

# IMPLEMENTAÇÃO DE UM APLICATIVO MOBILE PARA REGISTRO E AGENDAMENTO DE ATIVIDADES

DA SILVA, Hariel Viana Morales<sup>1</sup>  
RODRIGUES, Thiago Veleda Ianzer<sup>1\*</sup>  
CAMARGO, Salvador Loní Tadeo<sup>2</sup>  
BASTOS, Rafael Rodrigues<sup>2</sup>

<sup>1</sup>Discentes do Curso Sistemas de Informação, Nível 6 2020/1 - Faculdade IDEAU – Bagé/RS.

<sup>2</sup>Docentes do Curso Sistemas de Informação - Faculdade IDEAU – Bagé/RS.

\*E-mail para contato: thiagoianzer@gmail.com

**RESUMO:** O presente estudo aborda o processo de desenvolvimento de um aplicativo móvel para dispositivos android que funcionará integrado à um sistema web desenvolvido em um trabalho anterior (Thiago; Hariel 2019), com intuito de facilitar o gerenciamento e o desenvolvimento das atividades de uma empresa, possibilitando o acesso as informações em qualquer momento e lugar, bem como, tornando possível que novas informações sejam inseridas em tempo real, necessitando para isso ter acesso à internet. No entanto, consultas às informações já registradas podem ser realizadas off-line. Foram realizadas pesquisas bibliográficas com foco no desenvolvimento de aplicativos android, e elaborados diagramas UML durante a fase do projeto para permitir uma maior compreensão do funcionamento do mesmo. O aplicativo foi elaborado seguindo o paradigma de orientação a objetos e a linguagem de implementação foi o Java, sendo utilizada a IDE Android Studio para o desenvolvimento e testes do aplicativo. O banco de dados a ser integrado, feito em PostgreSQL, foi desenvolvido em um projeto anterior. O aplicativo permite o cadastro de clientes, agendamento e acompanhamento de serviços, registro de serviços realizados, sendo possível ter acesso aos serviços prestados aos clientes cadastrados de uma forma detalhada. Este aplicativo móvel foi elaborado de forma a ser uma extensão do sistema web desenvolvido previamente, com funcionalidades reduzidas, mas permitindo acesso a todos os principais dados cadastrados no banco do sistema, e também, fornecendo a possibilidade de inserção de dados desde que o dispositivo esteja conectado à internet.

**Palavras-chave:** Sistemas android; Android Studio; Aplicativo móvel; Integração de sistemas.

**ABSTRACT:** The present study addresses the process of developing a mobile application for android devices that will work integrated to a web system previously developed (Thiago; Hariel 2019), the main intention is bring more facilities for management and development of activities in business, enabling access to information at anytime and anywhere, allowed that user can insert new information in real time as long as the device has internet access. however registered queries can be accessed offline. Bibliographic searches were carried out on android application development, UML diagrams were built during the design phase to allow a better understanding of how it works. The application was developed following the Object-oriented paradigm and the implementation language was Java, Android Studio IDE being used for application development and testing. The database that will be integrated, built in PostgreSQL, was developed in a previous project. The application allows the registration of customers, scheduling and monitoring services, realized services record, being possible to have access to the services provided to registered customers in a more detailed way. This application was built to be an extension of the previously developed web system, with reduced functionality, but permitting the access of all main data registered in the data base of system, and also, providing the possibility of insertion of data as long as the device is connected to the internet

**Keywords:** Android systems; Android Studio; Mobile applications; Systems integration.

## 1. INTRODUÇÃO

Estamos em uma era onde a velocidade e a acessibilidade a informações são fatores primordiais para o crescente desenvolvimento tecnológico. A sociedade mundial está voltada a informação, visto que, tornaram-se dependentes de *smartphones* e outros dispositivos

móveis que permitem buscar e acompanhar as informações em tempo real, e em praticamente qualquer lugar.

O uso de internet disseminou-se por meio dos dispositivos móveis, configurando uma alternativa eficiente e prática frente ao cenário tecnológico atual, contribuindo, deste modo, para a propagação das informações a qualquer momento, acarretando em um crescente aumento de produtividade nas atividades do cotidiano.

Na última década os celulares passaram por diversas transformações, assemelhando-se a computadores pessoais e gerando um novo mercado para o desenvolvimento de aplicações móveis, capazes de transmitir dados, abrindo, desta forma, um universo de possibilidades para facilitar o dia a dia de pessoas e empresas, gerando assim, um novo mercado destinado ao desenvolvimento de aplicações móveis que prometem facilitar a vida dos usuários e melhorar a eficiência das empresas, trazendo para estas a possibilidade de um gerenciamento eficiente, graças a mobilidade e acesso em tempo real para monitoramento de suas atividades.

Diante dessa evolução e surgimento de novas tecnologias, a informatização tornou-se mais presente no cotidiano das pessoas, visto que as mesmas estão passando cada vez mais tempo conectadas, utilizando estas inovações não só no ambiente de trabalho e estudo, como também em atividades pessoais e de entretenimento.

Em função desse avanço, os aplicativos móveis tornam-se ferramentas importantes no suporte das tarefas diárias nas empresas, facilitando sua conexão com os clientes, bem como, acessando de modo eficiente seus dados arquivados em um banco de dados, sendo assim, mais um modo de manipular as atividades em qualquer momento e lugar de modo prático, dinamizando o resultado final.

Este projeto tem a finalidade de implementar um aplicativo para dispositivos móveis para ser integrado a um sistema de informação web previamente desenvolvido, para gestão comercial, desenvolvido em PHP. O aplicativo melhora a dinâmica das atividades dos usuários, trazendo para estes uma maior portabilidade, pois poderão realizar o manuseio do sistema em qualquer lugar que possua acesso à internet.

O aplicativo mobile foi concebido sob paradigma de Orientação a objetos e utilizou as seguintes ferramentas para o desenvolvimento: a linguagem de programação Java; a IDE Android Studio versão jdk; o gerenciador de banco de dados PostgreSQL.

