

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**Faculdade de Tecnologia Baixada Santista
Rubens Lara**

**Curso Superior de Tecnologia em
Sistemas para Internet**

**Alberto Marques Nunes
Rosangela da Silva Santana Borges
Tamires Yurie Mori**

**MEU CONDOMÍNIO: APLICATIVO ANDROID PARA
CONTROLAR ENTRADAS E SAÍDAS NA PORTARIA DE
CONDOMÍNIO**

**Santos, SP
2023**

**Alberto Marques Nunes
Rosângela da Silva Santana Borges
Tamires Yurie Mori**

**MEU CONDOMÍNIO: APLICATIVO ANDROID PARA
CONTROLAR ENTRADAS E SAÍDAS NA PORTARIA DE
CONDOMÍNIO**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientadora: Profa.Ma. Rosana Cammarosano

**Santos, SP
2023**

RESUMO

A tecnologia da informação possibilita a criação de soluções, para o cotidiano das pessoas, permite o desenvolvimento de sistemas que satisfaçam necessidades específicas. O presente trabalho buscou soluções de problemas relacionados a um condomínio residencial, estudado para implementação em quaisquer portarias. Identificadas algumas deficiências em controle de entrada de pessoal visitante, prestadores de serviços e entregadores *delivery*, foi proposto um aplicativo que atendesse as necessidades, com a aplicação dos conceitos e técnicas aprendidas no curso de Sistemas para *Internet*.

Com a criação de um aplicativo cliente-servidor, com banco de dados hospedado em nuvem que: 1) permite a integração das informações; 2) possibilita um controle na área de recepção; 3) autoriza a entrada de pessoas nas dependências do condomínio; 4) manutenção de cadastro de visitantes, com informações do horário de entrada e saída diminuindo e desencorajando a incidência de sinistros, como vandalismo e até assaltos ou arrastões tão em voga nos tempos atuais. Os resultados obtidos demonstram que o aplicativo traz organização, controle das rotinas do condomínio, dando suporte para as principais atividades.

Palavras-chave: Aplicativo Móvel. *Flutter*. Segurança Condominial. Controle. *Android*.

ABSTRACT

Information technology enables the creation of solutions for people's daily lives, and allows the development of systems that meet specific needs. This study sought solutions to problems related to a residential condominium, as well as studied for implementation in any entrance. Some deficiencies were identified in the control of the entry of visitors, service providers and couriers. An application was proposed that would respond to the needs, with the application of the concepts and techniques learned in the Internet Systems course.

With the creation of a client-server application, with a database hosted in the cloud that: 1) allows the integration of information; 2) allows control in the reception area; 3) authorizes people to enter the condominium premises; 4) maintains the register of visitors, with information on entry and exit times, reducing and discouraging the incidence of accidents, such as vandalism and even thefts or trawling, so in vogue these days. The results obtained demonstrate that the application brings organization, control of the condominium routines, helping in the main activities.

Keywords: Application. Flutter. Condominium Security. Control. Android.

LISTA DE FIGURAS

Figura 1 - Página inicial do aplicativo <i>Shielder Portaria Online</i>	17
Figura 2 - Página Inicial do aplicativo SIGAi – Visitantes.....	18
Figura 3 - Página Inicial do aplicativo <i>iPort Portaria</i>	18
Figura 4 - Diagrama de Caso de Uso do Meu Condomínio.....	23
Figura 5 - Fluxo de Eventos do aplicativo Meu Condomínio.	28
Figura 6 - Fluxo de Eventos do aplicativo Meu Condomínio.	28
Figura 7 - Telas de <i>login</i> e solicitação de cadastro.....	29
Figura 8 - Confirmação da solicitação e tela principal (<i>home</i>).	30
Figura 9 - Cadastro de visitante e Confirmação de cadastro de visitante.....	30
Figura 10 - Consulta de visitante.	31
Figura 11 - <i>Wireframe</i> da tela de saída antes da alteração.	35
Figura 12 - <i>Wireframe</i> da tela de saída antes da alteração.	36
Figura 13 - Páginas de <i>login</i> e cadastro.	37
Figura 14 - Telas de cadastro de usuário e consulta.	38
Figura 15 - Telas de Cadastro, Consulta e Saída.....	38
Figura 16 - Limpar Memória do aplicativo.....	39

LISTA DE TABELAS

Tabela 1 - Requisitos funcionais do sistema Meu Condomínio.	22
Tabela 2 - Requisitos não funcionais do sistema Meu Condomínio.	22
Tabela 3 - Descrição dos requisitos de negócio.....	22

LISTA DE ABREVIATURAS E SIGLAS

ARM - <i>ADVANCED RISC MACHINE</i>	18
CSS - <i>CASCADE STYLE SHEETS</i>	18
HTML - <i>HYPertext MARKUP LANGUAGE</i>	18
JS - <i>JAVASCRIPT</i>	18
SGBD - <i>SISTEMA GERENCIADOR DE BANCO DE DADOS</i>	25
SQL - <i>STRUCTURED QUERY LANGUAGE</i>	24
UI - <i>USER INTERFACE</i>	20
UML - <i>UNIFIED MODELING LANGUAGE</i>	22

LISTA DE GRÁFICOS

Gráfico 1 - Resposta dos usuários à primeira pergunta.....	33
Gráfico 2 - Resposta dos usuários à segunda pergunta.	32
Gráfico 3 - Resposta dos usuários à terceira pergunta.....	33
Gráfico 4 - Resposta dos usuários à quarta pergunta.	33
Gráfico 5 - Resposta dos usuários à quinta pergunta.....	34
Gráfico 6 - Tempo de iteração dos usuários.....	34

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Objetivo	16
1.1.1 Objetivo Geral.....	16
1.1.2 Objetivos Específicos	17
1.2 Estado da Arte.....	17
1.2.1 Aplicativos existentes no mercado.....	18
1.2.1.1 <i>Shielder</i> Portaria Online	18
1.2.1.2 SIGAi – Visitantes	18
1.2.1.3 <i>iPort</i> Portaria	19
1.3 Análise do projeto.....	19
2 DESENVOLVIMENTO	20
2.1 Análise do Projeto.....	20
2.1.1 <i>Flutter</i>	20
2.1.2 Linguagem de programação <i>Dart</i>	20
2.1.3 Programação Orientada a Objeto.....	21
2.1.4 <i>XML</i>	22
2.2 Análise de Requisitos	22
2.3 Diagrama de caso de Uso.....	23
2.4 Banco de Dados	24
2.5 Banco de Dados Relacional	25
2.6 Linguagem <i>SQL</i>	25
2.7 Sistema Gerenciador de Banco de Dados.....	26
2.8 <i>MySQL</i>	26
2.9 Computação em Nuvem	26
2.10 Engenharia de <i>Software</i>	27
2.11 Requisitos.....	27
2.12 Desenvolvimento Ágil.....	27
2.13 <i>Scrum</i>	28
2.14 Camada de Negócio	28
2.15 <i>Wireframe</i> do Projeto	29
3. RESULTADO	32
3.1 Testes de usabilidade.....	33

3.1.1 Resposta dos testes de usabilidade	36
3.2 Telas do sistema	38
3.2.1 <i>Mockup</i> das telas	39
3.3 Conclusões do teste de usabilidade.....	42
4. Considerações finais	42
REFERÊNCIAS	43

1 INTRODUÇÃO

Para que uma empresa consiga se manter no mercado, deve explorar as oportunidades e buscar se modernizar, a fim de obter vantagem competitiva e conseguir atender e satisfazer as necessidades de seus clientes.

Atualmente, as tecnologias da informação são vistas como ferramentas importantes e estratégicas para as empresas, pois servem de apoio no controle de negócios e auxiliam nas tomadas de decisões dos gestores.

Apesar da tecnologia estar presente no cotidiano das organizações, muitas das pequenas empresas não a utilizam como ferramenta para otimizar seus processos de negócios ou controlar suas informações, sendo o controle muitas vezes feito, por exemplo, na base de agendas e cadernos de papel.

Condomínios residenciais também podem ser analisados nesse cenário, pois alguns utilizam tecnologias e outros não. Tendo como base um condomínio localizado no Estado de São Paulo, na região da Baixada Santista com quinze andares e com quarenta apartamentos no Bairro do Embaré, pôde-se identificar possíveis funções que o sistema teria para aperfeiçoar os processos de negócios a baixo custo.

Uma vantagem de ter um sistema informatizado é, por exemplo, possibilitar a manutenção do cadastro de seus condôminos e visitantes, tornando rápida a localização de dados relevantes quando necessário.

