EE e CE.

3.19 PONTOS POR FUNÇÃO

3.19.1 Arquivos Lógicos Internos (ALI)

Os arquivos lógicos internos são dados que são informações de controle. Residem na fronteira da aplicação e são identificados pelo usuário. Sua função é de armazenar dados alimentados pela execução de um processo de aplicação. O Dado Elementar Referenciado (DER) consiste num campo único, não repetido e

reconhecido pelo usuário. O Registro Lógico Referenciado (RLR) é um subgrupo de dados que é reconhecido pelo usuário dentro de um ALI ou um AIE.

A contagem de complexidade, tanto para os arquivos lógicos internos como para os arquivos de *interface* externa é definido pela seguinte tabela:

		DER	
RLR	1-19	20-50	> 50
1	SIMPLES	SIMPLES	MÉDIO
2-5	SIMPLES	MÉDIO	COMPLEXO
> 5	MÉDIO	COMPLEXO	COMPLEXO

Tabela 11 – Tabela para cálculo de complexidade de ALI e AIE.

No projeto em questão, temos as seguintes tabelas de arquivos lógicos internos:

Tabela ALI Agendamento

Nº Ordem	Itens de Dados	Tamanho	Tipo
1	Cod_Agendamen	6	int
2	Data_Agendamen	10	Data
3	Horario_Agendamen	6	Alfanumérico
4	Disponibilidade	0	Binário
5	Cod_Clin	6	Numérico
6	Cod_Serv	6	Numérico

DER = 3**RLR=3****Complexidade= Simples**

Tabela 12 - Arquivo lógico interno de Pedido.

Tabela ALI Serviço			
Nº Ordem	Itens de Dados	Tamanho	Tipo
1	Cod_Serv	6	int
2	Descrição_Serv	50	Varchar
3	Valor_Serv	15	double

DER = 3**RLR=1****Complexidade= Simples**

Tabela 13 - Arquivo lógico interno de Pedido.

Tabela ALI Realizar Agendamento			
Nº Ordem	Itens de Dados	Tamanho	Tipo
1	Cod_realiz	6	int
2	Cod_Barb	3	Int
3	Cod_Clin	1000	Int
4	Cod_Agendamen	1000	int

5	Cod_Serv	50	Int
---	----------	----	-----

DER= 5**RLR = 2****Complexidade = Simples**

Tabela 14 - Arquivo lógico interno da realização de agendamento.

3.19.2 Arquivos de Interface Externa (AIE)

Os arquivos de interface externa são dados que são informações logicamente relacionadas e residem fora da fronteira da aplicação.

Tabela AIE Barbeiro			
Nº Ordem	Itens de Dados	Tamanho	Tipo
1	cod_Barb	3	int
2	Nome_Barb	100	Varchar
3	Senha_Barb	8	Varchar

DER= 3**RLR=2****Complexidade = Simples**

Tabela 15 - Arquivo lógico interno de Produto.

Tabela AIE Cliente			
Nº Ordem	Itens de Dados	Tamanho	Tipo
1	cod_cliente	6	int
2	Nome_Clin	100	Varchar
3	Telefone_Clin	11	Varchar
4	Observação_Clin	200	Varchar
DER = 4		RLR = 1	Complexidade = Simples

Tabela 16 - Arquivo lógico interno de Cliente.

3.19.3 Entradas Externas (EE)

As entradas externas correspondem às telas de interface do programa, em que o usuário do software entrará com os dados para realizar as operações do sistema. Para que calculemos a complexidade das entradas externas, utilizamos a seguinte tabela para determinar a quantidade de arquivos lógicos referenciados e dados elementares referenciados:

Tabela 22: Tabela de referência para DER e ALR para telas do programa.

DER			
ALR	De1 a 4	De 5 a 15	Maior que 15
0 a 1	SIMPLES	SIMPLES	MÉDIO
2	SIMPLES	MÉDIO	COMPLEXO
>2	MÉDIO	COMPLEXO	COMPLEXO

Fonte: Elaboração Própria.

Apresentamos a seguir as telas de entrada de dados do programa, com seus campos de texto simples, caixa de texto pré-formatados, caixa de combinação de textos, botões de ação.

As funções dos botões são:

- a) O botão confirma inclui os dados digitados e ou selecionados no banco de dados;
- b) O botão novo permite a inserção de novos dados na tela;
- c) O botão cancelar cancela a ação a ser executada;
- d) O botão limpar é utilizado para limpar os campos da tela.

A tela principal apresenta as seguintes referências:

- a) DER 04;
- b) ALR 04;
- c) Complexidade Médio.