O aplicativo irá apresentar algumas funcionalidades reduzidas em comparação com o sistema web. Destacam-se as seguintes funcionalidades no aplicativo: calendário, cadastro de pessoas, manipulação de agendamento, pesquisas e relatórios, bem como, anotações

pertinentes ao desenvolvimento das atividades. Ressalta-se que todo o tipo de inserção de dados somente poderá ser realizado quando o dispositivo estiver conectado à internet, no entanto, pesquisas e consultas de informações já inseridas podem ser realizadas off-line.

## 2. DESENVOLVIMENTO

### 2.1 Referencial Teórico

O desenvolvimento do aplicativo móvel para registros e agendamento de atividades empresariais, funcionará integrado a um sistema web pré-existente, onde as informações poderão ser acessadas em qualquer momento e lugar através de um telefone móvel. Para inserir dados e informações basta estar conectado à internet, no entanto, para ter acesso às informações já arquivadas não será necessário estar conectado à internet.

Para atender os objetivos deste trabalho, realizaram-se pesquisas bibliográficas referentes ao tema proposto, onde a partir destas, foi possível desenvolver e finalizar o trabalho proposto.

#### 2.1.1 Dispositivos e aplicações móveis

Conforme Mateus e Loureiro (1998), a computação móvel representa um novo paradigma computacional surgindo como a quarta revolução na computação, antecedida pelos grandes centros de processamento de dados da década de 60, o surgimento dos terminais nos anos 70 e as redes de computadores na década de 80.

Bastables (2010) cita que vivemos em uma sociedade voltada à informação, onde adultos e crianças do século XXI tornaram-se dependentes de *smarthphones* e, somos desafiados a acompanhar as atividades em tempo real, pois na atualidade tornou-se muito importante para produção de conhecimento.

Castells e Espanha (1999) relatam que a Era da informação impactou pelo fato de que as tecnologias de internet congregaram e conectaram pessoas ao redor do mundo, proporcionando as mesmas possibilidades de acesso rápido às informações que até então não era possível a circulação em tempo real.

Segundo Fast-Growth (2015), o tempo gasto em um telefone móvel triplicou entre os anos de 2012 e 2015, os aplicativos presentes nesses dispositivos podem facilitar a vida dos usuários, mas também podem apresentar alguns riscos, visto que os dados pessoais e de lugares visitados, além de mensagens podem ser informações que podem ser utilizadas de

forma inapropriada. Portanto conhecer todos os recursos e capacidades disponíveis é essencial para melhorar a segurança.

Em 2007 a Apple promoveu uma quebra de paradigma com o iPhone, por tratar-se de um dispositivo móvel, que possuía a capacidade e o poder muito próximo de um computador tradicional, com a diferença de ser portátil e possibilitar que seus usuários pudessem acessar recursos computacionais como e-mail, internet e outras aplicações.

Dwivedi (2010) enumera algumas boas práticas que, se adotadas, podem aumentar a confiança do usuário ao utilizar um aplicativo móvel, e também ajudam o desenvolvedor a minimizar custos e riscos desnecessários, dentre elas:

- 1) descartar os dados coletados o mais rápido possível após o uso;
- 2) manter os dados de forma anônima;
- 3) mostrar ao usuário quando e quais dados estão sendo coletados;
- 4) ter uma política de privacidade bem definida e consistente;
- 5) utilizar a menor precisão desejável para funcionalidade;
- 6) não repassar dados coletados para outros serviços e pessoas;
- 7) conhecer as leis locais que são relevantes para o serviço;
- 8) adequar a política de privacidade e a utilização para cada local onde ela existe.

### 2.1.2 Sistema Android

O sistema foi baseado em *Linux* desenvolvido pela *Android, Inc.* a qual foi adquirida pelo *Google* em 2005. Em 2007 foi formada a *Open Handset Alliance* para desenvolver, manter e aprimorar o *android*, trazendo assim, inovação para a tecnologia móvel, melhorando a experiência do usuário e reduzindo os custos.

Uma das vantagens de desenvolver aplicativos *android* está na abertura da plataforma. O sistema operacional é de código fonte aberto e gratuito, permitindo ver o código fonte do *android* e como seus recursos são implementados. Qualquer pessoa pode contribuir para o *android* relatando erros ou participando de grupos de discussão *Open Source Project*.

O Android SDK (software development kit) fornece as ferramentas necessárias para construir aplicações *android*. É disponibilizado gratuitamente aos desenvolvedores.

Segundo Da Silva e Pereira (2009), o android nos apresenta um conjunto de aplicações fundamentais que são: um cliente de e-mail, programa de SMS, agenda, mapas, navegador, contatos, entre outros. Todos esses aplicativos presentes no sistema foram desenvolvidos usando a linguagem de programação Java.

### 2.1.3 UML (*Unified Modeling Language*)

Conforme Pressman (2006), UML é uma linguagem padrão para descrever/documentar projetos de software. A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de software/intenso. Em outras palavras, assim como os arquitetos criam plantas e projetos para serem usados por uma empresa de construção, os arquitetos de software criam diagramas UML e especificações para ajudar os desenvolvedores de software a construir o software.

Grady Booch, Jim Rumbaugh e Ivar Jacobson desenvolveram a UML na década de 1990 com muita realimentação da comunidade de desenvolvimento de software. A UML combinou um grupo de notações de modelagem concorrentes usadas pela indústria do software na época.

Na UML 2.0 temos diferentes tipos de diagramas para o uso na modelagem de software, entre eles podemos citar: diagramas de classe, distribuição, caso de uso, sequência, atividade, comunicação e estados.

A linguagem UML permite expressar todos os aspectos importantes de um sistema, e além de ajudar parte do projeto do sistema, pois servem como uma planta dele, podem ser utilizados para fins de documentação do sistema.

### 2.1.4 Banco de Dados e Sistema Gerenciador de Banco de Dados

Dado é um fato que pode ser armazenado em um ambiente digital, normalmente em tabelas de Banco de Dados Relacional, permitindo que se gere relacionamentos e outras transformações, gerando uma informação para o usuário do sistema. Por exemplo: CPF, que identifica um cliente de uma loja ou um eleitor que é identificado pelo número de seu título eleitoral.

