



UNIVERSIDADE DO SUL DE SANTA CATARINA

ANA MÁRCIA GOULART NETO

GUSTAVO COELHO NAVARRO PIRES

**USO DE BANCO DE DADOS EMBARCADOS PARA O DESENVOLVIMENTO DE
TECNOLOGIAS MÓVEIS**

Palhoça

2021

**ANA MÁRCIA GOULART NETO
GUSTAVO COELHO NAVARRO PIRES**

**USO DE BANCO DE DADOS EMBARCADOS PARA O DESENVOLVIMENTO DE
TECNOLOGIAS MÓVEIS**

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade do Sul de Santa Catarina como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Flávio Ceci, Dr.

Palhoça
2021

**ANA MÁRCIA GOULART NETO
GUSTAVO COELHO NAVARRO PIRES**

**USO DE BANCO DE DADOS EMBARCADOS PARA O DESENVOLVIMENTO DE
TECNOLOGIAS MÓVEIS**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Graduação em Sistemas de Informação da Universidade do Sul de Santa Catarina.

Palhoça, 15 de junho de 2021.

Professor e orientador Flávio Ceci, Dr.
Universidade do Sul de Santa Catarina

Prof. Aran Bey T. Morales, Dr.
Universidade do Sul de Santa Catarina

Prof. Alexandre Vitoreti, MSc.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Ana Márcia agradece a:

As pessoas envolvidas na minha formação desde o pré-escolar na E.E.B Manoel Cruz, onde tive meu primeiro incentivo sobre tecnologia. Sou grata a cada professor (a) e colegas que sempre estiveram ao meu lado nesta jornada e tanto me apoiaram.

Aos meus pais Ivonete da Silva Jesus e João Batista Neto, que sempre ouviram minhas reclamações, rezaram pelos meus medos, conquistas, e me deram forças para conseguir chegar até aqui. Com certeza cada puxão de orelha valeu a pena, sempre levarei comigo todo o esforço que eles fizeram para que eu e minha irmã Thaísa da Silva Neto tivéssemos uma boa educação.

A minha madrinha Maria Zalmira da Silveira e melhor amiga Evelise da Silveira, pelos conselhos e apoio quando decidi ir embora de São Joaquim para ingressar na universidade em outra cidade, além de todos os ensinamentos e apoio, que me ajudaram a ser quem eu sou hoje.

A todos os professores do curso em especial a coordenadora Vera Rejane Niedersberg Schuhmacher, por todo o apoio e consideração durante esses quatro anos e meio. E é claro ao nosso orientador Dr. Flavio Ceci, principalmente pela paciência, compreensão, e é claro pelas boas histórias, que tornou tudo mais leve.

A Rejane Feltes que esteve ao meu lado e sempre que necessário me socorreu nos momentos de apuro na cidade “grande”.

Ao amigo e colega Diego de Souza Costa, por todos os trabalhos em grupos, e pelo melhor churrasco. Mesmo com as piadas sobre plantações de batatas.

E por fim, não menos importante, meu companheiro de TCC e também de vida Gustavo, que me fez lembrar onde eu queria chegar, que foi minha base quando eu decidi iniciar minha carreira como desenvolvedora. Sempre esteve ao meu lado, sempre foi meu confidente, e ao longo do curso se tornou mais que um amigo. E juntos batalhamos pelos nossos sonhos e metas.

Autor 2 agradece a:

Primeiramente aos meus pais, Roney Navarro Pires e Roseane Coelho, que são os responsáveis pela minha educação, me apoiaram e fizeram o possível e o impossível para que eu conseguisse chegar até aqui hoje.

Também ao Dr. Flávio Ceci, nosso orientador que com muita paciência e leveza conseguiu nos ajudar e apoiar a finalizar este trabalho.

Ao Diego de Souza Costa, amigo que fez parte de praticamente todos os trabalhos em grupo com muita parceria e tornou mais leve todos esses quatro anos e meio de faculdade.

Por último, a Ana Márcia, que me ajudou e me apoiou durante todo o curso, não apenas como colega mas também como melhor amiga, e neste tempo acabou passando de uma simples amizade para minha parceira de vida.

RESUMO

Com o avanço da tecnologia, as empresas passaram a fazer uso de softwares como recursos para o dia a dia da sua área operacional e, também gerencial. Esses softwares servem principalmente para armazenar informações de suma importância para os usuários.