Quando se trata de visitante, outro ponto essencial é o controle de quem está adentrando no condomínio e o que foi fazer - garantindo que a entrada do mesmo não seja de modo de engenharia social, do tipo *delivery* para uma unidade que não tenha pedido nada, ou um representante de uma prestadora de serviços de telefonia, abastecimento de água, dentre outras mais - tirando uma foto do visitante na apresentação na portaria do condomínio. Se for um falso representante, este será reprimido, pois nenhuma pessoa que venha fazer algum assalto ou arrastão vai deixar-se ser fotografado, e se assim for, a foto estará armazenada em nuvem, possibilitando a visualização em qualquer lugar mesmo que os equipamentos como câmeras, celulares ou computadores sejam furtados ou roubados, evitando situações que possam abalar a confiança do cliente na empresa prestadora.

No caso de um vandalismo dentro do condomínio, um sistema informatizado permitiria o registro e controle dos horários de entrada e saída dos visitantes, atualizaria o status e observações, proporcionaria assim um controle em tempo real, para que o gestor do condomínio identifique o autor mais facilmente, em conjunto com as câmeras que hoje são tão utilizadas e que por diversas vezes inúteis, para identificar exatamente quem foi o autor, pois são muitas horas de gravação a serem assistidas para a tomada de decisões de acordo com as informações disponíveis.

Mañas (2011, p. 1) explica que “trabalhando a informação de forma correta, a empresa consegue direcionar seus colaboradores para atender o cliente da melhor maneira possível”.

1.1 Objetivo

Nesse item serão abordados o objetivo geral, bem como, os objetivos específicos desse trabalho.

1.1.1 Objetivo Geral

Desenvolver um sistema que propicie ao condomínio ter um controle de cadastro de seus visitantes, fornecedores de serviços, além de permitir a gestão de tráfego dentro do condomínio com horários de entrada e saída, e que tenha o banco de dados hospedado na nuvem para acesso integrado e remoto de seus usuários.

Possibilidades do aplicativo: a) a identificação dos visitantes, com hora de entrada e hora de saída do condomínio; b) persistência de dados a serem consultados posteriormente sem depender da capacidade de armazenamento do *Smartphone* (telefone celular com funções de um computador, acesso à *internet* e um sistema operacional capaz de executar aplicativos baixados) utilizado ou de um possível furto do mesmo numa situação de sinistro; c) gestão mais organizada e segura dos dados coletados e facilmente consultados posteriormente; d) o aplicativo permite com o número de identificação do visitante seja recuperada a foto para conferência no momento da entrada, possibilitando assim a reentrada do visitante somente por conferência e autorização do condômino e registrada a entrada no aplicativo de forma

informatizada, contendo funções que abordem os processos citados anteriormente; e) diminui a dependência de papéis e controles manuais, centralizando as informações e disponibilizando de forma ágil aos colaboradores envolvidos, trazendo velocidade nas respostas aos condôminos.

1.1.2 Objetivos Específicos

- Conhecer sobre as regras de negócio do condomínio e tecnologias para desenvolvimento do sistema;
- Definir o processo de engenharia de *software* a ser aplicado, com o modelo identidade-relacional de dados;
- Desenvolver o sistema codificando as classes, banco de dados e interface e testes de entrada e saída de dados no sistema.

1.2 Estado da Arte

Este aplicativo foi encomendado sob medida, e na época que foi desenvolvido não existia algo similar no *Google App Store* (loja de aplicativos criada pelo *Google*, onde se encontram jogos, filmes, música, livros e muito mais), que fizesse tal controle. A primeira versão foi desenvolvida com o extinto *Phonegap* da *Adobe*, um *framework* (uma abstração que une códigos comuns entre vários projetos de *software*) de desenvolvimento móvel que permitia aos desenvolvedores de *software* criar aplicativos para dispositivos móveis usando apenas as tecnologias *JS* (*JavaScript*, uma linguagem de programação interpretada estruturada, de *script* em alto nível), *HTML 5* (Linguagem de Marcação de HiperTexto) e *CSS 3* (*Cascading Style Sheets* ou Folhas de Estilo em Cascata) e nesta versão em *Flutter* (um *SDK -Software Development Kit*, ou *kit* de desenvolvimento de *software*) foi aperfeiçoada.

A aceitação dos usuários está ótima, já existem mais de 50 usuários em todo o país, sendo utilizada não somente em condomínios, mas também em empresas para registro dos visitantes no Rio de Janeiro e Manaus.

Comparando os sistemas que existem no *Google App Store* atualmente, nota-se diversos aplicativos para condomínios, porém em sua maioria não é voltado a portaria, mas para comunicação de vencimentos, gastos e outras relações entre o condômino, administrador e síndico.

1.2.1 Aplicativos existentes no mercado

Através da pesquisa na *Google App Store* foram localizados alguns produtos com características similares ao projeto proposto neste trabalho, seguem alguns exemplos:

1.2.1.1 *Shielder Portaria Online*

A Figura 1 mostra o aplicativo *Shielder Portaria Online*, lançado no segundo semestre de 2019. As fotos dos visitantes são armazenadas, as entradas e saídas são lançadas no sistema e o condômino autoriza pelo aplicativo.

Existem muitas reclamações na revisão do aplicativo, alguns comentários relatam a dificuldade em parametrizar, a baixa performance quando há vários condomínios, a demora na abertura do aplicativo e atrasos nas mensagens de aviso.

O sistema também tem a nuvem como destino dos dados, porém as fotos ficam armazenadas no celular do porteiro, dificultando um pouco na otimização do espaço de arquivos.

Figura 1 - Página inicial do aplicativo *Shielder Portaria Online*.



Fonte: *Google Play Store*, 2023

1.2.1.2 SIGAi – Visitantes

Este aplicativo foi lançado em março de 2021 e tem como característica o registro de entrada dos visitantes com a data e a hora, porém não tem funcionalidade de tirar foto.

Apesar de ser um aplicativo lançado recentemente tem menos de vinte celulares instalados e não existe nenhum tipo de avaliação.

Assim como o *Shielder*, também faz parte de um sistema integrado em nuvem.

A Figura 2 é a página inicial do aplicativo.

Figura 2 – Página Inicial do aplicativo SIGAi – Visitantes.



Fonte: Google Play Store, 2023

1.2.1.3 *iPort* Portaria

No *iPort* Portaria há uma integração com outro sistema que faz a gestão do condomínio *on-line*. Possui as mesmas características do *Shielder* - mencionado anteriormente no item 1.2.1 - porém sem os problemas do primeiro e sem avaliações negativas.

A Figura 3 é a página inicial do aplicativo.

Figura 3 - Página Inicial do aplicativo *iPort* Portaria.



Fonte: Google Play Store, 2023

1.3 Análise do projeto

Comparando os três aplicativos mencionados com o projeto do grupo, conclui-se que, apesar de ser um aplicativo de simples de controle de visitantes, tem-se algumas vantagens, como:

- Tirar a foto do visitante;
- A saída não precisa ser registrada pela procura pelo nome, a foto do visitante fica em evidência, possibilitando a fácil localização na lista;
- Com apenas dois cliques a saída é registrada.

2 DESENVOLVIMENTO

2.1 Análise do Projeto

Neste capítulo são apresentadas as tecnologias utilizadas durante o desenvolvimento do aplicativo Meu Condomínio. Para criar as telas foi utilizado o *framework Flutter*, e nas funcionalidades do aplicativo foi utilizada a linguagem de programação *Dart* (uma linguagem de *script* voltada à *web* desenvolvida pela *Google*), juntamente com banco de dados *Mysql* (um sistema de gerenciamento de banco de dados, que utiliza a linguagem *SQL*) como *interface* (nome dado para o modo como ocorre a “comunicação” entre duas partes distintas e que não podem se conectar diretamente).

2.1.1 Flutter

O *Flutter* é usado no desenvolvimento de aplicações multiplataforma a partir de um único código, com o objetivo de possibilitar a criação de interface nativa e de alta performance (*FLUTTER*, 2019).

Segundo Abranches (2018), no *Flutter* o código é compilado em *ARM* (*Advanced RISC Machine*), ou seja, compila o código para cada plataforma agilizando a abertura e o desempenho do aplicativo. Ainda segundo o autor, no *Flutter* não existe distinção de arquivos para a personalização do *layout* (*JS*, *HTML*, *CSS*) e tudo é definido em um *Widget Flutter* (cada elemento numa tela do aplicativo), facilitando e aumentando a velocidade de desenvolvimento da aplicação.

Os aplicativos desenvolvidos no *framework Flutter* funcionam semelhante a um aplicativo nativo, sendo mais rápidos em comparação com *apps* desenvolvidos em outros *frameworks* multiplataforma.