Banco de dados é uma coleção de dados relacionados que são coerentes e possui um significado inerente. Heuser (1998), cita que um Banco de Dados é um conjunto de tabelas contendo atributos, relações, índices, tipificações e outras características próprias.

A ideia de Bancos de dados disseminou-se e popularizou-se, pois permite uma estruturação, consistência, persistência e suporta o desenvolvimento e a operação de aplicações de grande utilidade para comunidade usuária.

Um banco de dados quando informatizado é mantido e acessado por meio de um software conhecido com SGBD (Sistema Gerenciador de Banco de Dados), que são

programas que permitem ao usuário definir, construir e manipular base de dados para as mais diferentes finalidades.

O uso de um banco de dados tem como vantagem a possibilidade de poder ser acessado por vários usuários simultaneamente. O modelo relacional é o modelo mais usado para representar e armazenar dados em um SGBD, e suas estruturas têm a forma de tabelas compostas por linhas e colunas.

Quem criou o modelo relacional em 1970 foi Edgar Franck Codd, e trata-se de uma teoria matemática criada com o objetivo de descrever como as bases de dados devem funcionar. É um modelo flexível e adequado ao solucionar problemas que se colocam no nível da concepção e implementação da base dados. A estrutura principal do modelo relacional é a relação, ou seja, a tabela. A relação é construída por um ou mais campos que definem os tipos de dados a serem armazenados.

Segundo Bittencourt (2004), a estrutura de um banco de dados relacional é armazenada em um dicionário de dados ou catálogo do sistema, este dicionário de dados é composto de um conjunto de relações idênticas em propriedades às relações utilizadas para armazenar dados. Em um dicionário de dados podem ser encontradas informações como: definição de colunas que compõe cada tabela, restrição de integridade imposta sobre relações, informações de segurança e definições de outros elementos estruturais.

Para Bittencourt (2004), sistema de banco de dados possui como vantagem o controle centralizado de seus dados operacionais, como: redução de redundâncias, evitar inconsistência, padronização, compartilhamento dos dados, manutenção da integridade, restrição de segurança, equilíbrio das necessidades conflitantes e independência dos dados.

A linguagem utilizada em um SGBD é geralmente SQL (Structured Query Language), que é subdividido em grupos de comandos de acordo com a função de cada comando. A finalidade principal de usar um SGBD é ter um controle central dos dados e dos programas de acesso. O responsável por esse controle sobre o sistema é chamado de administrador do banco de dados (DBA), que se trata da pessoa ou grupo de pessoas responsável pelo controle do sistema.

#### 2.1.5 Vantagens dos aplicativos móveis

Os aplicativos móveis possuem muitas vantagens entre os seus usuários, dentre elas:



a) trazem aos seus usuários uma experiência rápida e agradável, além de possibilitar uma melhor utilização de recursos gráficos e de interface, possibilitando também que os conteúdos sejam acessados sem conexão com a internet;

b) menor custo de acesso, pois toda parte da interface instalada no celular implica em um tráfego de dados muito menor para se acessar um determinado conteúdo na internet;

c) acesso a recursos nativos do celular, como *bluetooth*, câmera fotográfica, GPS, agenda telefônica e etc;

d) maior usabilidade, pois permite um carregamento mais rápido das suas funções e uma maior funcionalidade, podendo ser utilizado em qualquer lugar que o usuário estiver.

#### 2.1.6 Desvantagens das aplicações móveis

Dentre as desvantagens das aplicações móveis podemos citar:

a) a atualização da versão do aplicativo requer que o usuário efetue uma nova instalação da nova versão disponibilizada;

b) plataformas distintas, pois como existem muitos fabricantes e plataformas de desenvolvimento, ocorre que as aplicações estejam limitadas as versões específicas do android, chamadas APIs, fazendo com que o seu uso seja limitado a certos aparelhos;

c) certas aplicações possuem um alto consumo de energia para executar suas tarefas, consumindo, desta forma, muita energia da bateria do aparelho;

d) alguns aplicativos possuem um alto consumo de memória, fazendo assim, com que o desempenho do aparelho seja drasticamente reduzido, deixando a memória insuficiente para o funcionamento de outros recursos e aplicações.

#### 2.1.7 IDE (ambiente de desenvolvimento integrado)

Trata-se de um programa de computador que possui funcionalidades e características de apoio ao desenvolvimento de software, com a finalidade de agilizar esse processo. É um software que combina ferramentas comuns de desenvolvimento em uma única interface gráfica do usuário. Um IDE consiste geralmente em:

- a) editor de código-fonte;
- b) automação de compilação local;
- c) debugger.

Os desenvolvedores utilizam IDEs, visto que elas permitem uma maior produtividade, ajudando a programar novas aplicações de forma rápida, pois os inúmeros utilitários não

precisam ser ajustados e integrados manualmente durante a configuração. Os IDEs também ajudam os desenvolvedores a organizar o seu fluxo de trabalho e resolver problemas.

Existem diversos IDEs para diferentes linguagens e plataformas, como por exemplo o *Android Studio* que é destinado ao desenvolvimento de aplicações móveis para dispositivos android, possui todas as ferramentas necessárias para o desenvolvimento nessa plataforma.

Baseando-se em dados publicados por um dos fabricantes conceituados da ferramenta *Red Hat*, que classificou as características técnicas segundo os seguintes parâmetros: quantidade de linguagens compatíveis, sistemas operacionais suportados, funcionalidades de automação, impacto no desempenho de sistemas, plug-ins e extensões.

## 2.2 Programação Orientada a Objetos

Conforme Souza, Scigliano e Rosa (2018), o paradigma de programação orientada a objetos (POO) trabalha com o conceito de organizar, classificar e abstrair informações. Em uma definição menos formal, pode-se dizer que significa organizar o mundo real como uma coleção de objetos e seus atributos e métodos. Surgiu com a finalidade de facilitar o desenvolvimento de softwares que atendam às necessidades dos usuários.