Para guardar estas informações, em geral os softwares utilizam servidores e bancos de dados que precisam ficar a todo momento a disposição das aplicações clientes para garantir que todas as informações necessárias sejam armazenadas e o fluxo do dia a dia das empresas não seja afetado. As aplicações clientes também precisam conseguir se comunicarem com os servidores, tendo acesso constante a internet.

Com esta premissa em mente e sabendo que diversos problemas podem ocorrer que impeçam o acesso a internet por parte das aplicações cliente, é necessário encontrar soluções que possibilitem manter os softwares disponíveis mesmo sem conexão com a internet e ao mesmo tempo manter a consistência dos dados.

Para isso, foi proposto o uso de banco de dados locais e desenvolvido um protótipo com a finalidade de demonstrar a solução apresentada. Além disso, foi elaborada e distribuída uma pesquisa para identificar a eficácia da solução proposta. Com a análise realizada em cima dos resultados dessa pesquisa, foi identificado que a solução apresentada é sim eficaz e capaz de solucionar os possíveis problemas que causariam a falta de conexão.

Palavras-chave: Banco de dados embarcado. Aplicativos.

ABSTRACT

With the advancement of technology, companies started to use software as resources for the day-to-day operations of their operational and management areas. These software mainly serve to store information of utmost importance to users.

To store this information, software generally uses servers and databases that need to be available to client applications at all times to ensure that all the necessary information is stored, and the day-to-day flow of companies is not affected. Client applications also need to be able to communicate with servers, having constant access to the internet.

With this premise in mind and knowing that several problems can occur that prevent internet access by client applications, it is necessary to find solutions that make it possible to keep the software available even without internet connection and at the same time maintain data consistency.

For this, the use of a local database was proposed, and a prototype was developed in order to demonstrate the presented solution. In addition, a survey was designed and distributed to identify the effectiveness of the proposed solution. With the analysis carried out on the results of this research, it was identified that the presented solution is indeed effective and capable of solving the possible problems that would cause the lack of connection.

Keywords: Embedded database. Applications.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxograma das etapas metodológicas.....	29
Figura 2 - Arquitetura da solução	30
Figura 3 - Diagrama de classes	33
Figura 4 - Protótipo da tela inicial do aplicativo	38
Figura 5 - Tela para leitura de itens de um novo pedido	39
Figura 6 - Tela para acompanhamento do status dos pedidos	40
Figura 7 - Diagrama de caso de uso do sistema proposto.....	41
Figura 8 - Domínio problema	45
Figura 9 - Diagrama Entidade Relacionamento do banco de dados local	45
Figura 10 - Ferramentas.....	47
Figura 11 - Componentes do Android	48
Figura 12 - Logomarca Java	50
Figura 13 - Classe de configuração do Room.....	53
Figura 14 - Tela inicial	54
Figura 15 - Modal pedidos pendentes.....	55
Figura 16 – Tela iniciar nova leitura do pedido.....	56

LISTA DE QUADROS

Quadro 1 - Uso do telefone celular em todo o mundo por país.....	16
Quadro 2 - Requisitos funcionais	36
Quadro 3 - Requisitos não funcionais.....	37
Quadro 4 - Regras de negócio	37
Quadro 5 - Caso de uso UC01	42
Quadro 6 - Caso de uso UC02	42
Quadro 7 - Caso de uso UC03	42
Quadro 8 - Caso de uso UC04	43

LISTA DE TABELAS

Tabela 1 - Comparativo Bancos de dados	25
--	----

SUMÁRIO

1	INTRODUÇÃO	15
1.1	PROBLEMA	18
1.2	OBJETIVOS	18
1.2.1	Objetivo Geral.....	18
1.2.2	Objetivos Específicos	19
1.3	JUSTIFICATIVA	19
1.4	ESTRUTURA DA MONOGRAFICA	20
2	REFERENCIAL BIBLIOGRÁFICO	21
2.1	BANCO DE DADOS	21
2.1.1	Banco de dados relacional.....	22
2.1.2	Banco de dados não relacional.....	23
2.1.3	Banco de dados embarcados	23
2.2	DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS	25
2.2.1	Plataformas de Desenvolvimento para Dispositivos móveis	26
2.2.2	Arquiteturas de software	26
2.2.2.1	Modelo cliente-servidor.....	27
2.2.2.2	Modelo aplicação monolítica.....	27
3	MÉTODO DE PESQUISA.....	28
3.1	DEFINIÇÃO DA PESQUISA	28
3.2	ETAPAS METODOLÓGICAS	29
3.3	ARQUITETURA DA SOLUÇÃO	30
3.4	DELIMITAÇÕES.....	30
4	DESENVOLVIMENTO DA SOLUÇÃO	32
4.1	UML	32
4.2	ICONIX	33
4.3	ESPECIFICAÇÃO DO PROTÓTIPO.....	35
4.3.1	Requisitos	35
4.3.2	Requisitos funcionais	36
4.3.3	Requisitos não funcionais.....	36
4.3.4	Protótipo de tela.....	38
4.3.5	Casos de uso.....	40