Além dos benefícios de *performance* nativa, o *Flutter* possui muita flexibilidade para a criação de *interfaces* personalizadas, animações e facilidades em acessar os recursos do aparelho (PEREIRA, 2019). É um *framework* de fácil instalação e configuração, ao contrário de seu principal concorrente *React-Native* (uma biblioteca *Javascript* criada pelo *Facebook*, usada para desenvolver aplicativos para os sistemas *Android* e *iOS* de forma

nativa) - além de produzir menos código, pois o desenvolvimento é feito com um único código na linguagem *Dart*, poupando tempo e trabalho para os desenvolvedores (CAPPELLI, 2018).

2.1.2 Linguagem de programação *Dart*

A linguagem de programação *Dart* foi apresentada pelo *Google* em 2011 com objetivo de substituir o *JavaScript* como principal linguagem embutida nos navegadores (ANDRADE, 2019). Porém, a linguagem evoluiu com o passar dos anos e hoje pode ser considerada uma linguagem multiparadigma, embora a linguagem apresente fortes estruturas típicas de linguagens orientadas a objeto (GUEDES, 2019). A linguagem *Dart* foi desenvolvida para funcionar em todas as plataformas e ter maturidade elevada para desenvolvimento de *interfaces* de usuário.

Suas principais características são:

- Otimização para *UI (User Interface)*: visto que ela foi escrita com foco na otimização da *interface* de usuário;
- Desenvolvimento Produtivo: *Hot reload* é uma ferramenta muito importante para aumentar a produtividade; e
- Rápida em todas as plataformas: compila para código de máquina *ARM* e *x64* para celular, *desktop* e *backend*. Além disso compila para *JavaScript* em sistemas *web* (PEREIRA, 2019).

2.1.3 Programação Orientada a Objeto

A programação orientada a objetos é um paradigma de programação que propõe a abstração dos dados em classes que buscam representar objetos do mundo real. Essas classes possuem atributos, que são a representação das características e estados dos objetos, e métodos, que são comportamentos que modificam e permitem a interação desses atributos.

Segundo Lima e Reis (2002), ao contrário do paradigma de programação procedural, predecessor da orientação a objetos, em que concebiam os dados e códigos como elementos separados, a orientação a objeto propõe justamente a

ideia de eliminar a distância entre dados e código e fazer com que ambos ficassem interligados.

Os autores ainda afirmam que, para ser considerada inteiramente orientada a objeto, uma linguagem deve comportar três mecanismos básicos:

- Encapsulamento, a proteção dos atributos das classes, permitindo sua interação somente por meio dos métodos que a classe possui, tornando-os inacessíveis fora do contexto daquela classe;

- Herança, uma relação entre classes onde a classe derivada herda atributos e/ou comportamentos da classe pai;

- Polimorfismo, propriedade de uma classe derivada de outra de apresentar comportamento diferente para métodos herdados.

2.1.4 XML

A *eXtended Markup Language*, ou *XML*, é uma linguagem de marcação que facilita o entendimento do documento por humanos e máquinas.

Segundo Heitlinger (2001, p. 16), “A *Extensible Markup Language (XML)* provê um conceito para descrever, armazenar, intercambiar e manipular dados estruturados”, permitindo a “partilha de informações entre aplicações baseadas em sistemas incompatíveis” por meio da “*Internet*, por *Intranets*, ou qualquer outro modo de transporte” de dados, sendo os dados descritos em formato de texto, se tornando um “formato universal para a partilha de dados entre aplicações”.

2.2 Análise de Requisitos

Tabela 1 - Requisitos funcionais do sistema Meu Condomínio.

[RF01] O sistema deve permitir que o usuário faça o cadastro solicitando nome, e-mail e telefone.
[RF02] O sistema gera uma senha simples que será passada ao usuário
[RF03] O usuário faz o <i>login</i> com o nome do condomínio e a senha

[RF04] O sistema deve permitir que o usuário realize o cadastro dos visitantes com os seguintes dados: nome, CPF, veículo (se necessário), aonde irá, o motivo da visita (quem autorizou) e anexar uma foto do visitante.
[RF05] O sistema deve mostrar os visitantes que estão no condomínio.
[RF06] O sistema deve permitir que o usuário controle a saída os visitantes, permitir a visualização da foto do visitante.

Fonte: Os autores (2023)

Tabela 2 - Requisitos não funcionais do sistema Meu Condomínio.

[RNF01] O sistema necessita de <i>internet</i> de pelo menos 5mbs

Fonte: Os autores (2023)

Tabela 3 - Descrição dos requisitos de negócio.

[RN01] O sistema deverá operar por 24 horas por dia 7 dias por semana.
[RN02] O cadastro que será autorizado pelo administrador por 30 dias.
[RN03] O sistema não deve permitir login que não esteja liberado.
[RN04] Depois da autorização do administrador serão solicitados o nome do condômino e a função que ele exerce.

Fonte: Os autores (2023)

2.3 Diagrama de caso de Uso

Para Guedes (2011, p. 30), o diagrama de Caso de Uso é o mais geral e informal da *UML* (Linguagem de Modelagem Unificada), sendo utilizado geralmente no levantamento e análise de requisitos.

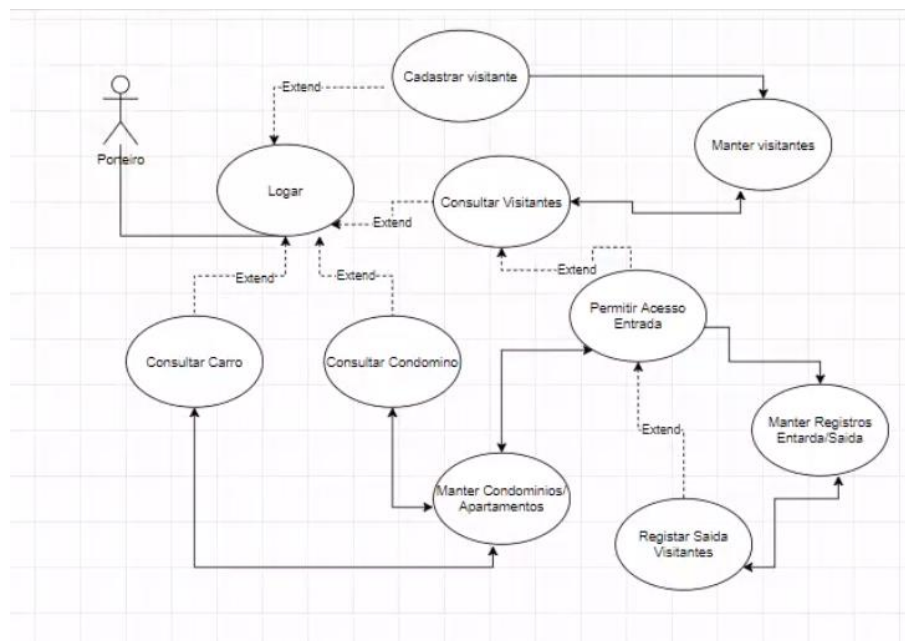
O diagrama de Caso de Uso é aplicado para a visualização do relacionamento do sistema com os atores que interagem com ele. Para os usuários, permite a compreensão de como utilizar o sistema, e para o desenvolvedor, como o *software* será implementado. (BOOCH; RUMBAUGH; JACOBSON, 2000).

A Figura 4 representa o diagrama de caso de uso do aplicativo, onde tem o login principal, quando logado aparecem as opções citadas, se o visitante já estiver cadastrado aparece a foto, após informar o número do apartamento, o

morador autoriza sua entrada e tem o menu da saída do visitante, caso não tenha registro, tem a opção de cadastramento.

Para a sequência da construção do aplicativo tem o menu de consulta de moradores fixos e de temporada, onde há o controle dos proprietários com seus respectivos veículos, onde o porteiro pode controlar a entrada para pessoas sem controle remoto a partir desses cadastramentos.

Figura 4 – Diagrama de Caso de Uso do Meu Condomínio.



Fonte: Os autores (2023)

2.4 Banco de Dados

As tecnologias relacionadas ao banco de dados surgem como alternativas quando há a necessidade de guardar, recuperar e manipular dados. “Os dados são fatos que podem ser gravados e que possuem um significado implícito.” (ELMASRI; NAVATHE, 2006, p. 4).

Para Elmasri e Navathe (2006, p. 4), o banco de dados “é uma coleção de dados relacionados” e complementa que “possui algumas fontes das quais os dados são derivados, alguns níveis de interação com os eventos do mundo real e um público efetivamente interessado em seus conteúdos”.

Os dados precisam ser coletados e organizados de forma lógica e coerente, sendo projetado, construído e povoado por dados a fim de atender uma proposta específica. (ELMASRI; NAVATHE, 2006).

Date (1989) explica que o banco de dados permite às empresas um controle centralizado dos seus dados operacionais, o autor considera um dos ativos mais valiosos.