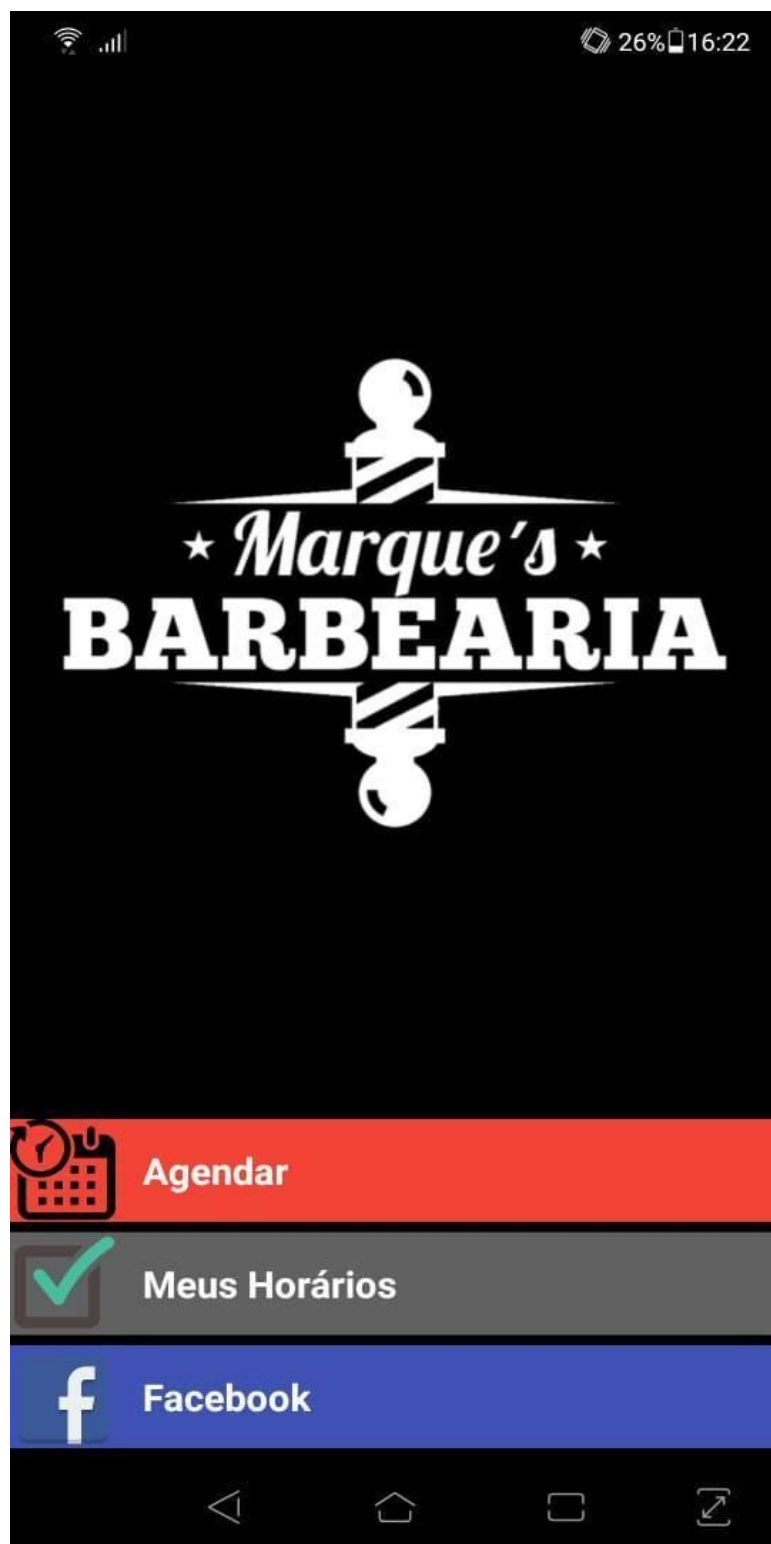


Figura 12: Cadastro do cliente (Elaboração Própria).

A tela do usuário para a escolha dos barbeiros apresenta as seguintes referências:

- a) DER 05;

- b) ALR 01;
- c) Complexidade Simples.

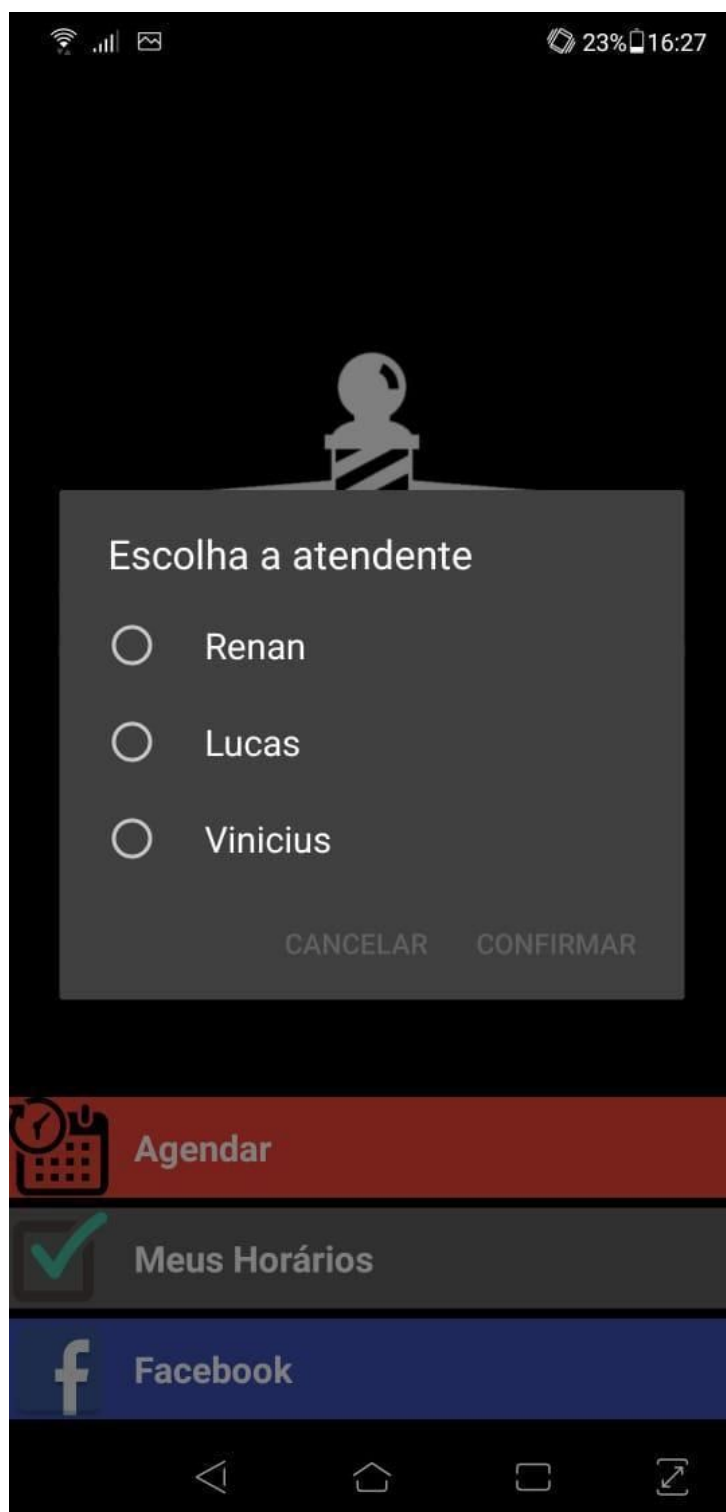


Figura 13: Cadastro do usuário (Elaboração Própria).

A tela de serviços disponíveis para os clientes apresenta as seguintes referências:

- a) DER 03;

- b) ALR 01;
- c) Complexidade Simples.