4.3.6	Modelo de domínio	44
4.4	MODELO DE DADOS	45
5	DESENVOLVIMENTO.....	47
5.1	FERRAMENTAS.....	47
5.2	LINGUAGENS	50
5.2.1	Java	50
5.2.2	SQL	51
5.3	HISTÓRICO DO DESENVOLVIMENTO.....	51
5.3.1	Telas e pontos chaves no desenvolvimento do código	52
5.3.1.1	Configuração do Room.....	52
5.3.1.2	Tela de inicial	53
5.3.1.3	Modal de seleção de pedidos pendentes	55
5.3.1.4	Tela de nova leitura de pedido.....	56
5.4	AValiação DO PROTÓTIPO.....	57
5.5	ANÁLISE DOS RESULTADOS	57
6	CONCLUSÕES E TRABALHOS FUTUROS.....	59
6.1	CONCLUSÃO.....	59
6.2	TRABALHOS FUTUROS	60
	APÊNDICES	64
	APÊNDICE A – EXEMPLO DE APÊNDICE	64
	ANEXOS.....	ERROR! BOOKMARK NOT DEFINED.
	ANEXO A – EXEMPLO DE ANEXO.....	ERROR! BOOKMARK NOT DEFINED.
	ANEXO B – TÍTULO.....	ERROR! BOOKMARK NOT DEFINED.

1 INTRODUÇÃO

Em 3 de abril de 1973, em Nova York, Martin Cooper, frente a empresa Motorola, apresentou o primeiro modelo de telefone móvel com grande capacidade, denominado de DynaTAC 8000X. Houveram outros protótipos de telefone móvel e de serviços de telefonia móvel antes desta data: para sistema ferroviários; militares da segunda guerra mundial; o primeiro “celular” desenvolvido pela empresa Ericsson, pesando 40 kg; o Altay, na URSS, pesando 11kg; Nenhum destes modelos tinham todas as características necessárias para um bom funcionamento - alta capacidade, um tamanho e peso aceitável e um consumo de energia relativamente baixo. Foi apenas com o DynaTAC8000X, com 25 cm de comprimento e 7 cm de largura pesando 1 kg, que foi realizada a histórica ligação do diretor de sistema de operações da Motorola. A partir deste ponto o telefone celular foi implantado no Japão e Suécia, posteriormente no Brasil, Portugal e assim por diante. Com este pontapé inicial, os países passaram a desenvolver e efetivar suas próprias tecnologias.

Em 2007 a empresa Apple revolucionou o celular para smartphone, com o lançamento do *iPhone* usando o sistema operacional *IOS*. Além da novidade do sistema operacional os aparelhos passam a ter acesso a rede sem fio (WIFI), capacidade de armazenamento interno maior e câmeras razoavelmente boas. Em 2008 a Google lança o sistema operacional Android no modelo HTC Dream, que se populariza entre as empresas Samsung, Motorola, e LG, que passam a fabricar modelos semelhantes ao *iPhone*. Depois dessa revolução tecnológica, Coutinho (2014, p. 5), esclarece que “novos softwares (programas) e hardwares (aparelhos) também seguem este ritmo acelerado, acompanhados por um crescimento quase exponencial de sua capacidade de processamento a cada novo lançamento”.

Com novas finalidades e funcionalidades os smartphones entraram no mercado com uma nova proposta para o uso do celular, pois ultrapassam sua função inicial, onde falar é o que menos importa. Por meio do desenvolvimento dos aplicativos, hoje ele pode ser usado como meio de comunicação nas mídias sociais; como gerenciador bancário e cartão de crédito (carteira eletrônica); biblioteca de músicas, livros, revistas, vídeos e jogos; máquina fotográfica e filmadora; e ainda, interagir no ambiente físico como controle remoto; interface com aparelhos eletrodomésticos e controlar o funcionamento de casas. Enfim, não tem limites para a sua utilização; e a tendência é que a cada dia mais e mais funções e serviços sejam disponibilizados através deste dispositivo. (GARRIDO, 2013, p. 2).