Trata-se de uma abstração de objetos do mundo real, tendo como finalidade a compreensão do programa a ser criado. Aproxima o mundo real do mundo virtual, partindo como ponto principal, a simulação do mundo real dentro do computador. O programador é responsável por manipular os objetos e formalizar as suas interações. As ações do objeto são definidas pelo programador.

Para Ricarte (2001), alguns itens que formam a POO são: classes, objetos, atributos, métodos, construtores, abstração, encapsulamento, herança, poliformismo, dentre outros.

Na POO os objetos são gerados a partir de classes pré-definidas, servindo como um modelo padrão para estes objetos. As classes são modelos ou plantas dos objetos a serem gerados; geralmente são desenvolvidas durante a modelagem do sistema e servem como um padrão para os objetos a serem criados. Algumas vantagens da POO:

a) todo software orientado a objetos é confiável, pois ao alterar uma parte nenhuma outra é afetada;

b) como o software orientado a objetos é dividido em partes, permitem que várias delas sejam desenvolvidas em paralelo, economizando tempo e distribuindo o desenvolvimento;



- c) as atualizações são mais precisas e consomem menos tempo, face ao desenvolvimento modular, visto que uma pequena modificação beneficia todas as outras que usarem o objeto;
- d) é reutilizável, pois podemos utilizar um objeto de um sistema em outro sistema que será criado posteriormente;
- e) utiliza o mesmo padrão durante todo o desenvolvimento;
- f) permite que os desenvolvedores tenham um maior nível de abstração durante o desenvolvimento.

### 2.3 Praticidade da integração de sistemas

A importância da integração do sistema é para que haja um aumento na produtividade e na eficiência das atividades, além de possibilitar uma maior confiabilidade das tarefas desempenhadas, pois mesmo que as atividades rotineiras sejam executadas em ambientes diversos e distantes, a integração permite que todos os acessos sejam inseridos de forma rápida, reduzindo erros e desinformações.

A integração dos sistemas possui minimamente dois aspectos importantes: um técnico e o outro econômico. Do ponto de vista técnico possibilita a harmonização dos dados arquivados referente a empresa e possui inúmeras vantagens, podendo trazer o aumento de produtividade, reduzir as falhas de comunicações, redução de erros, gestão inteligente de produção e controle de estoque, integração de uma plataforma de assinatura eletrônica, dentre outras vantagens.

Já do ponto de vista econômico permite que as empresas desenvolvam suas atividades em qualquer lugar e a qualquer hora, reduzindo gastos, agilizando as tarefas e diminuindo o tempo no desenvolvimento das mesmas, além de promover o aumento de produtividade de suas atividades do cotidiano.

Para que a integração ocorra de uma forma adequada e que seja capaz de atender as expectativas de seus usuários, deve atentar-se a alguns fatores, como: definir as finalidades da integração, identificar as soluções desejadas, treinar as equipes, relatar o controle dos processos internos, acompanhar alterações realizadas e executar testes antes de sua implantação.

## 3. MATERIAI E MÉTODOS

Foram realizados diversos estudos acerca do tema e dos elementos envolvidos, sendo estes aplicados no desenvolvimento do trabalho. O aplicativo móvel desenvolvido possui como finalidade proporcionar uma maior praticidade as atividades do usuário, possibilitando que tenha um acesso rápido e imediato aos arquivos pré-existentes, além de armazenar dados novos necessários para a organização dos serviços prestados, necessitando para isso, estar apenas conectado à internet.

O aplicativo trará portabilidade para melhorar a dinâmica das atividades dos usuários, com as seguintes funcionalidades: cadastro de pessoas, manipulação de agendamento, pesquisas e relatórios, anotações e calendário.

Foram levantados e analisados os requisitos do aplicativo, e após elaborado um documento de requisitos e também alguns diagramas UML, criados utilizando o software Dia, para auxiliar no desenvolvimento do aplicativo e posteriormente facilitar a sua manutenção. O método de programação orientado a objetos foi o escolhido, pois diante das circunstâncias e ferramentas utilizadas se mostrou o mais adequado.

A IDE utilizada para a criação do aplicativo foi o Android Studio e a linguagem de programação utilizada foi Java. A interface foi desenvolvida utilizando XML. O banco de dados aplicado e previamente desenvolvido foi o PostgreSQL.

### 3.1 Linguagens

#### 3.1.1 Java

A linguagem de programação Java é orientada a objetos criada em 1995, ela foi originalmente projetada para rodar em aparelhos de televisão. Permite que os desenvolvedores escrevam os programas apenas uma vez e o executem por meio de qualquer dispositivo. O Java é uma linguagem versátil que se diferencia de outras linguagens de programação principalmente por algumas características, vejamos:

- a) simplicidade: por ser eficiente e derivada de C e C++, porém sem fazer uso dos seus recursos complexos e pouco utilizados;
- b) distribuição de programas: fornece uma grande biblioteca padrão e diversas ferramentas de distribuição dos programas;
- c) multiplataforma: apesar de existirem muitas linguagens multiplataformas o Java se destaca nesse quesito, sendo relativamente fácil executá-lo em qualquer tipo de hardware, tornando-o portátil;

- d) função coletor de lixo: quando não existir referencias localizadas para um objeto o coletor de lixo o excluirá e desta forma irá liberar memória, evitando possíveis perdas de memória desnecessárias.
- e) multi-tarefas: permite executar várias tarefas simultaneamente e dentro da mesma aplicação, permitindo uma maior velocidade de execução e melhora no seu desempenho;
- f) segurança: trata-se de uma plataforma segura para desenvolver e executar aplicações que auto gerenciam a memória. Utiliza canais seguros que protegem os seus dados e por ter uma sintaxe rigorosa proporciona que não haja sua corrupção através da quebra de código.

### 3.1.2 XML

O XML (*Extensible Markup Language*) é linguagem de marcação que define uma série de regras para codificação de documentos, ou seja, um aglomerado de códigos que podem ser utilizados na leitura de dados ou textos. O XML permite que os desenvolvedores definam elementos de marcação de uma forma personalizada. Um arquivo XML é dividido em sessões:

- a) **Prolog**: são metadados administrativos que são utilizados para declaração XML, definir o tipo de documento, instrução de processamento opcional e comentários;
- b) **Body**: é composto pelas partes estruturais e pelo conteúdo.