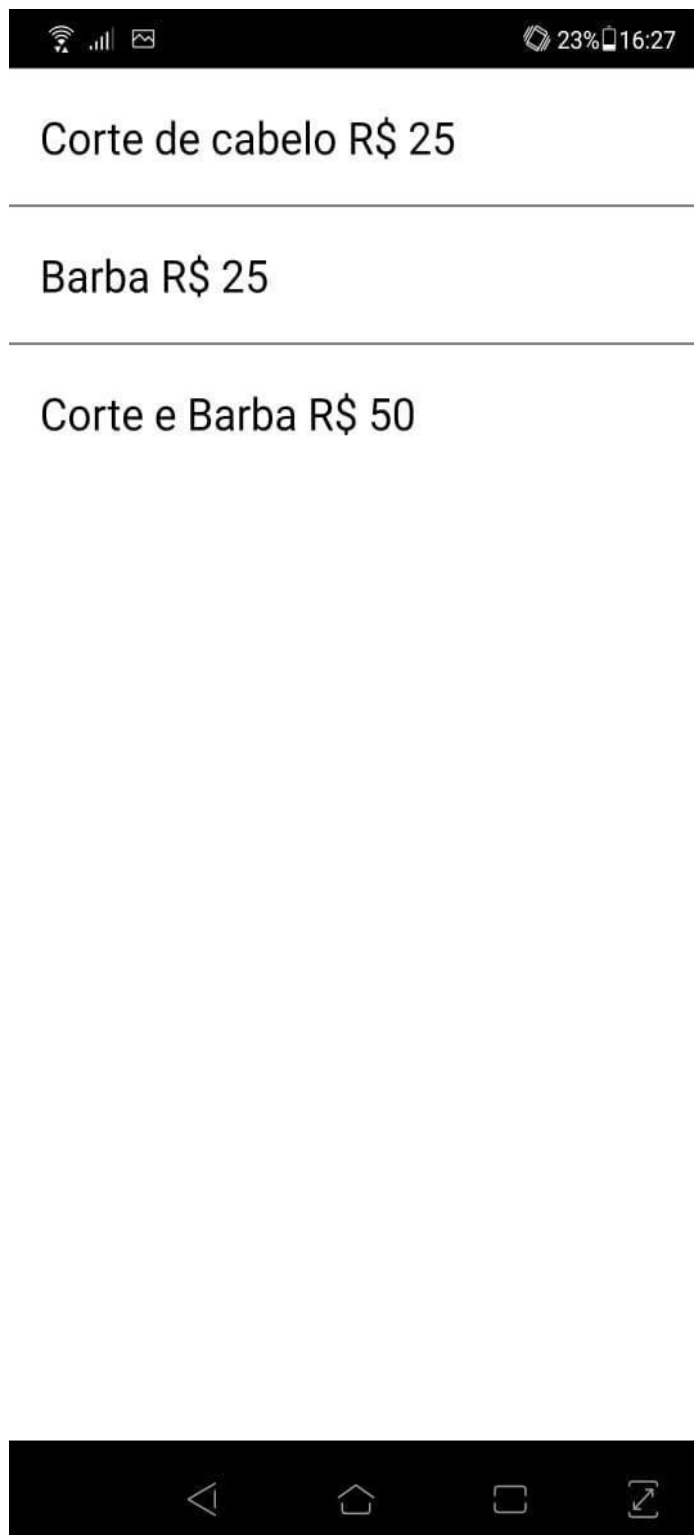


Figura 14: Cadastro da concessionária (Elaboração Própria).

A tela de Dados disponíveis para os clientes apresenta as seguintes referências:

- a) DER 05;
- b) ALR 01;
- c) Complexidade Simples.

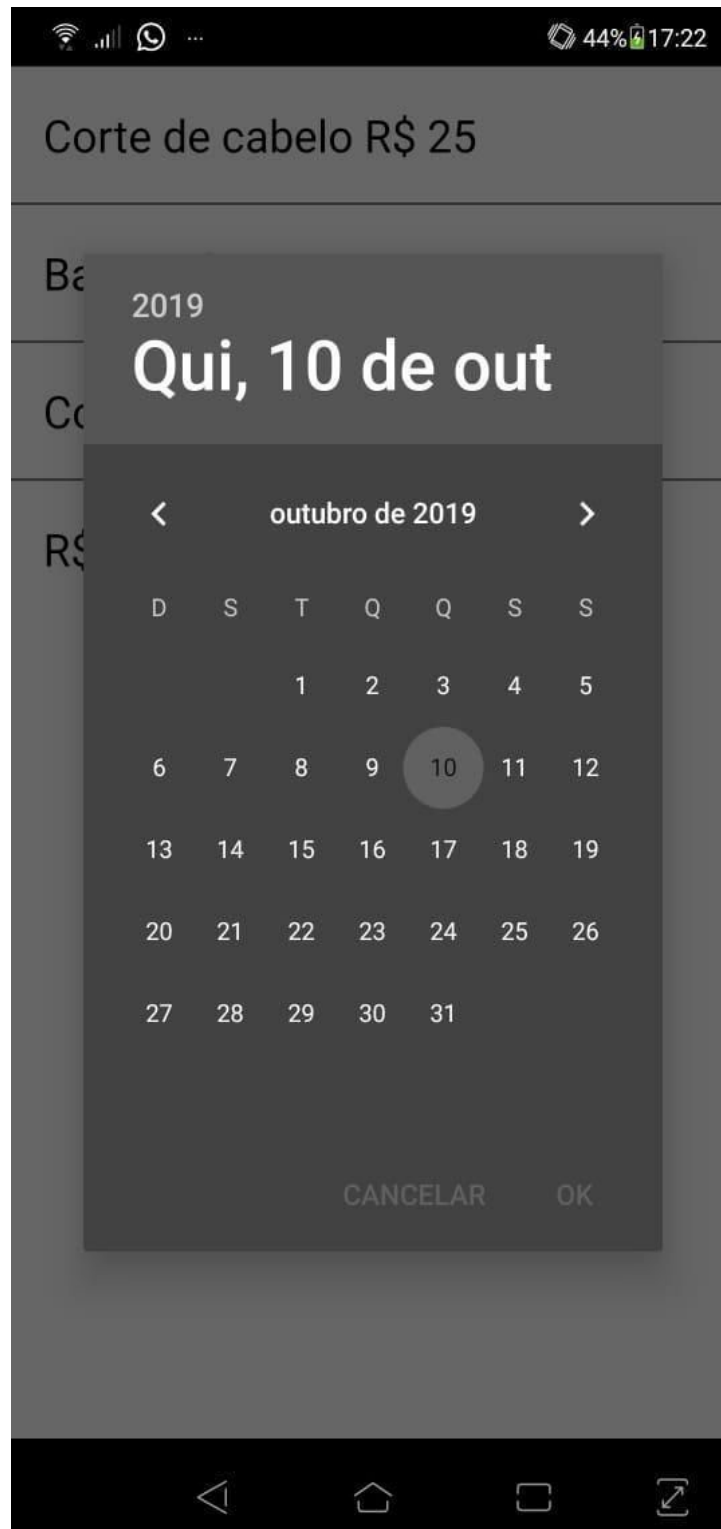


Figura 15: Cadastro do dispositivo de medição (Elaboração Própria).
A tela de Horários disponíveis para os clientes apresenta as seguintes referências:

- a) DER 22,
- b) ALR 01,
- c) Complexidade Médio.

7:00 as 7:30 - DISPONÍVEL

7:30 as 8:00 - DISPONÍVEL

8:00 as 8:30 - DISPONÍVEL

8:30 as 9:00 - DISPONÍVEL

9:00 as 9:30 - DISPONÍVEL

9:30 as 10:00 - DISPONÍVEL

10:00 as 10:30 - DISPONÍVEL

10:30 as 11:00 - DISPONÍVEL

11:00 as 11:30 - DISPONÍVEL

Figura 16: Cadastro de tarifas (Elaboração Própria).
A tela de cadastro dos dados do cliente apresenta as seguintes referências:

- d) DER 05;
- e) ALR 03;
- f) Complexidade Complexo.

The screenshot shows a mobile application interface for a checkout process. At the top, there is a status bar with icons for Wi-Fi, cellular signal, and battery, along with the time 16:28 and 23% battery. Below the status bar is a dark gray header with a white back arrow on the left and the word "Checkout" in white text. The main content area is dark gray and contains the following text in white: "DIA: 10 / 10", "Corte de cabelo 7:30 as 8:00", "Informe seus Dados:", "Nome Completo:", "Telefone:", and "Observações:". There are three white input fields: the first is labeled "Nome e Sobrenome", the second is labeled "Telefone", and the third is labeled "Observações:" and contains the text "pBS". At the bottom of the screen is a red button with the white text "Reservar". The very bottom of the screen shows the standard Android navigation bar with icons for back, home, and recent apps.