2.5 Banco de Dados Relacional

Segundo Korth, Silberschatz e Sudarshan (2006, p. 7), é “um tipo de banco de dados” que usa “um conjunto de tabelas para representar os dados e as relações entre eles”.

Ramakrishnan e Gehrke (2011) afirmam que algumas das vantagens do modelo relacional sobre os modelos mais antigos são a sua representação de dados simples e a facilidade com que consultas complexas podem ser expressas. Os autores também afirmam que o modelo relacional é o dominante no mercado, sendo a base dos sistemas gerenciadores de banco de dados líderes do mercado.

2.6 Linguagem SQL

A linguagem SQL, ou Linguagem de Consulta Estruturada em português, é considerada a linguagem padrão de banco de dados relacional. (Korth; Silberschatz; Sudarshan, 2006, p. 51).

Para Korth, Silberschatz e Sudarshan (2006, p. 51), embora seja comum se referir à linguagem SQL como “linguagem de consulta”, ela permite realizar funções além da consulta, como “definir a estrutura dos dados, modificar dados no banco de dados e especificar restrições de segurança”.

Os autores Elmasri e Navathe (2006) atribuem à razão do sucesso dos bancos de dados relacionais no mundo comercial à linguagem SQL, pois, pelo fato de os sistemas gerenciadores de banco de dados usarem o mesmo padrão de linguagem, a migração de um sistema para o outro não gera custo e tempo excessivos.

2.7 Sistema Gerenciador de Banco de Dados

Para que seja possível a criação e manutenção de um banco de dados, é difundido o uso de sistemas gerenciadores de banco de dados.

Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de banco de dados entre vários usuários e aplicações. (KORTH; SILBERSCHATZ; SUDARSHAN, 2006, p. 4).

Uma “característica fundamental” do sistema gerenciador de banco de dados é que não “possui não apenas o banco de dados, mas também uma completa definição ou descrição da estrutura desse banco de dados”. (Korth; Silberschatz; Sudarshan, 2006, p. 7).

2.8 MySQL

Suehring (2002) define o *MySQL* como um sistema gerenciador de banco de dados relacionais de código-fonte aberto, ou seja, livre para instalação e modificação de suas propriedades a fim de atender as próprias necessidades do usuário. O autor também destaca como vantagens do *MySQL* a estabilidade, suporte em várias plataformas e baixo custo.

2.9 Computação em Nuvem

A computação na nuvem é definida por Taurion (2009, p. 2) como “um ambiente de computação baseado em uma imensa rede de servidores, sejam estes virtuais ou físicos”, destacando como vantagem o uso de recursos de computadores independentes sem preocupação com localização física e sem investimentos em *hardware* (é a parte física do computador, ou seja, o conjunto de aparatos eletrônicos, peças e equipamentos que fazem o computador funcionar).

A computação na nuvem, apesar de não ser seu único propósito, é muito utilizada para a armazenagem de dados, ficando esses disponíveis de forma remota por meio da *internet*, o que proporciona, além da disponibilidade independentemente da localização física, a integração das informações geradas pelos diversos usuários.

2.10 Engenharia de Software

Sommerville (2004, p. 4) explica que a engenharia de *software* (é um serviço computacional utilizado para realizar ações nos sistemas de computadores) é uma disciplina “cuja meta é o desenvolvimento do sistema de *software* com boa relação custo-benefício”, abrangendo todos os aspectos da criação de um sistema, desde a especificação até a manutenção. O autor ainda comenta que seguir os conceitos de engenharia de *software* permite o desenvolvimento de forma sistemática e organizada, sendo a maneira mais eficaz de produzir um *software* de qualidade.

2.11 Requisitos

Fagundes (2011) explica que requisitos representam o escopo do projeto, sendo as especificações deste que será desenvolvido, simbolizando a necessidade do usuário que será atendida pelo sistema. É a partir dos requisitos que as funcionalidades do sistema são desenvolvidas.

Os requisitos podem ser funcionais, que definem as funções e o comportamento do sistema diante de situações ou ordens, e não funcionais, que definem outros aspectos relevantes para o sistema, como meios de exibição, performance, acessibilidade, restrições, entre outros. (FAGUNDES, 2011).

A identificação dos requisitos é fundamental para o desenvolvimento do projeto, pois é por meio deles que se define de forma clara o que o sistema atenderá, definindo o escopo do projeto.

2.12 Desenvolvimento Ágil

O desenvolvimento ágil é um método da engenharia de *software* que, segundo Maxim e Pressman (2016), prioriza a entrega mais do que a análise e o projeto, defendendo a satisfação do cliente e a entrega incremental antecipada, buscando a simplicidade no desenvolvimento geral, sendo também denominada pelo autor como “engenharia de *software* flexível”.

Ainda de acordo com os autores, o processo ágil deve ser adaptável, mas de modo incremental, ou seja, as adaptações devem promover a evolução do sistema, tendo como parâmetro o *feedback* do cliente.

2.13 Scrum

Scrum é uma metodologia de desenvolvimento ágil que, segundo Maxim e Pressman (2016), é eficaz em cenários de desenvolvimento de projetos com prazos curtos de entrega, requisitos mutáveis e urgência do negócio.

O método possui quatro atividades de desenvolvimento: análise, o projeto, a evolução e entrega. Os autores explicam que o *Scrum* conta com padrões de processo de *software* para a realização dessas atividades, sendo eles:

- *Backlog* – É uma lista de requisitos e funcionalidades do projeto que tenham valor para o cliente, organizada de acordo com a prioridade do cliente, podendo os itens da lista serem modificados a qualquer momento;

- *Sprints* – São atividades realizadas, com prazo determinado, para desenvolver os *backlogs* estabelecidos e atingir os requisitos. Durante o *sprint*, as especificações não devem ser ajustadas;

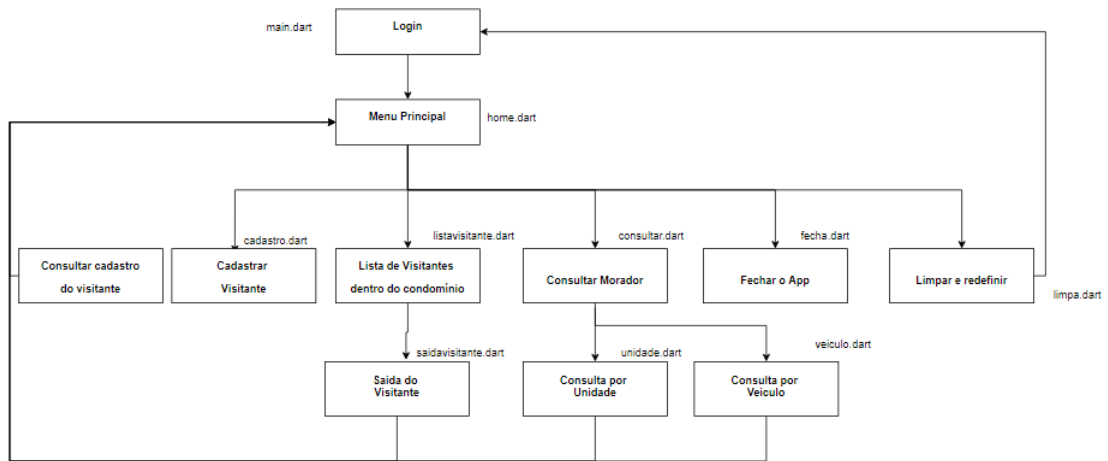
- Reuniões *Scrum* – São reuniões diárias de curta duração, organizadas pela equipe de desenvolvimento e seu líder, para verificar o que foi feito durante o *sprint*, quais dificuldades foram encontradas, e o que deve ser feito até a próxima reunião;

- Demos* – São entregas de partes, ou versões, do *software* para teste e avaliação do cliente, sendo elas entregues de forma incremental, ou seja, evoluindo o sistema conforme o próprio é desenvolvido.

2.14 Camada de Negócio

A Figura 5 é o fluxograma dos eventos do aplicativo referente a camada de negócio atual e construção futura da estrutura do aplicativo, onde a partir do *login*, vai para o menu principal com acesso as opções de consultas, cadastros, listagens e controle de entrada e saída dos visitantes.