O padrão XML possui simplicidade, usabilidade e generalidade, o que o torna usual em diversos serviços na web, e é utilizada também na criação de interfaces para dispositivos móveis

### 3.2 Android Studio

É um ambiente de desenvolvimento integrado que é destinado para o desenvolvimento de apps para android, sendo baseado em IntelliJ. O Android Studio conta com diversos recursos que auxiliam sua produtividade na compilação de aplicativos android, oferecendo desde um simples editor de texto a outras funções como: sistema de compilação flexível baseado em gradle, ambiente unificado que permite desenvolvimento para diversas APIs do android, emulador rápido com inúmeros recursos e versões do android, diversos frameworks e ferramentas de testes, compatibilidade com C++ e NDK e ferramentas de lint que permitem detectar problemas desempenho, dentre outros.

### 3.3 Dia

É um software de código aberto, destinado a criação de fluxogramas e diagramas UML, utilizados principalmente por desenvolvedores na parte de projeto do software. Ele possui suporte para diferentes tipos de diagramas como: fluxogramas, diagramas de rede, modelos de banco de dados. Possui inúmeros objetos prontos que ajudam a desenhar diagramas, além de poder ler e escrever vários formatos diferentes de imagens raster e vetorial.

### 3.4 PostgreSQL

É um banco de dados relacional avançado e robusto, de código aberto, que possui suporte a maior parte do padrão da linguagem SQL e fornece diversas funcionalidades atuais como: possibilitar a construção de sentenças SQL elaboradas com a simplicidade da linguagem e com funcionalidades capazes de executar a tarefa com boa performance.

Pode ser ampliado pelo usuário de várias formas, podendo ser adicionadas novas funções, tipos de dados, operadores, métodos de índice, linguagens procedurais e funções de agregação.

## 4. RESULTADOS E DISCUSSÕES

Este trabalho abordou o desenvolvimento de um aplicativo móvel para registro e agendamento de serviços. O seu objetivo será o de armazenar os registros de serviços, tornando mais dinâmico e organizado o ambiente de trabalho. O sistema possibilitará ao usuário realizar a organização de suas atividades, bem como, o monitoramento das mesmas e a possibilidade de acompanhar o histórico dos registros.

O início deste projeto se deu mediante a elaboração dos diagramas de casos de uso e de classe, tendo uma grande base no diagrama ER do banco de dados. Através dos diagramas foi possível modelar o aplicativo para ter uma maior compreensão do mesmo, bem como, entender o que deveria ser desenvolvido.

Após esta etapa iniciou-se a parte da implementação no Android Studio. Para isso ser possível foram realizados estudos que possibilitaram a aprendizagem do manuseio da IDE, Android Studio, visto que a IDE possui uma estrutura com inúmeras ferramentas de desenvolvimento para atender as mais diversas necessidades, pois os dispositivos android apresentam inúmeras diferenças. Foi necessário ter uma noção das linguagens Java para o desenvolvimento back-end e XML para o desenvolvimento front-end.

O Java se faz essencial no desenvolvimento de aplicações para android, e apesar de termos outras opções de linguagem como Kotlin, ela ainda é a linguagem mais utilizada e por consequência possui uma maior variedade de conteúdo para estudo. Através do Java foram desenvolvidas algumas funcionalidades do aplicativo.

O XML é utilizado para desenvolver a interface das aplicações android, e apesar do Android Studio oferecer uma ferramenta gráfica intuitiva para a elaboração das interfaces dos aplicativos, ainda se faz essencial o conhecimento do XML, pois muitas vezes é necessário codificar manualmente, e pelas experiências no desenvolvimento desse projeto se torna até mais prático.

O aplicativo ainda não foi desenvolvido, mas foram realizadas as etapas de modelagem e também se deu início a parte de implementação, sendo desenvolvidas neste processo algumas telas da interface de usuário do aplicativo, bem como, a programação de algumas de suas funcionalidades.

Esta aplicação deverá ser integrada a um sistema web, desenvolvido previamente para o gerenciamento e desenvolvimento de atividades cotidianas, tornando, desta forma, a aplicação versátil para os usuários, podendo tanto ser usada no dispositivo móvel quanto em qualquer dispositivo que possua um navegador.

#### **4.1 Diagrama ER do banco de dados**

Este diagrama foi desenvolvido durante a elaboração do banco de dados utilizado no sistema web desenvolvido anteriormente. O banco de dados, que foi desenvolvido em PostgreSQL, será utilizado no aplicativo móvel, pois o aplicativo e o sistema serão integrados futuramente.

A principal diferença é que o aplicativo não apresentará a opção de cadastro e manipulação do estoque, visto que, na visão dos desenvolvedores essa opção não se faz necessária para o aplicativo, pois o gerenciamento do estoque deve ser feito na sede do estabelecimento para evitar possíveis erros.