Figura 17: Cadastro do dispositivo de medição (Elaboração Própria).

A tela de aviso de responsabilidade apresenta as seguintes referências:

- a) DER 03;
- b) ALR 01;
- c) Complexidade Simples.

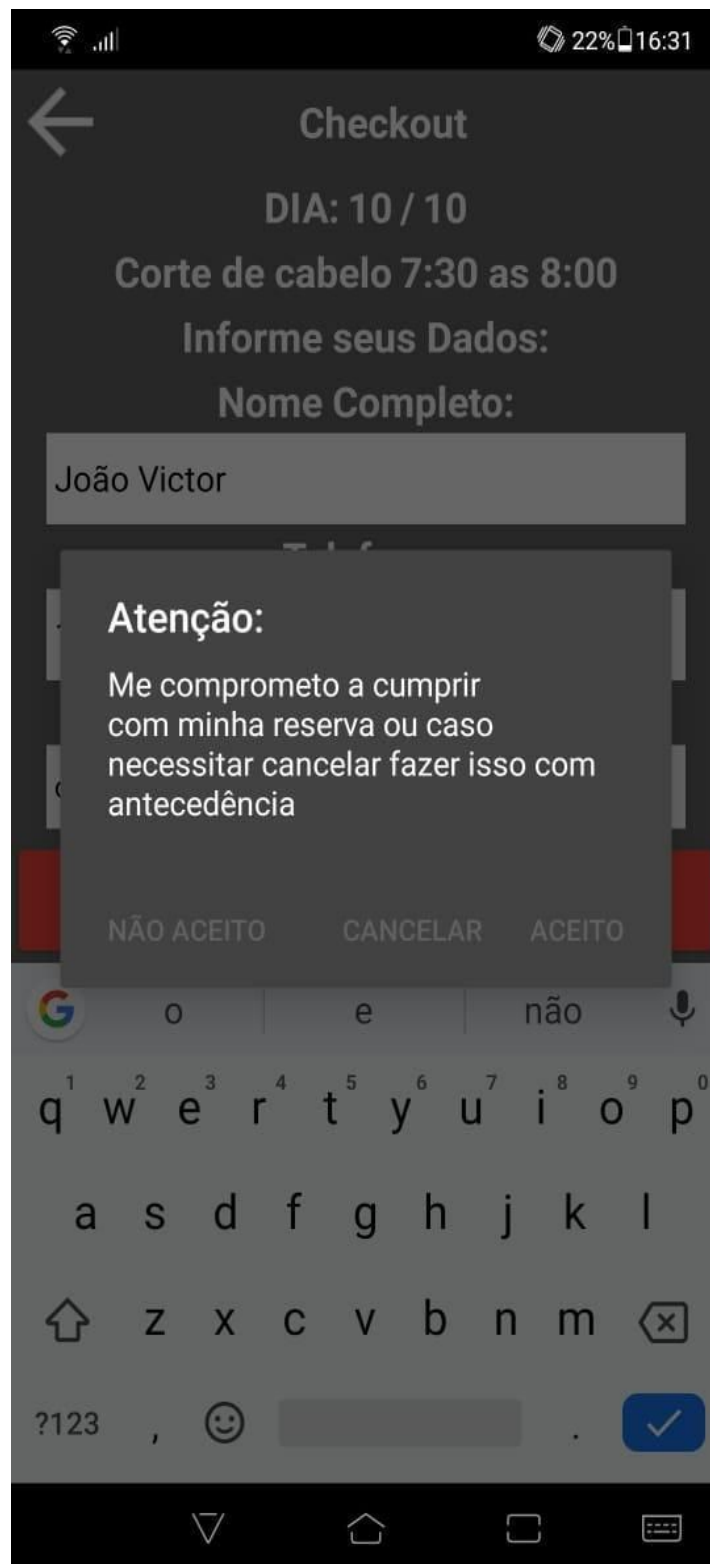


Figura 18: Cadastro do dispositivo de medição (Elaboração Própria).

A tela de aviso reservado com sucesso e sobre duvidas apresenta as seguintes referências:

- a) DER 01;
- b) ALR 01;
- c) Complexidade Simples.



Figura 19: Cadastro do dispositivo de medição (Elaboração Própria).
A tela do Barbeiro na área administrativo apresenta as seguintes referências:

- a) DER 01;
- b) ALR 01;
- c) Complexidade Simples.



Figura 20: Cadastro do dispositivo de medição (Elaboração Própria).
A tela de acesso do barbeiro apresenta as seguintes referências:

- a) DER 03;
- b) ALR 02;

c) Complexidade Simples.



The image shows a mobile application interface for a restricted access screen. At the top, there is a black status bar with white icons for Wi-Fi, cellular signal, and battery level (25%) along with the time 16:23. Below this, the title 'Acesso Restrito' is centered in bold black text, followed by the instruction 'Digite seu usuário e senha:'. There are two input fields: the first is labeled 'Usuário' and the second is labeled 'Senha'. Below the input fields is a red rectangular button with the white text 'Entrar'. At the bottom of the screen is a black navigation bar with four white icons: a back arrow, a home icon, a recent apps icon, and a share icon.

Figura 21: Cadastro do dispositivo de medição (Elaboração Própria).
A tela de menu barbeiro apresenta as seguintes referências:

- a) DER 03;
- b) ALR 03;

c) Complexidade Simples.



Figura 22: Cadastro do cliente (Elaboração Própria).

A tela de disponibilização de data de atendimento apresenta as seguintes referências:

- a) DER 05;
- b) ALR 01;
- c) Complexidade Simples.



Figura 23: Cadastro do dispositivo de medição (Elaboração Própria).
A tela de disponibilização de Horários para atendimento apresenta as seguintes referências:

- a) DER 22,
- b) ALR 01,
- c) Complexidade Médio.

Selecione os horários para o dia:

- ☐ 7:00 as 7:30
- ☐ 7:30 as 8:00
- ☐ 8:00 as 8:30
- ☐ 8:30 as 9:00
- ☐ 9:00 as 9:30
- ☐ 9:30 as 10:00
- ☐ 10:00 as 10:30
- ☐ 10:30 as 11:00
- ☐ 11:00 as 11:30
- ☐ 11:30 as 12:00
- ☐ 12:00 as 12:30
- ☐ 12:30 as 13:00
- ☐ 13:00 as 13:30
- ☐ 13:30 as 14:00
- ☐ 14:00 as 14:30
- ☐ 15:00 as 15:30
- ☐ 15:30 as 16:00
- ☐ 16:00 as 16:30
- ☐ 16:30 as 17:00
- ☐ 17:00 as 17:30

+

Figura 24: Cadastro de tarifas (Elaboração Própria).
A tela de adicionar serviço apresenta as seguintes referências:

- a) DER 01;
- b) ALR 01;
- c) Complexidade Simples.