Segundo a publicação de Valente (2019), 5.1 bilhões de pessoas em todo o planeta utilizam algum tipo de aparelho celular. Hoje, já conseguimos ver pessoas que possuem mais de um celular: um pessoal, um para o trabalho, etc. Além disso, podemos identificar empresas que junto com suas soluções, fornecem smartphones especificamente para o uso de algum aplicativo específico. Como resultado disso, temos mais de 8 bilhões de dispositivos móveis ao redor do mundo, segundo a pesquisa realizada pelo site americano BankMyCell (2020, tradução nossa).

É muito importante destacar que, entre essa quantidade enorme de dispositivos, o Brasil possui o 4º lugar no ranking de país com dispositivos móveis.

Quadro 1 - Uso do telefone celular em todo o mundo por país

RANK	País	COUNTRY/MARKET	TOTAL POPULATION	SMARTPHONE USERS	SMARTPHONE PENETRATION
1		China	1.42B	851M	59.9%
2		India	137B	346M	25.3%
3		United States	329M	260M	79.1%
4		Brazil	212M	96.9M	45.6%
5		Russian Federation	144M	95.4M	66.3%
6		Indonesia	270M	83.9M	31.1%
7		Japan	127M	72.6M	57.2%
8		Germany	82.4M	65.9M	79.9%
9		Mexico	132M	65.6M	49.5%
10		United Kingdom	67.0M	55.5M	82.9%

Fonte: BankMyCell (2020).

Com o avanço dos celulares, em 2001, entrou no mercado a internet móvel. De lá pra cá, com todas as evoluções da internet móvel que acompanharam a evolução dos celulares até o atual estado de *smartphones*, temos três principais tecnologias: 3G, 4G e 5G. O nome 3G é a abreviação de terceira geração (da internet móvel). Inicialmente essa terceira geração tinha a promessa de fornecer uma largura de banda em torno de 144 Kb/s (Kbits por segundo). Contudo, ocorreram evoluções em cima dessa terceira geração, permitindo que fosse fornecido até 168 Mb/s (Megabits por segundo). Essas evoluções foram chamadas de HSPA E HSPA+, e foram apelidadas de 3,5G. Em 2008, tivemos o início da discussão sobre o 4G (quarta geração da internet móvel). A ideia era ter estáveis 100Mb/s para o usuário. Isso era muito positivo, pois então permitiria que fosse realizado streaming de músicas e vídeos pelo celular além de jogos online. A evolução foi iniciada em 2013. Assim como o 3,5G, tivemos o 4,5G ou 4G Plus. Essa é uma evolução ao 4G, permitindo quatro ou cinco vezes mais velocidade para o usuário. A tecnologia está em implementação no Brasil pois depende

da desobstrução da faixa de espectro eletromagnético, que está em uso por canais analógicos de TV aberta. Ainda no escopo de 4G, tivemos a *VoLTE* (Voz via *LTE*). É uma tecnologia que permite realizar ligações via 4G, entregando ainda mais qualidade no serviço ao usuário. (GARRET, 2018).

Como próximo passo, tem-se o 5G. Ele promete até 50 vezes mais velocidade que o 4G, podendo chegar até 1Gb/s (Gigabit por segundo). Segundo Schaeffer (2020), o 5G teve testes iniciados no Brasil no ano de 2020, pela Claro e pela Vivo. Por outro lado, a Tim já tem planos para lançamento da rede 5G ao público em setembro (09/2020). Isso por aqui é muito novo ainda e por isso não estamos totalmente capacitados para oferecimento da tecnologia. Prova disso, é que teremos que fornecer o 5G na faixa de frequência do 3G e do 4G, o que implica uma velocidade menor, já que o 5G necessita de uma faixa de frequência maior. (MEDEIROS, 2020).