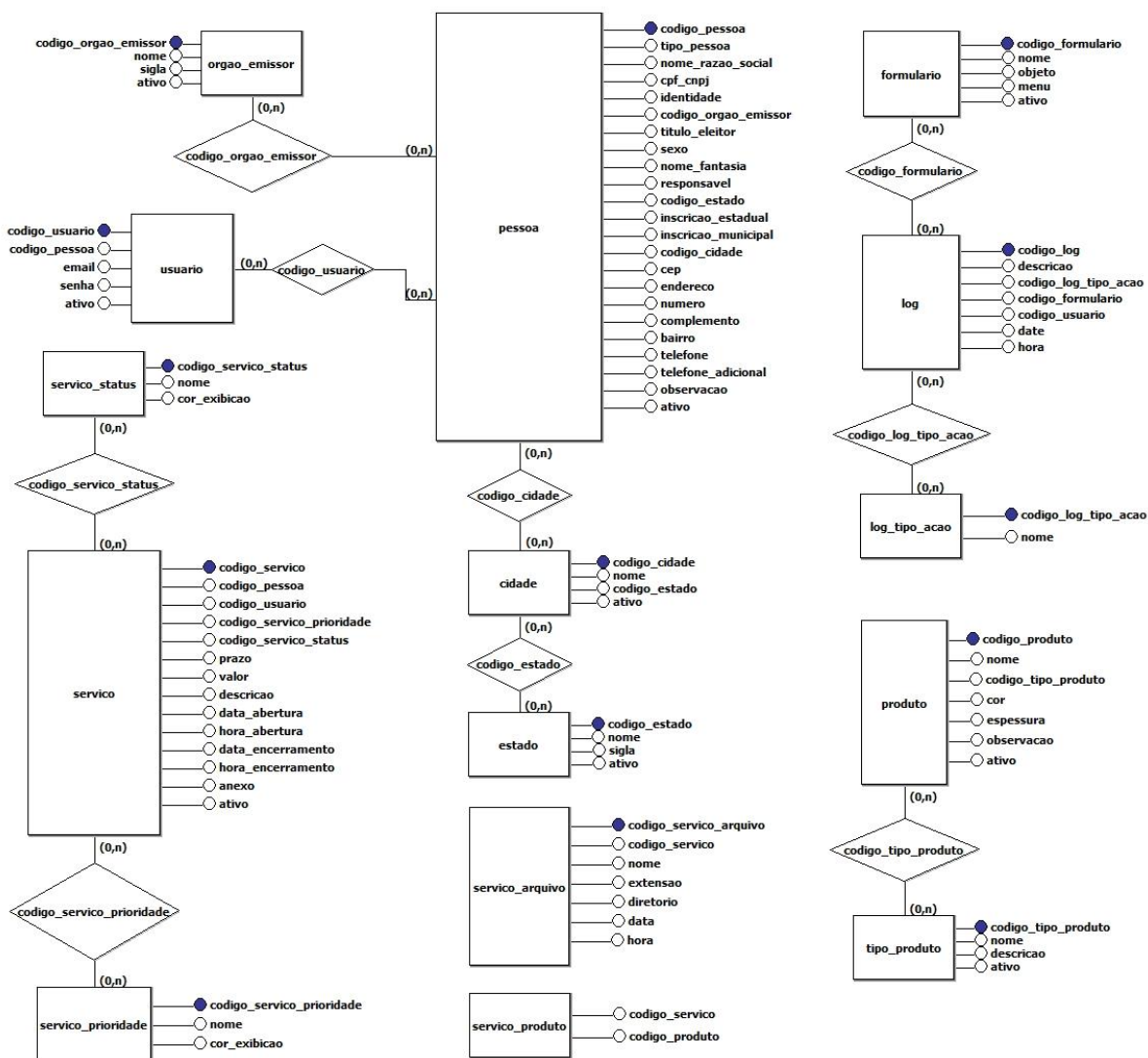


Figura 1-Diagrama ER (banco de dados)

### Quadro - Explica o Diagrama ER do banco de dados

Entidades	Descrição das entidades e seus atributos
usuario	representa a conta do usuário no sistema, bem como, os atributos pelos quais é composta.
pessoa	representa os dados e atributos referentes as pessoas físicas/jurídicas que são usados para o cadastro no sistema.
cidade	representa os dados e atributos referentes as cidades.
orgao_emissor	representa os dados e atributos do órgãos de registro de pessoas.
estado	representa os dados e atributos referentes aos estados.
produto	representa os dados e atributos referentes aos produtos a serem cadastrados no sistema para o gerenciamento do estoque.
tipo_produto	representa os dados e atributos referentes aos tipos de produtos que podem ser cadastrados no sistema.

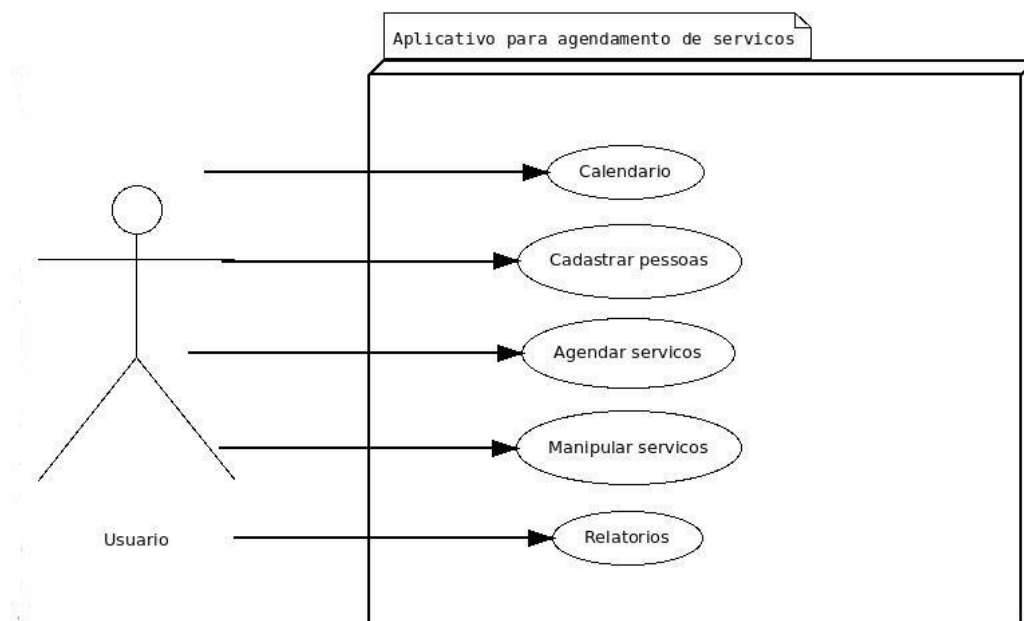


serviço	representa os serviços realizados pelo estabelecimento e seus dados.
servico_status	representa os status dos serviços como: pendente, em atendimento, concluídos, atrasados e cancelados.
servico_prioridade	representa a prioridade dos serviços como: baixa, media e alta.
servico_arquivo	representa o caminho onde os anexos estão salvos, vinculando o código do serviço com o código do arquivo.
servico_produto	representa a vinculação de todos os produtos que fazem parte de determinado serviço.
formulário	representa o cadastro de todos os formulários do sistema, cada um com seu código e caminho do arquivo.
log	representa o registro de todas as modificações feitas no sistema, tais como: salvar e editar.
log_tipo_acao	representa todas as ações que o usuário pode fazer no sistema.