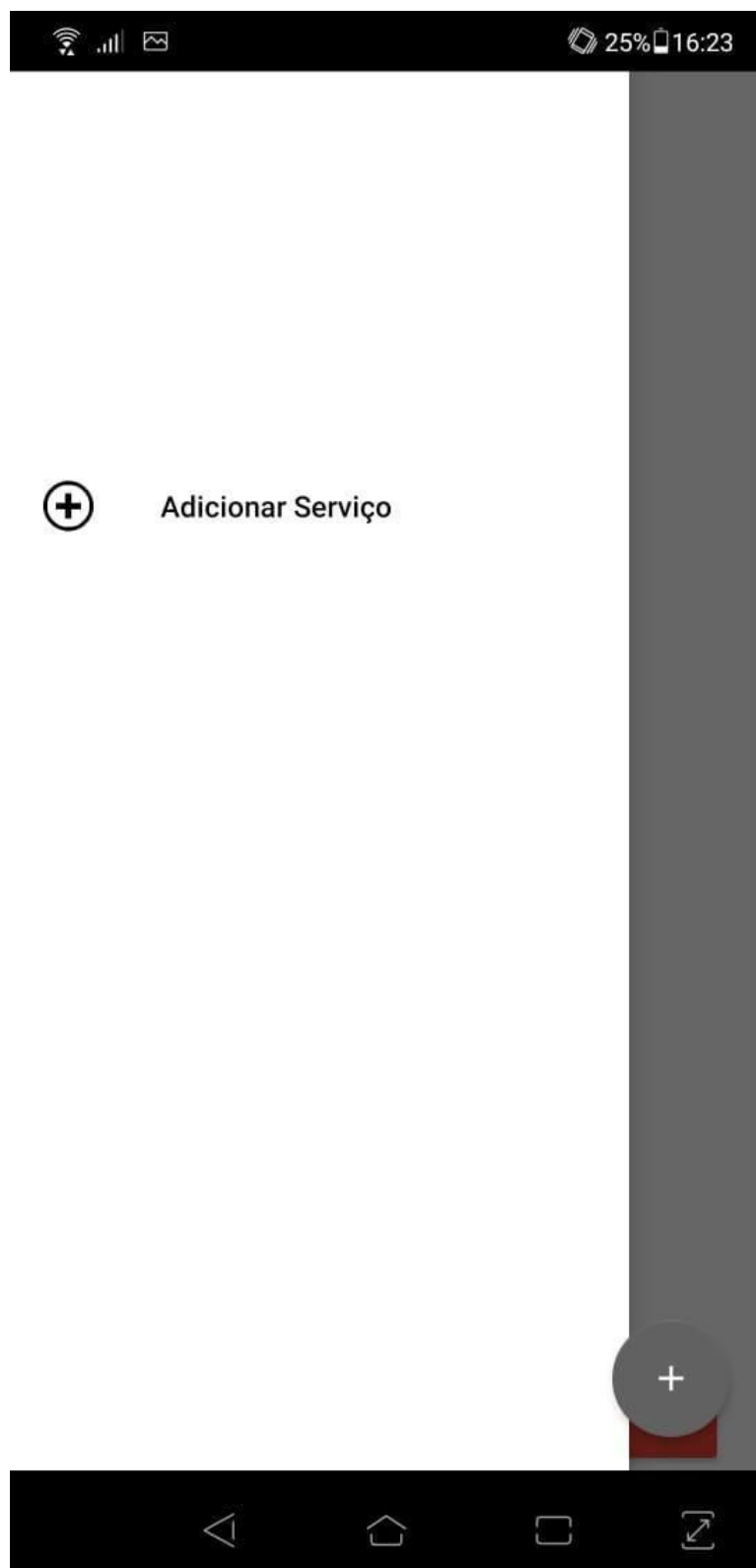


Figura 25: Cadastro do dispositivo de medição (Elaboração Própria).

A tela de cadastro de serviços apresenta as seguintes referências:

- a) DER 04;
- b) ALR 02;
- c) Complexidade Simples.

The screenshot shows a mobile application interface for adding a service. At the top, there is a status bar with icons for Wi-Fi, cellular signal, and battery, along with the text '25%' and '16:23'. Below the status bar, the text 'Adicionar Serviço:' is displayed. Underneath, there are two input fields: the first is labeled 'Título' and the second is labeled 'R\$'. Below these fields, the text 'Nenhum Serviço Cadastrado' is displayed. At the bottom of the screen, there is a red button labeled 'Salvar' and a black navigation bar with four icons: a back arrow, a home icon, a recent apps icon, and a full-screen icon.

Adicionar Serviço:

Título

R\$

Nenhum Serviço Cadastrado

Salvar

Figura 26: Cadastro do dispositivo de medição (Elaboração Própria).

3.19.4 Consultas Externas (CE)

As consultas externas consistem em toda e qualquer operação de consulta no sistema, seja de busca ou resultado visualizado.

A tela do menu de consultas não possui entradas externas, apenas opção de escolha das telas de consultas e buscas.

A tela de consulta reservas do cliente ou delete apresenta as seguintes referências:

- a) DER 02,
- b) ALR 01,
- c) Complexidade Simples.



Figura 27: Consulta do Fator de Potência (Elaboração Própria).

A tela de consulta de reserva para o barbeiro apresenta as seguintes referências:

- a) DER 22,
- b) ALR 01,
- c) Complexidade Médio.



Figura 28: Consulta do Fator de Carga (Elaboração Própria).

A tela de consulta para saber dados do cliente apresenta as seguintes referências:

- a) DER 01,
- b) ALR 01,

c) Complexidade Simples.



Figura 29: Consulta da Demanda de Consumo ativa (Elaboração Própria).

3.20 PLANEJAMENTO POR DECOMPOSIÇÃO

A decomposição é a subdivisão das entregas do projeto em componentes menores e mais facilmente gerenciáveis, até que o trabalho e as entregas estejam definidos até o nível de pacote de trabalho.

O nível de pacote de trabalho é o nível mais baixo na Estrutura Analítica do Projeto – EAP, é o ponto no qual o custo e o cronograma do trabalho podem ser estimados de forma confiável.

O nível de detalhe dos pacotes de trabalho irá variar de acordo com o tamanho e complexidade do projeto.

A decomposição de uma entrega ou subprojeto que será realizado em um futuro distante talvez não seja possível. A equipe de gerenciamento de projetos normalmente espera até que a entrega ou o subprojeto estejam esclarecidos para poder desenvolver os detalhes da EAP. Esta técnica é frequentemente chamada de planejamento em ondas sucessivas.

Entregas diferentes possuem níveis diferentes de decomposição.

Para alcançar um esforço de trabalho mais facilmente gerenciável (ou seja, um pacote de trabalho), o trabalho para algumas entregas precisa ser decomposto somente até o próximo nível, enquanto outras exigem mais níveis de decomposição.

A capacidade de planejar, gerenciar e controlar o trabalho aumenta à medida que o trabalho é decomposto em níveis mais baixos de detalhe.

No entanto, uma decomposição excessiva pode levar a um esforço de gerenciamento improdutivo, ao uso ineficiente de recursos e a uma menor eficiência na realização do trabalho. A equipe do projeto deve procurar alcançar um equilíbrio entre níveis excessivos e níveis muito baixos de detalhe no planejamento da EAP.