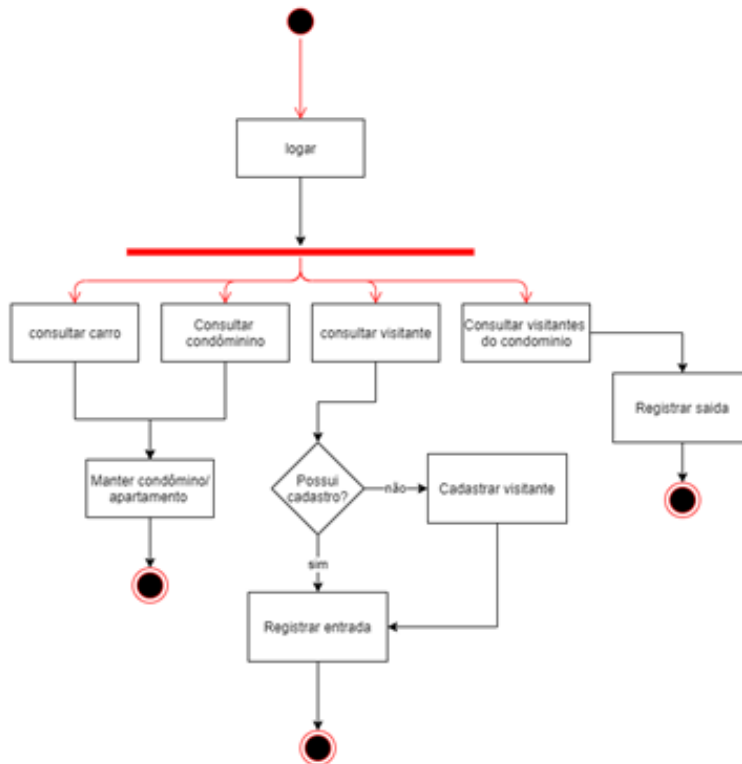
Figura 5– Fluxo de Eventos do aplicativo Meu Condomínio.



Fonte: Os autores (2023)

A Figura 6 são os fluxogramas dos eventos do aplicativo, onde após *login* há o menu de opções de consultas, registro de entrada e saída.

Figura 6 – Fluxo de Eventos do aplicativo Meu Condomínio.



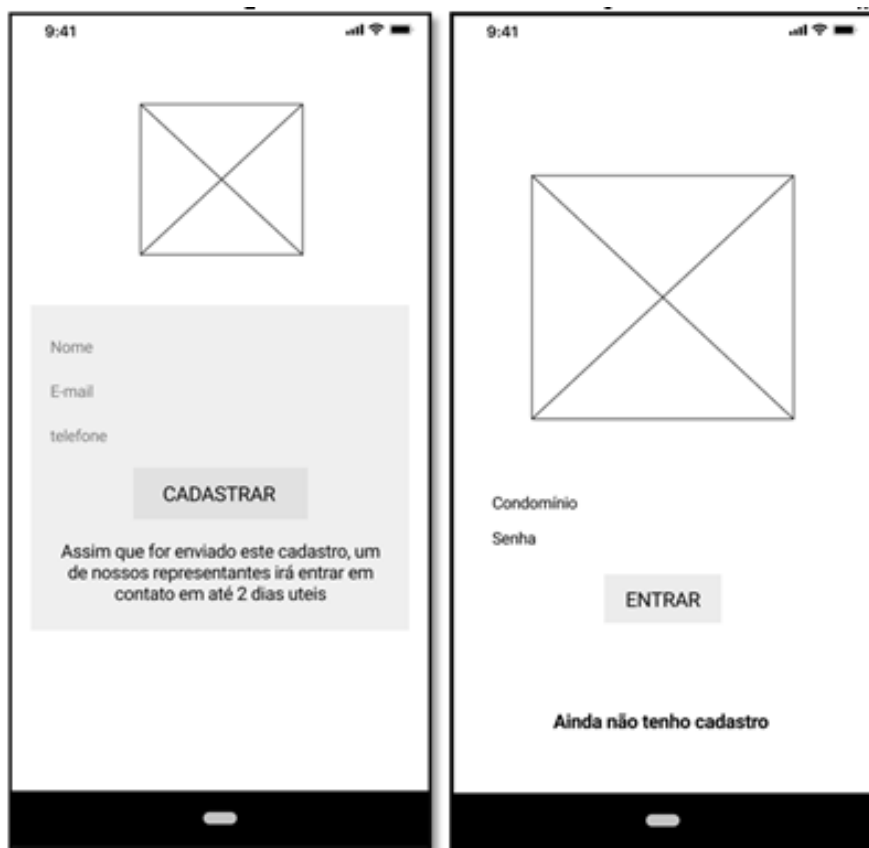
Fonte: Os autores (2023)

2.15 Wireframe do Projeto

O primeiro passo foi a elaboração das telas principais com suas características técnicas, através de um *wireframe* (diagrama visual que esboça a estrutura de uma tela de um site ou de aplicativo) e logo em seguida um *mockup* (um modelo em escala ou de tamanho real de um projeto ou dispositivo) Abaixo segue o resultado do *wireframe*, que deu origem às telas finais e ao teste com usuários, abordados no próximo capítulo.

A Figura 7 são as telas de *login* e solicitação de cadastro do aplicativo, onde o cliente registra os dados e aguarda confirmação. Se já cadastrado, efetua a entrada no aplicativo através do *login* e senha.

Figura 7 - Telas de *login* e solicitação de cadastro



Fonte: Os autores (2023)

A Figura 8 são as telas de confirmação da solicitação de cadastro e tela principal do aplicativo. Através da confirmação, o acesso a entrada do aplicativo permite a navegação do aplicativo com suas opções.

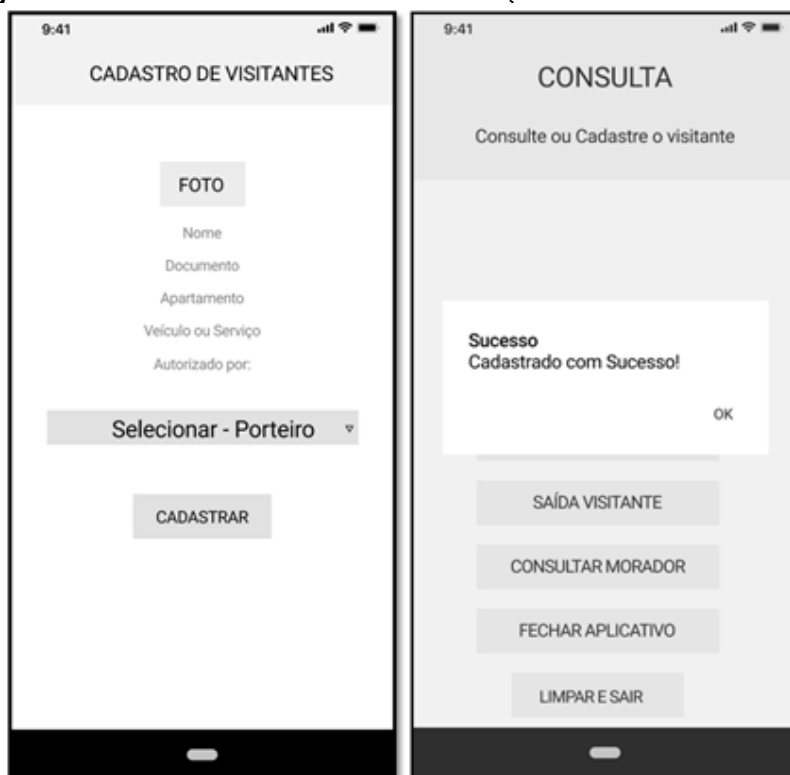
Figura 8 - Confirmação da solicitação e tela principal (*home*).



Fonte: Os autores (2023)

A Figura 9 são as telas de cadastro de visitante e confirmação de cadastro do aplicativo. Através dos dados cadastrados se faz a consulta dos visitantes e moradores, registrando a entrada e saída dos visitantes no condomínio.

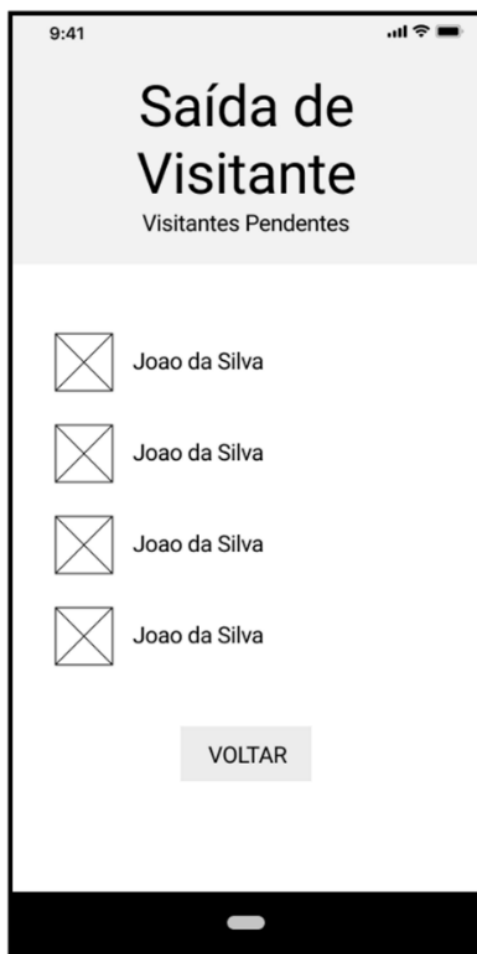
Figura 9 - Cadastro de visitante e Confirmação de cadastro de visitante.