## 4.2 Diagramas UML

Os diagramas UML compreenderam a parte da modelagem do aplicativo, fornecendo plantas aos desenvolvedores neste projeto, que permitiram que se estabelecesse uma base do funcionamento do aplicativo.

O diagrama de casos de uso foi necessário para termos uma compreensão básica das funcionalidades necessárias no aplicativo. Apesar de serem simples e minimalistas eles fornecem uma forte base para o início do desenvolvimento.



**Figura 2** – Diagrama de Casos de Uso

O usuário (ator) pode acompanhar o andamento dos serviços agendados pelo seu estabelecimento, cadastrar seus clientes (físicos e jurídicos), agendar novos serviços, manipular os serviços em andamento e também ter acesso a relatórios dos serviços.

Através do diagrama de classes foi possível detalhar melhor o funcionamento do aplicativo, pois possui um maior nível de detalhamento das funcionalidades do aplicativo. Este diagrama serve como uma planta que os objetos devem seguir, sendo um dos mais importantes para projetos elaborados em uma linguagem de programação orientada a objetos.

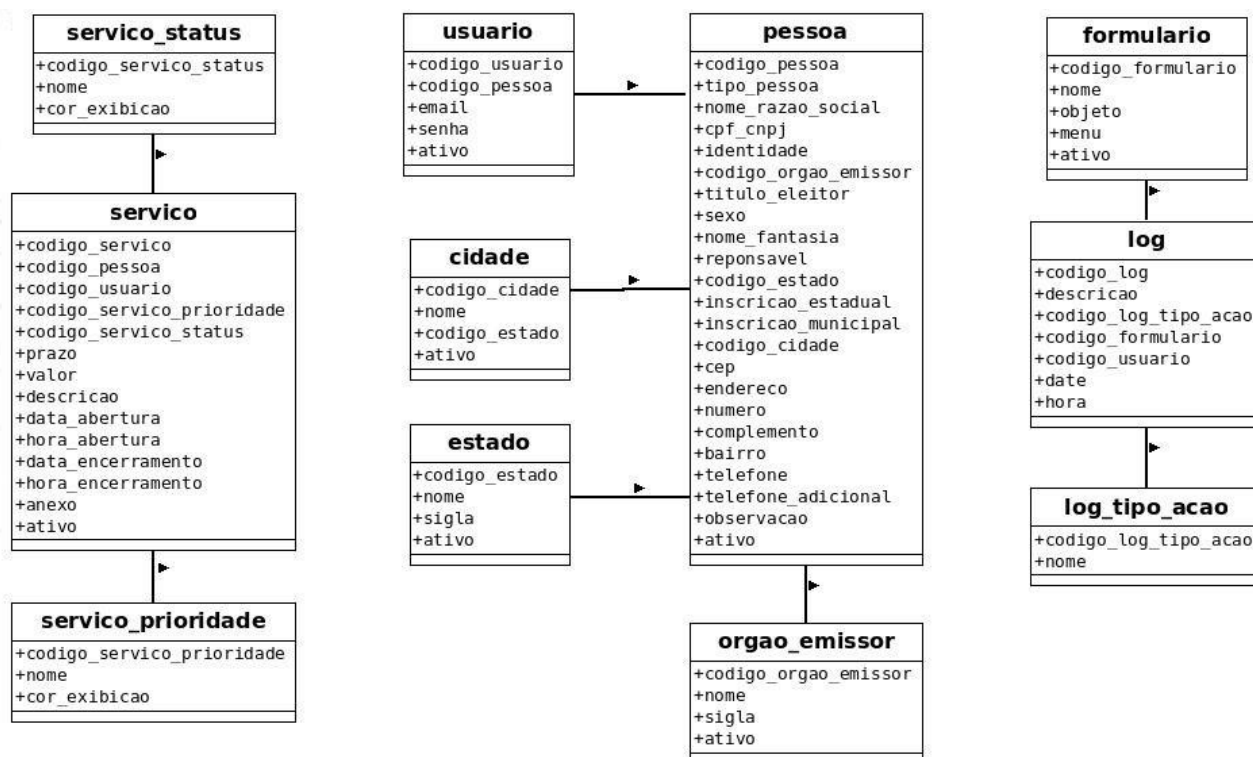


Figura 3 – Diagrama de classe

Podemos observar na figura 3 as classes e seus atributos, bem como, o relacionamento entre elas. A classe servico e as subclasses servico\_status e servico\_prioridade compreendem a funcionalidade de agendamento e manipulação dos serviços.

A classe pessoa compreende ao cadastro das pessoas (físicas e jurídicas) do aplicativo e se relaciona com as classes: usuario que compreende o cadastro necessário para a utilização do aplicativo, cidade que compreende ao cadastro das cidades no aplicativo, estado que apresenta os estados cadastrados no aplicativo e orgao\_emissor que contém os órgãos que realizam o registro e emissão das identidades das pessoas.