A decomposição do trabalho total do projeto normalmente envolve as seguintes atividades:

- Identificação das entregas e do trabalho relacionado;

- Estruturação e organização da EAP;

- Decomposição dos níveis mais altos da EAP em componentes detalhados de

nível mais baixo;

Desenvolvimento e atribuição de códigos de identificação aos componentes da EAP.

Verificar se o grau de decomposição do trabalho é necessário e suficiente. É preciso analisar a declaração do escopo detalhada do projeto para identificar as principais entregas do projeto e o trabalho necessário para produzir essas entregas. Essa análise exige um nível de opinião especializada para identificar todo o trabalho, inclusive as entregas do gerenciamento de projetos e as entregas exigidas por contrato. A estruturação e a organização das entregas e do trabalho do projeto associado em uma EAP que pode atender aos requisitos de controle e gerenciamento da equipe de gerenciamento de projetos é uma técnica analítica que pode ser realizada usando um modelo de EAP. A estrutura resultante pode assumir várias formas, como:

Usar as principais entregas e subprojetos como o primeiro nível de decomposição, conforme mostrado em modelos da EAP.

Usar os subprojetos, conforme ilustrado em modelos da EAP, na qual os subprojetos podem ser desenvolvidos por organizações fora da equipe do projeto. Por exemplo, em algumas áreas de aplicação, a EAP do projeto pode ser definida e desenvolvida em várias partes, como uma EAP do resumo do projeto com vários subprojetos dentro da EAP que podem ser contratados. O fornecedor então desenvolve a estrutura analítica do projeto contratado de apoio como parte do trabalho contratado.

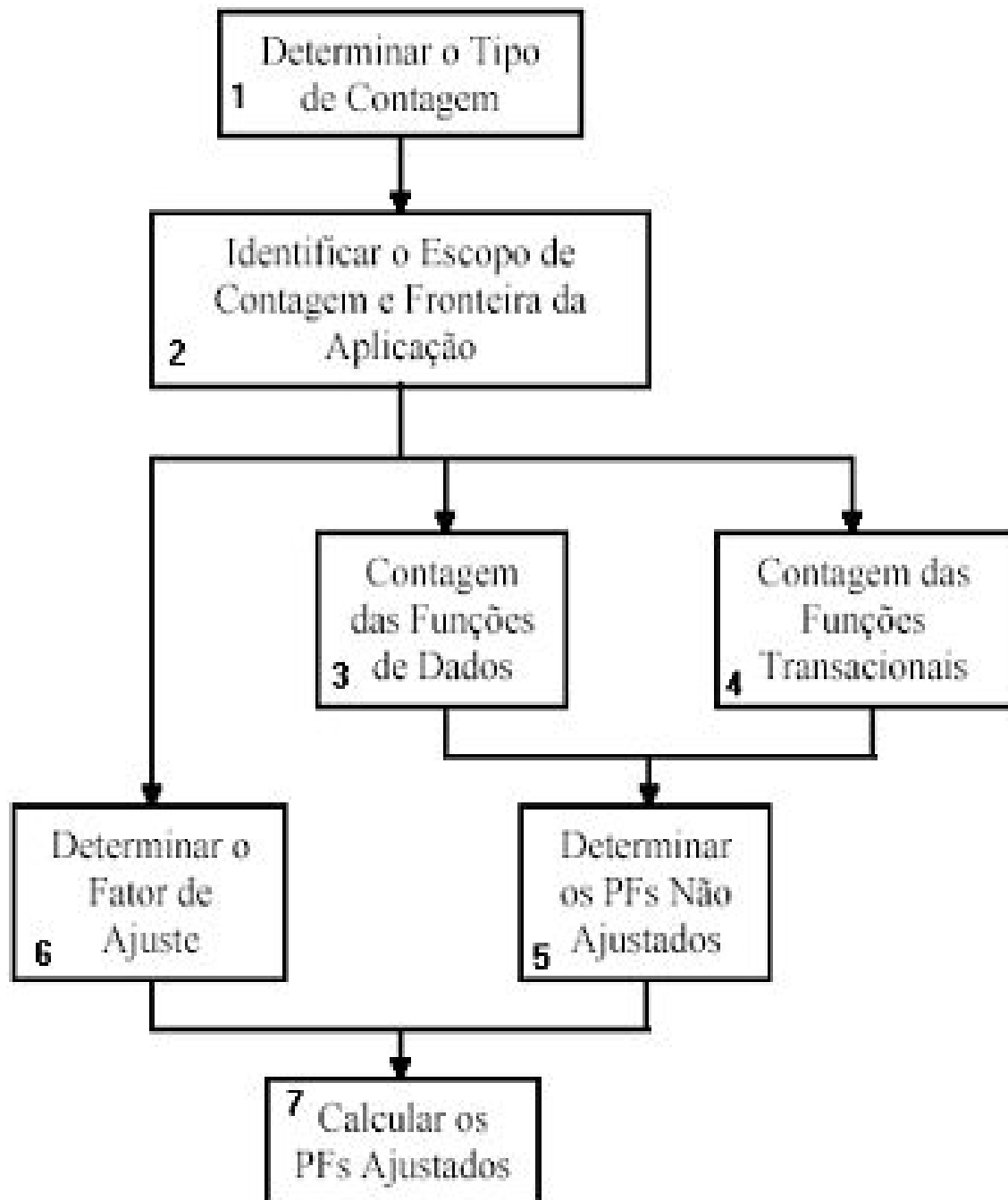


Figura 30: Procedimento de Contagem de Pontos de Função (HAZAN, 2001)

3.20.1 Tabela de contagem

Extraídas a complexidade e o total de pontos de função tanto das funções de dados (ALIs e/ou AIEs) quanto das funções de transação (EEs, CEs e/ou SEs), é preciso calcular os pontos de função não ajustados através da multiplicação do número de funções identificadas para uma determinada complexidade por sua

contribuição. Por fim, soma-se todos os pontos de função encontrados.

O foco da contagem de pontos por função é a visão do usuário, portanto somente são contados os Requisitos Funcionais, aqueles percebidos e reconhecidos por ele. Eles descrevem o que o software deverá fazer em termos de tarefas e serviços.

A seguinte tabela de contagem expressa a contagem da complexidade funcional dos itens analisados apresentados no capítulo anterior.

Tabela 23: Tabela de cálculo para aquisição dos pontos por função.

Componentes Lógicos	Complexidade Funcional	Multiplicadores	Total de Complexidade	Total Tipo de Complexidade
Arquivos Lógicos Internos	3 Simples	$X 7 =$	21	21
	0 Média	$X 10 =$	0	
	0 Complexo	$X 15 =$	0	
Arquivos de Interface Externa	2 Simples	$X 5 =$	10	10
	0 Média	$X 7 =$	0	
	0 Complexo	$X 10 =$	0	
Entradas Externas	11 Simples	$X 3 =$	33	51
	3 Média	$X 4 =$	12	
	1 Complexo	$X 6 =$	6	
Consultas Externas	2 Simples	$X 3 =$	6	10
	1 Média	$X 4 =$	4	
	0 Complexo	$X 6 =$	0	
Total de Pontos				92

Fonte: Elaboração Própria.