Fonte: Os autores (2023)

A Figura 10 é a tela de consulta do visitante. Trata-se da atualização da saída dos visitantes no condomínio.

Figura 10 - Consulta de visitante.



Fonte: Os autores (2023).

As figuras acima representam o começo da criação do aplicativo, para melhor visualização da estrutura do projeto.

3. RESULTADO

Neste item serão abordados os testes de usabilidade, as telas e uma conclusão dos objetivos do sistema Meu Condomínio.

3.1 Testes de usabilidade

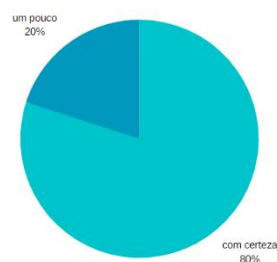
Os testes foram realizados no dia 19 de novembro de 2020, por cinco pessoas de modo *on-line*, através do *link*:

<https://2sm5kl88.optimalworkshop.com/optimalsort/l4avhs7p>

As tarefas solicitadas aos usuários foram: localizar a página de cadastro de visitantes, registrar a saída do visitante, descobrir como obter um *login* e como filtrar moradores através do veículo e do apartamento. Os participantes analisaram a dificuldade de executar as tarefas.

Gráfico 1 - Resposta dos usuários à primeira pergunta.

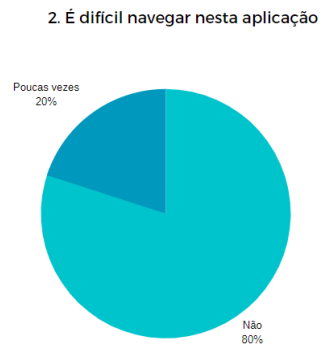
1. Esta aplicação trata de um assunto que é interessante para mim



Fonte: Os autores (2023).

A primeira questão levantada foi se o assunto do aplicativo é interessante para o participante. De acordo com o Gráfico 1, os dados levantados pelas pesquisas apontam que 80% das pessoas que foram entrevistadas responderam que com certeza a aplicação trata de um assunto que é interessante, 80% disseram sim, 20% disseram que é um pouco interessante e 0% disseram não.

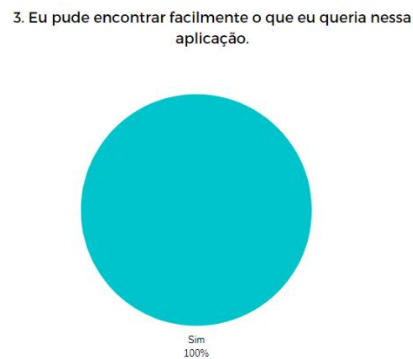
Gráfico 2 – Resposta dos usuários à segunda pergunta.



Fonte: Os autores (2023).

No Gráfico 2 é mostrado que 80% das pessoas que foram entrevistadas responderam que não é difícil navegar na aplicação, 20% disseram poucas vezes, 0% disseram algumas vezes e 0% disseram sempre.

Gráfico 3 – Resposta dos usuários à terceira pergunta.

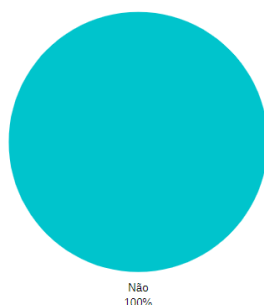


Fonte: Os autores (2023).

O Gráfico 3 aponta que 100% das pessoas que foram entrevistadas responderam que encontraram facilmente o que queriam nessa aplicação sendo que nenhuma pessoa disse que é difícil

Gráfico 4 - Resposta dos usuários à quarta pergunta.

4. Essa aplicação precisa de mais explicação introdutória.

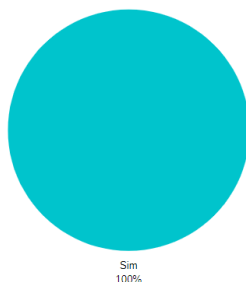


Fonte: Os autores (2023).

De acordo com o Gráfico 4, os dados levantados pelas pesquisas apontam que 100% das pessoas que foram entrevistadas responderam que o conteúdo mostrado era suficiente como informação introdutória e 0% responderam que não.

Gráfico 5 - Resposta dos usuários à quinta pergunta.

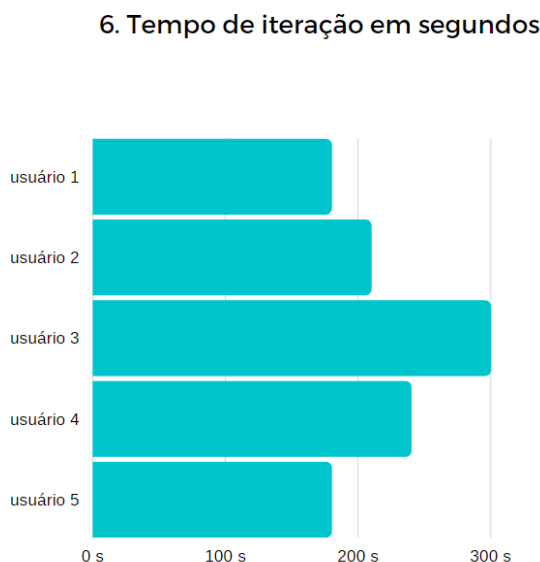
5. Eu me sinto no controle quando utilizo essa aplicação



Fonte: Os autores (2023).

O Gráfico 5 evidencia que os dados levantados pelas pesquisas apontam que 100% das pessoas que foram entrevistadas responderam que se sentiram no controle quando utilizaram a aplicação e 0% disseram não.

Gráfico 6 – Tempo de iteração dos usuários.



Fonte: Os autores (2023).

O gráfico 6 ilustra que a média de tempo que os usuários levaram para fazer o teste solicitado no aplicativo foi de 3,42 minutos, sendo 5 minutos o maior tempo, e o menor tempo foi 3 minutos.

3.1.1 Resposta dos testes de usabilidade

Com os testes realizados, foi perceptível que o nível de aceitação entre os entrevistados foi extremamente satisfatório. Não foram obtidos pontos negativos.

Foi solicitada a alteração no processo de registro de saída de visitantes. Antes da modificação, com apenas um clique na imagem do cadastro já seria feito o registro do visitante, o que poderia ocasionar em remover da lista um visitante que ainda estava no prédio e sem a opção de desfazer o erro.

Abaixo está o exemplo do processo anterior. A figura 11 representa as telas do visitante antes e depois de sua saída.

Figura 11 - Wireframe da tela de saída antes da alteração.



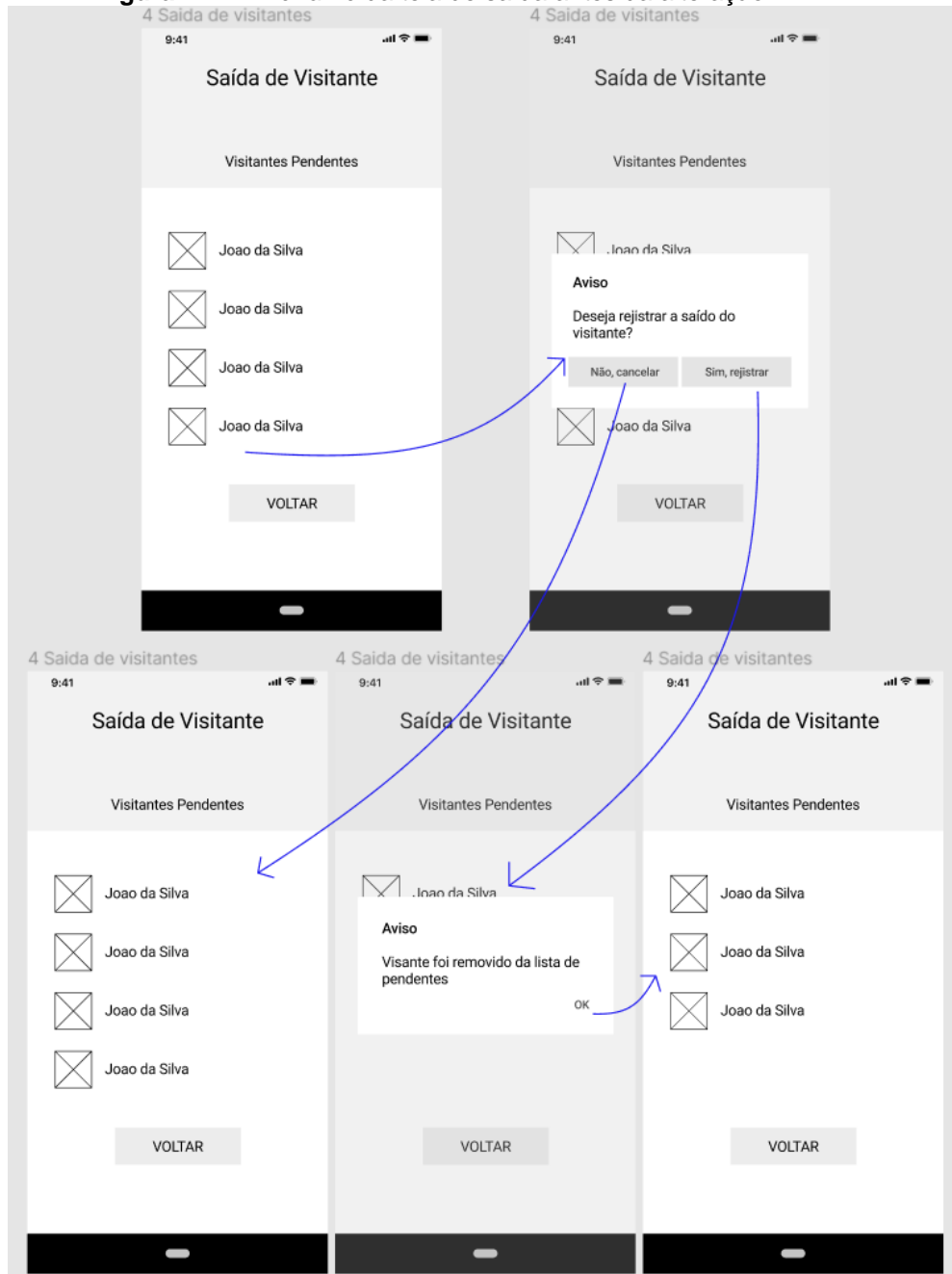
Fonte: Os autores (2023).