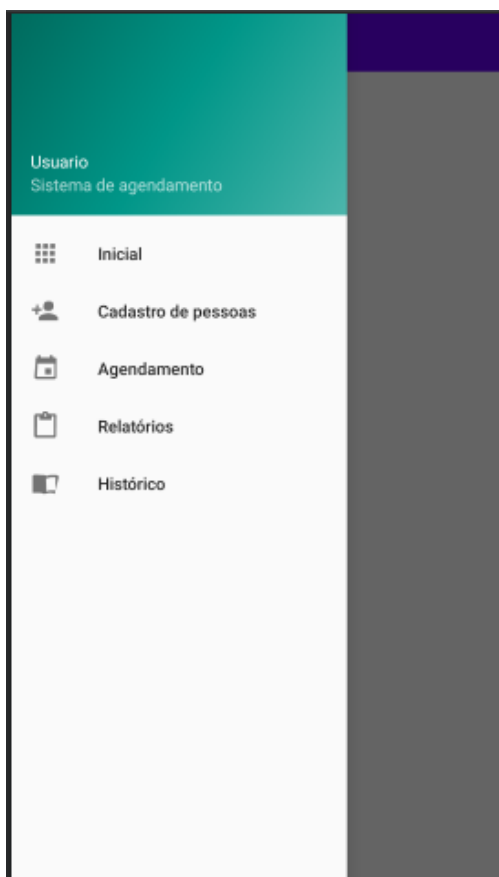
A classe formulario e as subclasses log e log\_tipo\_acao compreendem as funcionalidades relacionadas aos formulários do aplicativo.



The screenshot shows a login interface with two input fields: 'Usuário' (Username) and 'Senha' (Password). Below these fields is a button labeled 'ACESSAR' (Access). A mouse cursor is visible over the button.

**Figura 4** – Tela de login

A tela de login da aplicação irá solicitar o nome de usuário e senha. Importante ressaltar que a falta do botão de cadastro na tela deve-se ao fato dos cadastros dos usuários serem feitos somente pelos administradores, desenvolvedores, do aplicativo, seguindo o modelo do sistema web desenvolvido anteriormente.



**Figura 5** – Menu inicial

O menu do tipo *Navigation Drawer* pode ser acessado por meio de um botão no canto superior esquerdo, que permite que o menu seja expandido para que seja possível acessar as funcionalidades do aplicativo, como: cadastro de pessoas, agendamento (de serviços), relatórios e também histórico dos serviços prestados.

## 5. CONCLUSÃO

Com a finalidade de permitir uma maior portabilidade às informações de uma empresa, aumentando a eficiência e agilidade no desenvolvimento de suas atividades rotineiras, foi desenvolvido um aplicativo móvel para dispositivos android, com o objetivo de ser integrado a um sistema web anteriormente desenvolvido.

O aplicativo móvel permitirá o acesso as informações já inseridas no sistema da empresa, e também fornecerá a possibilidade de cadastrar novas informações, desde que o dispositivo possua acesso à internet. O acesso as informações já inseridas anteriormente podem ser realizadas de maneira off-line.

Os aplicativos móveis permitem uma maior eficiência e agilidade ao acesso e inserção de informações, podendo ser realizado em tempo real e em qualquer lugar, até mesmo dentro

da própria empresa, sem necessitar de um computador fixo, facilitando o desempenho e acesso das atividades realizadas na empresa.

Um aplicativo móvel integrado a um sistema web possibilita um controle mais amplo e versátil para os seus usuários, permitindo que as atividades sejam todas registradas no momento de sua ocorrência, aumentando, deste modo, as possibilidades de gestão das empresas, visto que se torna possível ter acesso às informações de uma maneira mais eficiente.

O trabalho teve o objetivo de criar um aplicativo móvel para o cadastro de clientes e agendamento de serviços, e este funcionará integrado a um sistema web previamente desenvolvido, que além de permitir acesso as informações já armazenadas no sistema, possibilitará que sejam inseridas novas informações e detalhamentos de serviços.

O aplicativo está sendo implementado, seguindo o paradigma de orientação a objetos, em linguagem Java e utilizando o Android Studio que se trata de uma IDE para desenvolvimento de aplicações para dispositivos android. Foram elaborados também diagramas UML como: diagrama de casos de uso e diagrama de classes, que serviram de base para a implementação do aplicativo. Foi utilizado o banco de dados, desenvolvido em PostgreSQL, do sistema web, que teve como modelo um diagrama ER desenvolvido durante a fase do projeto do sistema.

O desenvolvimento e integração do banco de dados do aplicativo terá continuidade posteriormente, pois frente ao momento crítico que vivenciamos devido a pandemia do COVID-19, tivemos como consequência imediata e obrigatória o isolamento social, tendo este como foco principal, impedir aglomerações para evitar a propagação do vírus. Devido a real necessidade de isolamento houve a suspensão de todas as atividades acadêmicas presenciais, o que acarretou prejuízos que dificultaram o desenvolvimento prático deste projeto.

## REFERÊNCIAS BIBLIOGRÁFICAS

BASTABLES, Susan B., **O Enfermeiro como Educador**. 3ª ed. Porto Alegre/RS: Armend, 2010.

BITTENCOURT, Rogério Gonçalves. **Aspectos Básicos de Banco de Dados**. 1ª ed. Florianópolis/SC, 2004.

CASTELLS, Manuel; ESPANHA, Rita. **A era da informação: economia, sociedade cultura**. Paz e Terra, 1999.

DA SILVA, Michel Lourenço; PEREIRA, Lúcio Camilo Oliva. **Android para Desenvolvedores**. 1ed. Rio de Janeiro: Brasport, 2009.

DWIVEDI, Chris |Clark, D.T.H. **Mobile Application Security**. 1ºed. McGraw-Hill Companies, 2010.

FAST -GROWTH, nations clock up the most hours mobile web usage. 2015. Disponível em: <http://www.globalwebindex.net/blog/fast-growth-nations-clock-up-the-most-hours-for-mobile-web-usage>. Acesso em: 03 mai. 2020

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 4ªed. Porto Alegre/RS: Editora Sagra Luzzato, 1998.

MATEUS, Geraldo R. ; LOUREIRO, Antonio A.F. **Introdução a Computação Móvel**. 2ª ed. Rio de Janeiro: DCC/UFGM, 1998.

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. São Paulo: MCGRAW-Hill, 2006.

RICARTE, Ivan Luiz Marques. **Programação Orientada a Objetos: Uma Abordagem com Java**. Campinas/SP, 2001.

SCIGLIANO, Felipe; SOUZA, Cassemiro; ROSA, Suéli Rodrigues. **Conceitos de Programação Orientada a Objetos**. São Paulo, 2018.