3.20.2 Questões de avaliação de complexidade de software

Adicionalmente inclui-se um questionário padronizado com questões relativas ao software que determinarão seu peso de complexidade, seguindo as métricas da

seguinte tabela.

As 14 questões foram elaboradas pelas métricas padronizadas na engenharia de software e possuem um certo cunho subjetivo, não obstante; não devem ser levianamente ignoradas pois tais dados revelam através das respostas, a complexidade do projeto.

É importante lembrar que os fatores mostrados estão relacionados com características da aplicação e podem influenciar no seu tamanho

Complexidade						
Numeração	0	1	2	3	4	5
Peso	Nenhum	Mínimo	Moderado	Médio	Significante	Essencial

Quadro 24: Valor de complexidade para respostas das questões (Elaboração Própria).

Tabela 25: Questionário.

Nº	Questão	Pts.
1.	O sistema exige backup e recuperação de dados?	1
2.	É requerida comunicação de dados?	5
3.	Existem funções de processamento distribuído?	2
4.	O desempenho é crítico?	0
5.	O sistema funcionará num sistema operacional existente e intensamente utilizado?	5
6.	São requeridas entrada de dados on-line?	5
7.	As entradas on-line requerem que as transações de entrada sejam construídas com várias telas e operações?	3
8.	Os arquivos são atualizados on-line?	5
9.	Entradas, saídas, arquivos e consultas são complexos?	1
10.	O processamento interno é complexo?	2
11.	O código é projetado para ser reusável?	1
12.	A conversão e a instalação estão incluídas no projeto?	3
13.	O sistema é projetado para múltiplas instalações em diferentes organizações?	5
14.	A aplicação é projetada de forma a facilitar o uso pelo usuário?	5
Total de pontos do questionário:		43

Fonte: Elaboração Própria.

3.20.3 Cálculo do Fator de Ajuste

Assim que o total de pontos por função não ajustados for obtido, o fator de ajuste deve ser determinado.

O cálculo de fator de ajuste representa a funcionalidade geral da aplicação provida ao usuário pela aplicação e é obtido através da análise das respostas pelo questionário.

O cálculo de fator de ajuste é utilizado para transformar os pontos de função brutos em pontos de função líquidos, em que se obtêm uma pontuação tal para interagir com as próximas equações que determinarão a qualidade da documentação e o custo do software.

Para obter os pontos por função líquidos, aplica-se a seguinte equação:

$$\text{PFL} = \text{PFB} * [0,65 + 0,01 * \sum_{i=1}^{14} (Fi)]$$

Em que:

- a) PFL é o que se pretende encontrar, os pontos por função líquidos;
- b) PFB são os pontos por função brutos obtidos da tabela de contagem;
- c) *Fi* é a soma dos valores de ajustes da complexidade.

Assim, a equação ficará da seguinte forma:

$$\text{PFL} = 92 * [0,65 + 0,01 * (43)]$$

$$\text{PFL} = 99,36$$

O resultado do projeto apontou um total de aproximadamente 99 pontos por função líquidos.

3.20.4 Produtividade, qualidade, preço e documentação

Obtendo-se os pontos por função líquidos através do fator de ajuste, tem-se a possibilidade de analisar 4 fatores principais no que concerne ao aspecto do software:

- a) produtividade, o cálculo do índice de produtividade é apurado pela divisão do PFL pela quantidade de pessoas ao mês;
- b) qualidade, o cálculo do índice de qualidade é apurado pela divisão da quantidade de erros do programa pelo PFL;

- c) custo, divide-se o valor monetário de custo do software pelo PFL, e obtemos a estimativa do custo.
- d) documentação, divide-se o número de páginas do documento de requisitos pelo PFL, e obtemos a estimativa da qualidade da documentação.

Apesar de os números apresentados terem um certo teor subjetivo, isto é, passível de um julgamento individual e que pode variar conforme a visão de cada um, mesmo assim é pertinente obtermos esses dados para o documento de requisitos.

As vantagens dessa métrica é que são mensuráveis independentes da linguagem de programação, são ideais para aplicações que usam linguagem não procedimental e são baseados em dados mais fáceis de serem conhecidos, durante a evolução do projeto. As desvantagens são de que o cálculo é baseado em dados subjetivos, não sendo uma medida direta, apenas números.

Na escala de 0 a 100 pontos, a produtividade pode ser definida como: 0 a 33 baixa, de 33 a 66 média, e acima de 66 alta, deve-se levar em conta o tempo de desenvolvimento.

A escala da qualidade vai de 0,0 a 0,1 e deverá levar em conta no máximo 02 erros do software, assim quanto mais próximo de 0 considera-se que o programa apresenta melhor qualidade e quanto mais próximo a 0,1 apresenta pior qualidade.

A escala do custo vai de 0 a 50, considera-se 0 custo-baixo, 25 custo-médio, e 50 custo-alto.

A escala da documentação vai de 0 a 1 sendo numa escala de 0 a 1, considera-se níveis abaixo de 0,50 uma documentação com falhas, e o número 1 como uma documentação impecável.

Podemos concluir que o ponto de função é uma métrica focada no usuário (qualquer pessoa ou coisa que se comunica ou interage com o software a qualquer momento).

Tabela 26: Métricas de pontuação.

Métricas de Pontuação			
Produtividade (0 a 100)	0 a 33 Baixa	33 a 66 Média	>66 Alta
Qualidade (0,0 a 0,1)	0 a 0,033 Boa	0,033 a 0,066 Média	>0,066 Ruim
Custo (0 a 50)	0 a 16 Baixo	16 a 32 Médio	>32 Alto
Documentação (0 a 1)	0 a 0,33 Ruim	0,33 a 0,66 Médio	>0,66 Bom

Fonte: Elaboração Própria.

Tabela 27: Resultados finais de pontuação.

Fator	Conta	Resultado	Conclusão
Produtividade	99 / 2	49,5	Média, ideal seria 01 pessoas
Qualidade	2 / 99	0,020	Boa Qualidade
Custo	2.500 / 99	25,25	Médio Custo
Documentação	68 / 99	0,68	Documento Bom

Fonte: Elaboração Própria

3.21 VISÃO SOBRE DISTRIBUIÇÃO E IMPLEMENTAÇÃO (DEPLOYMENT)

A visão sobre distribuição e implementação do software precisa considerar uma gama de tópicos de avaliação sobre hardware e a relação entre todos os dispositivos interligados no sistema.

Ao visualizar um sistema sendo implementado, o engenheiro precisa compreender de maneira clara e objetiva sobre que tipo de hardware e quais relações entre todos os dispositivos de hardware interligados receberão o software para devida operação.

Neste projeto apresenta-se um modelo de visão de implementação como uma aplicativa web representada por uma tabela de descrição de requisitos de hardware e software.

Tabela 28: Requisitos de hardware.

Requisitos de Hardware	
Equipamento	Característica
Desktop e clientes	CPU 2GHz, 4GiB RAM, 500 GiB de disco, Eth 10/100 Wi-fi
Servidor Web	CPU 4 x 2.6GHz, 4 GB RAM, 1 Tib de disco, Eth10/100 Wi-fi
Rede Internet	ADSL ou rede corporativa
Modem	GSM, GPRS ou 3G
Gateway	CLP Controlador Multigrandezas com armazenamento local

Fonte: Elaboração Própria.