Com as mudanças realizadas, quando é solicitada a saída, é apresentada uma mensagem de confirmação de saída do visitante, confirmando se a opção selecionada realmente deverá ser executada.

A seguir está o exemplo do novo processo após as alterações:

A figura 12 representa as telas do visitante antes da alteração de sua saída e depois de sua saída.

Figura 12 – Wireframe da tela de saída antes da alteração.



Fonte: Os autores (2023).

3.2 Telas do sistema

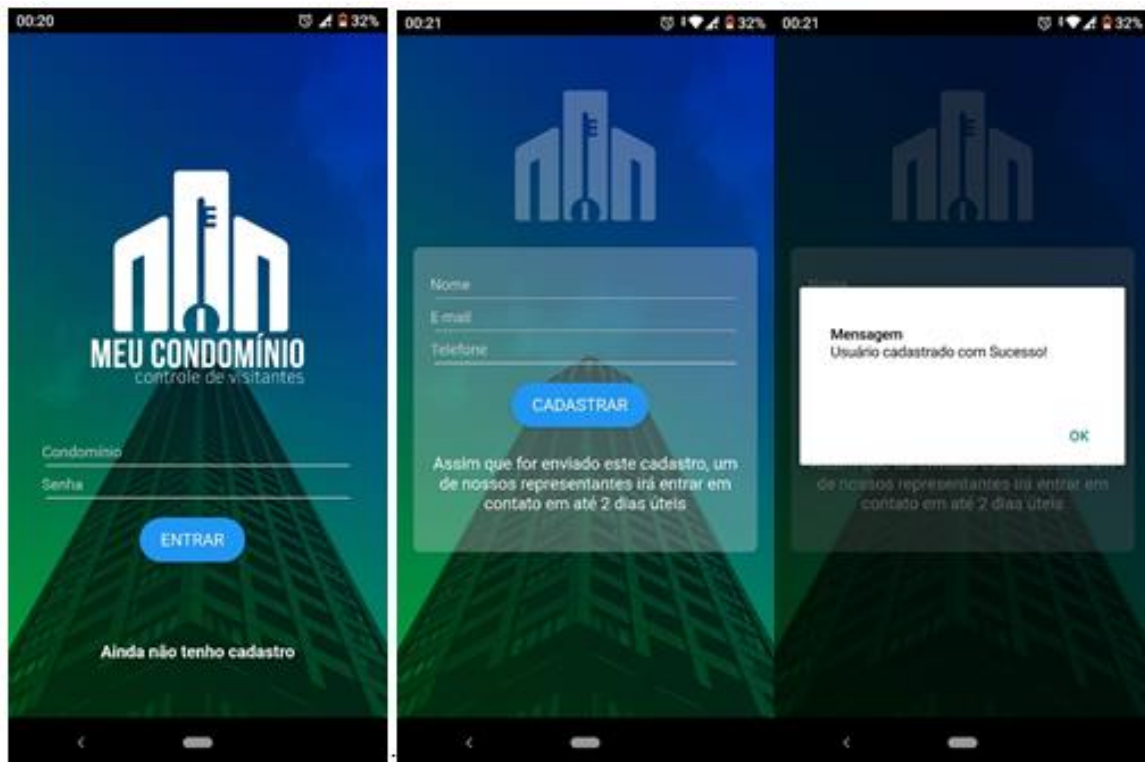
As telas foram desenvolvidas no *Figma*, uma ferramenta de *design* e prototipagem. Através da ferramenta foi possível desenvolver o projeto e tornar o protótipo interativo. As tarefas solicitadas foram disponibilizadas no *link* para testes.

3.2.1 Mockup das telas

As telas de *login* e cadastro foram elaboradas com a finalidade de ser fácil de fazer o cadastro e entrar no sistema.

A figura 13 representa as páginas de *login* e cadastro do aplicativo.

Figura 13 – Páginas de *login* e cadastro.

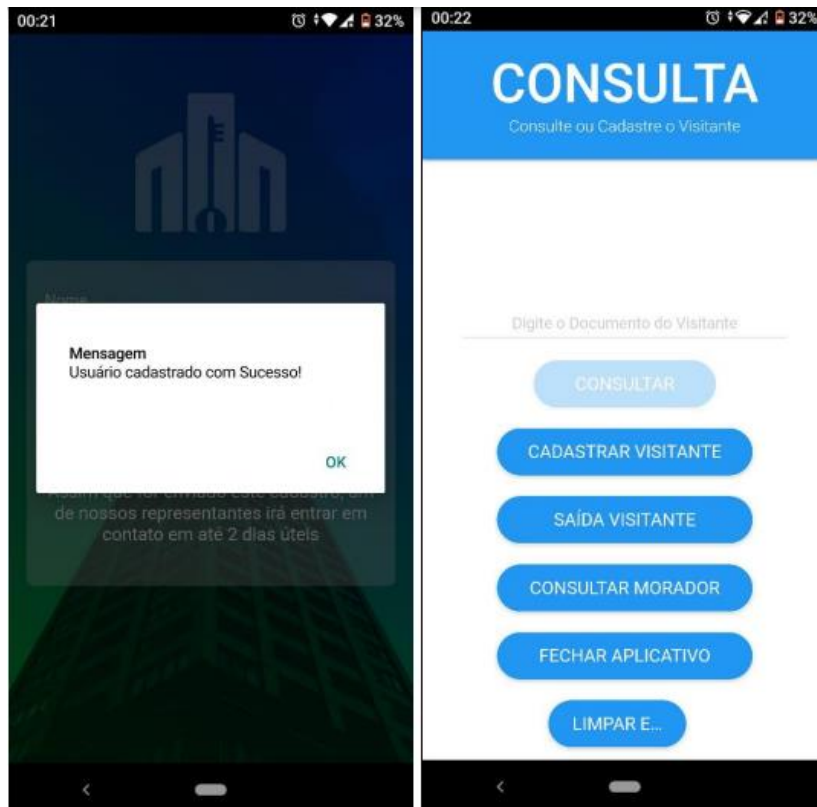


Fonte: Os autores (2023)

Foram criadas as telas para consultar e cadastrar usuários, para verificar se uma pessoa já foi cadastrada.

A figura 14 representa as telas de cadastro do usuário e consulta.