Tabela 29: Requisitos de software.

Requisitos de Software	
Aplicação	Característica
Sistema Operacional	Windows Server 2016
Servidor Web	Apache
Servidor Dados	PhpMyAdmin / MySql
Navegadores	Opera ou Google Chrome atualizados
Segurança	Firewall integrado no sistema operacional

Fonte: Elaboração Própria.

4. CONSIDERAÇÕES FINAIS

Os resultados encontrados no desenvolvimento da pesquisa, sugerem que a automatização de processos não é mais um diferencial e sim uma obrigação para os negócios que, de fato, querem ser competitivos. Durante o desenvolvimento da pesquisa, tivemos alguns problemas que atrasaram o desenvolvimento do conceito do trabalho. Um deles é a falta de fontes de informação que falem sobre Gestão de agendamentos e desenvolvimentos de aplicativos, que são vagos no google acadêmico, em alguns casos, tivemos que recorrer a alguns blogs de pequeno porte que direcionam tópicos para essa área de interesse. A maior fonte de informação contida neste trabalho, foi retirada do SEBRAE, é uma fonte que realmente explora a área da automatização comercial e documenta seus resultados positivos e negativos. Uma boa opção, também foi recorrer a casos de automatizações que, de fato, tiveram um resultado positivo e números que mostram que a automatização realmente virou uma obrigação para a empresa que quer ser competitiva perante o mercado atual.

Também surgiram várias dúvidas durante a escolha das ferramentas adequadas de desenvolvimento, visto que os membros do grupo não tinham experiência com os métodos dos desenvolvimentos de software, sendo assim, esse foi o primeiro protótipo desenvolvido pelo grupo. Inicialmente, a proposta de

utilização era totalmente direcionada ao App Inventor, porém, foi avaliado que o Kodular se encaixa muito mais com as nossas ideias e expectativas, assim como a metodologia e o jeito de produzir o produto. A programação em blocos trouxe uma grande oportunidade para as pessoas que estão começando a desenvolver aplicativos e não tem experiência nessa área, possibilitando tirar seus projetos do papel e produzir um protótipo para expor suas ideias de maneira mais ilustrativa e interativa. A estratégia para um na nossa produtividade, foi a divisão de tarefas internas do trabalho entre: Desenvolvimento e documentação do Protótipo e Desenvolvimento do trabalho escrito.

Atualmente os dados e informações são cada vez mais importantes para a empresa, o modo como são mantidas é indiscutivelmente essencial para que o negócio tenha uma boa organização e administração. O desenvolvimento dos sistemas de informações está sendo utilizados em larga escala, a qual só cresce, para atender as necessidades das empresas, permitindo assim, um melhor controle no gerenciamento de serviços prestados.

No presente trabalho, o sistema foi desenvolvido para aprimorar o atendimento e gestão de uma barbearia. O objetivo principal foi atingido com êxito, o Sistema de agendamento foi desenvolvido, cumprindo todos os requisitos.

Agora é possível ao usuário do sistema/proprietária do estabelecimento, automatizar o agendamento aos seus clientes e serviços e principalmente ter um controle dos agendamentos realizados, pois como fora mencionado no início deste trabalho, os agendamentos são realizados de maneira manual, agenda de anotações com caneta, o que podem ser perdidos ou danificados a qualquer momento. Agora com o sistema, é possível ter tudo armazenado e salvo no banco de dados, melhor organizados e mantidos, podendo ser acessados a qualquer momento.

O desenvolvimento deste trabalho trouxe para o conhecimento pessoal, um aprimoramento nos conceitos de sistemas de informação, os quais foram adquiridos para implementar técnicas da área principalmente quanto ao programar em blocos juntamente com o banco de dados. Conclui-se com este trabalho que os sistemas de informações estão auxiliando sempre que estudadas possibilidades de melhoria nos diversos tipos de negócios. Basta traçar objetivamente as metas que quer chegar para depois elaborar o projeto com todos os requisitos desejados, e então chegar na

parte principal que é o desenvolvimento, sendo assim os sistemas de informações podem simplificar e muito a vida das empresas, organizando e gerenciando melhor seus dados, e inevitavelmente mantendo-os seguros.

REFERÊNCIAS

ALVES, W. P. **Banco de Dados: Teoria e Desenvolvimento**. 1ª Edição, São Paulo: Editora Érica, 2009.

BARROS, D. DE A. **Projeto_Ortodontia_-_Documento_de_Requisitos_v_2**. Tatuí, SP, Brasil, 2012.

CRISTINA, Ana. **UML – Diagrama de Sequências**. 2009. Disponível em: <<https://www.devmedia.com.br/artigo-engenharia-de-software-15-uml-diagrama-de-sequencias/13820>>. Acesso em: 03 abr.2018.

HAZAN, Cláudia. **Análise de Pontos por Função**. 2001. Disponível em: <<http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/APF.pdf>>. Acesso em: 23 jun. 2018.

PRESSMAN, R. S. **Engenharia de Software**. 3ª Edição, São Paulo: Makron Books, 1995.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. Disponível em: <<http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/graduacao/SIII/Uml/diagramas/usecases/usecases.htm>>. Acesso em: 25 mar.2018.

RIBEIRO, Rene. **Kodular: crie aplicativos para Android mesmo sem saber nada de programação**. 2019. Disponível em: <<https://olhardigital.com.br/noticia/kodular-crie-aplicativos-para-android-mesmo-se-m-saber-nada-de-programacao/81286>>. Acesso em: 08 out. 2019.

SEBRAE. **Barbearia**. Disponível em: <<http://www.sebrae.com.br/sites/portalsebrae/ideias/como-montar-uma-barbearia,f8787a51b9105410vgnvcm1000003b74010arcrd>>. Acesso em: 10 abr. 2019.

SEBRAE. **Como montar um salão de beleza**. Disponível em: <<http://m.sebrae.com.br/sites/PortalSebrae/ufs/mg/artigos/como-abrir-um-salao-de-beleza-em-minas-gerais,b7bc401746af8410VgnVCM2000003c74010aRCRD>>. Acesso em: 10 jun. 2019.

SEBRAE. **Demanda do público masculino pelo mercado da beleza.** Disponível em:<<http://m.sebrae.com.br/sites/PortalSebrae/ufs/pb/artigos/demanda-do-publico-masculino-pelo-mercado-da-beleza,e2bd83c66797a610VgnVCM1000004c00210aRCRD>> Acesso em: 12 Jun. 2019.

SEBRAE. **Ideias de negócios.** Disponível em:<<http://intranet.df.sebrae.com.br/download/barbearia.pdf>>. Acesso em: 03 jun. 2019.

SOMMERVILLE, Ian. **Engenharia de Software.** 6ª Edição, São Paulo: Pearson Addison Wesley, 2003.

SOMMERVILLE, Ian. **Engenharia de Software.** 9ª Edição, São Paulo: Pearson Prentice Hall, 2013.