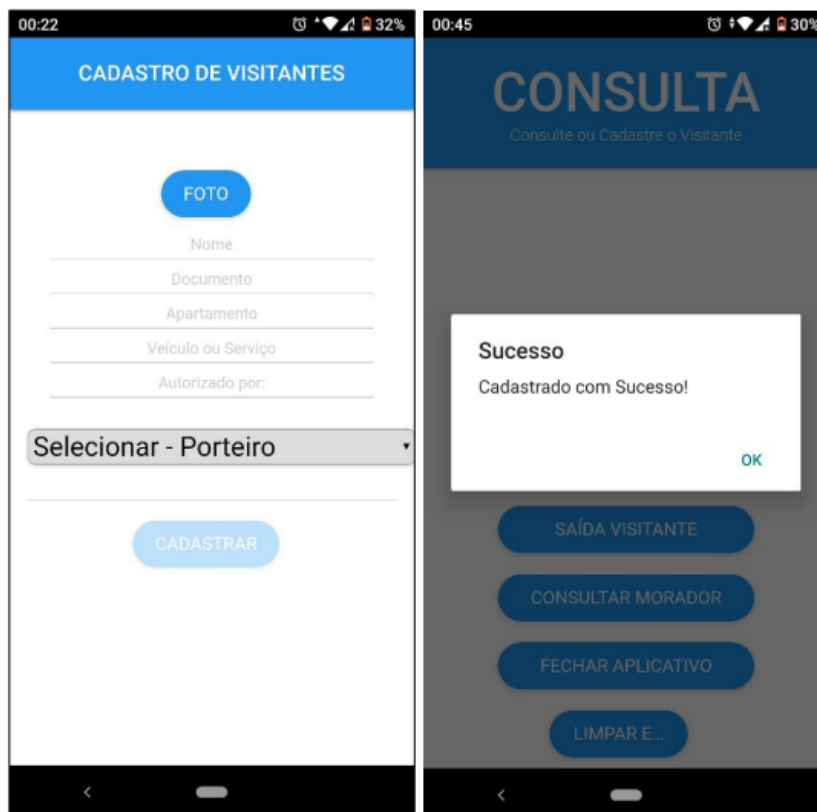
Figura 14 – Telas de cadastro de usuário e consulta.

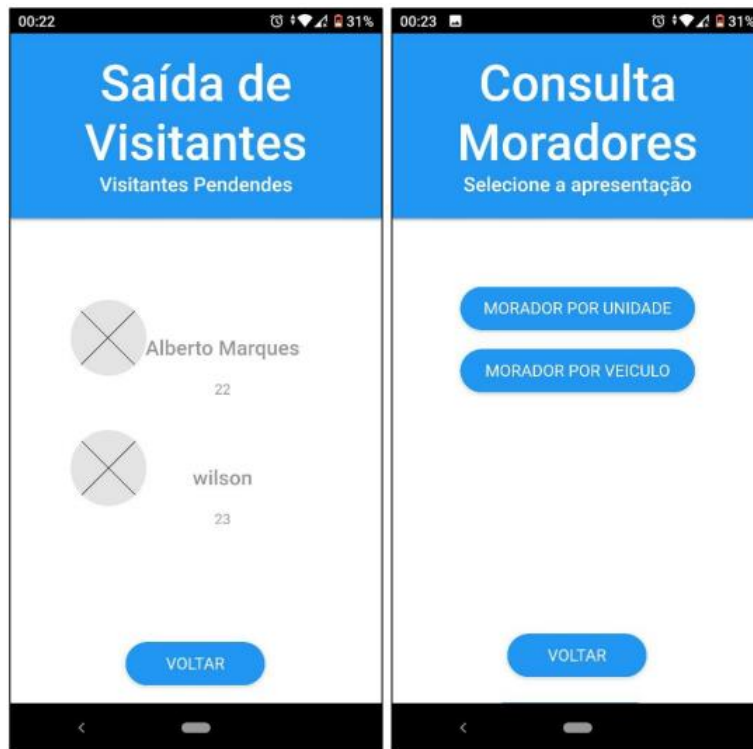


Fonte: Os autores (2023).

A figura 15 representa as telas de cadastro, consulta e saída do usuário.

Figura 15 – Telas de Cadastro, Consulta e Saída.



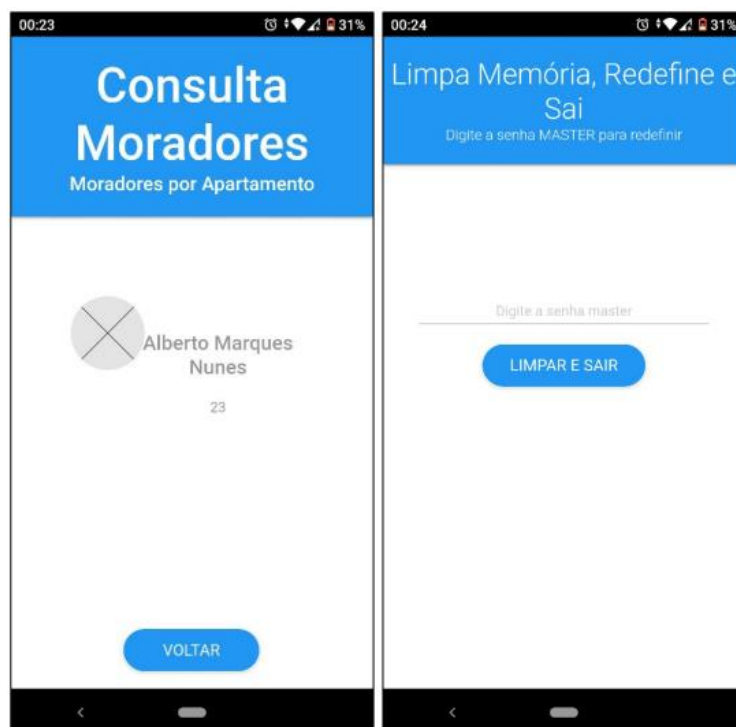


Fonte: Os autores (2023).

Há também uma tela para deletar moradores, conforme mostra a figura a seguir:

A figura 16 representa as telas de deletar moradores cadastrados no aplicativo.

Figura 16 – Limpar Memória do aplicativo.



Fonte: Os autores (2023).

3.3 Conclusões do teste de usabilidade

De um modo geral, os testes foram satisfatórios e colaboraram com uma melhoria no aplicativo. Os usuários são os clientes do final do projeto, e podem ter percepções diferentes do desenvolvedor. Para melhor atender às necessidades, foi realizada a alteração na mensagem de confirmação de saída do visitante, para que evite erros de quem está executando a tarefa.

4. Considerações finais

O problema identificado no início da pesquisa foi a falta de um controle eficiente da movimentação de pessoas no condomínio residencial. Foi proposto a criação de um sistema que permitisse controlar a entrada e a saída de visitantes e prestadores de serviço, registrando os horários e fotos, aumentando a segurança e prevenção de sinistros por desencorajamento.

Os resultados obtidos demonstram que o sistema atende as necessidades pois, no que se refere ao controle, as informações integradas permitem a rápida inclusão e recuperação de informações fotos e horários apoiando a área de segurança; pelo fato do sistema ser integrado e ter a hospedagem do banco de dados em nuvem. Porém, esta solução traz uma dependência da *Internet*, algo tolerável, levando em consideração a ampla difusão do uso da *Internet* hoje em dia.

Com a solução proposta, e o banco de dados do sistema estar em nuvem, todas as funcionalidades do sistema estão disponíveis quando necessário.

Uma possível evolução do sistema seria a criação de módulos de controle de veículos e moradores temporários, ampliando a característica remota que uma arquitetura cliente-servidor disponibiliza.

REFERÊNCIAS

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML - Guia do usuário**. Rio de Janeiro: Campus, 2000.

DATE, C. J. **Introdução a Sistemas de BANCO DE DADOS**. 7^o ed. Rio de Janeiro: Campus, 1989.

ELMASRI, R.; NAVATHE, S. B. **SISTEMAS DE BANCO DE DADOS**. 4^a ed. São Paulo: Pearson, 2006.

FAGUNDES, R. M. **Engenharia de Requisitos: do perfil do analista de requisitos ao desenvolvimento de requisitos com UML e RUP**. Salvador: Clube de Autores, 2011.

GUEDES, G. T. A. **UML 2 - UMA ABORDAGEM PRÁTICA**. 2^a ed. São Paulo: Novatec, 2011.

GEHRKE, J.; RAMARKRISHNAN, R. **SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS**. 3^a ed. Porto Alegre: AMGH, 2011.

HEITLINGER, P. **O Guia Prático da XML**. Porto: Centro Atlântico, 2001.

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5^a ed. Rio de Janeiro: Elsevier, 2006.

MAÑAS, A. V. **Administração de Sistemas de Informação**. 8^a ed. São Paulo: Érica, 2011.

MAXIM, B. R.; PRESSMAN, R. S. **Engenharia de Software - UMA ABORDAGEM PROFISSIONAL**. 8^a ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson, 2004.

SUEHRING, S. **MySQL - A Bíblia**. 7^a ed. Rio de Janeiro: Elsevier, 2002.

TAURION, C. **COMPUTAÇÃO EM NUVEM** - Transformando o mundo da Tecnologia da Informação. Rio de Janeiro: Brasport, 2009.