Guerra (2019) afirma que, dada a evolução dos celulares até o estado atual de smartphones, iniciamos a abertura de um mercado enorme: o mercado de aplicativos. No momento em que os fabricantes de celulares descobriram que podemos usar os aparelhos para muito mais além de ligações e armazenamento de contatos, um tipo novo de produto muito valioso passou a ser fabricado: os aplicativos. Hoje tem-se duas principais lojas de aplicativos - *Play Store*, da Google e *App Store*, da Apple. Cada loja está vinculada a um sistema operacional: *Android* e *IOS*, respectivamente. A loja de aplicativos da Google conta com 3,6 milhões de aplicativos disponíveis para download, contra 2,1 milhões de aplicativos na loja da Apple. O mercado de aplicativos veio evoluindo muito rapidamente. Hoje contamos com aplicativos em todos os segmentos: entretenimento, educacional, comunicação, etc. A evolução desse mercado é tão rápida que chega a ter uma deficiência grande de profissionais. Com o grande avanço desse mercado, e da tecnologia (muito mais rápido do que o avanço de nossas tecnologias de internet móvel), viemos enfrentando uma grande questão: como contornar a falta de rede em diversos lugares ao redor do mundo? Muitas empresas hoje possuem essa demanda e precisam pensar em alguma solução para poder suprir este cenário.

1.1 PROBLEMA

Um software que precise estar sempre conectado à rede, para comunicação com um servidor ou algo do tipo, como irá lidar em caso de estar sendo executado em um local onde não há rede? Uma boa resposta seria deixar para usar o aplicativo no momento em que houver rede. Mas e se o aplicativo precisar ser utilizado no momento exato? Como por exemplo para marcar o fim da jornada de trabalho, fazer um pedido de venda, encontrar algum lugar entre uma lista para obter a localização ou um número de telefone. Em alguns casos, é possível anotar os dados em um papel e posteriormente colocar no aplicativo (o que não faz sentido algum, visto que tiraria toda a finalidade de um software, que visa facilitar a vida de um usuário e não ter que fazê-lo usar mais um pedaço de papel), em outros, como na localização, se não tivermos buscado as informações desejadas antes e as anotado em algum local, não iríamos conseguir obter a informação de um servidor.

Diante deste cenário como permitir o uso de uma funcionalidade de um aplicativo ou até dele em sua totalidade, durante uma falta de rede para que posteriormente, sem que percam os dados que devem ir para o servidor ou ainda, como obter os dados essenciais para uso da aplicação mesmo sem ter rede de comunicação? Se analisarmos, conseguiremos chegar a algumas soluções. Contudo, uma das melhores soluções com certeza é o uso de um banco de dados embutido em um aplicativo.

1.2 OBJETIVOS

Nesta seção, será apresentado os objetivos gerais e específicos desta monografia.

1.2.1 Objetivo Geral

Analisar o uso de banco de dados embarcados para o desenvolvimento de tecnologias móveis.

1.2.2 Objetivos Específicos

- Levantar os bancos de dados que podem ser embarcados em tecnologias móveis.
- Definir um cenário de teste para aplicar os conceitos estudados.
- Desenhar uma proposta de solução baseada em mobile.
- Desenvolver uma aplicação com o banco de dados SQLite.
- Desenvolver um protótipo funcional a partir da proposta de solução.
- Avaliar o protótipo desenvolvido com alguns possíveis usuários da solução.

1.3 JUSTIFICATIVA

Um aplicativo que não consegue cumprir a sua função, impossibilitando que o usuário consiga utilizá-lo para fins pessoais ou profissionais, é inútil. Não há sentido em possuir uma ferramenta que funciona apenas parcialmente. A maior parte dos aplicativos existentes necessita de conexão de rede para comunicação com um servidor que contenha todas as informações necessárias para funcionamento. Contudo, condições adversas impedem a comunicação constante através da internet, como por exemplo localizações que possuem áreas de sombra de sinal ou até mesmo fenômenos da natureza. Dessa forma, as ferramentas acabam se tornando inutilizadas e desperdiçando suas funcionalidades, podendo causar um impacto gigantesco aos usuários de aplicativos de uso pessoal e principalmente em usuários e empresas que usufruem de aplicativos para uso profissional.

Na maioria dos sistemas ou aplicativos desenvolvidos existe a necessidade de se armazenar algum tipo de informação permanente que será utilizada em algum momento posterior. A utilização de um sistema de banco de dados como forma de armazenamento para informações de aplicativos móveis permite a opção de uso desses aplicativos em modo offline, ou seja, mesmo sem internet o uso da aplicação não é comprometido. (COSTA; SANTOS, 2015, p. 20)

Sendo assim, devido à dificuldade de disponibilizar o uso offline de aplicativos em smartphones pela falta de internet móvel em diversas localidades e situações ao redor do mundo, essa pesquisa se justifica através do estudo do uso de banco de dados embutidos em aplicações de dispositivos móveis com a finalidade de disponibilizar o uso dos aplicativos em todas situações de falta de rede, tornando as ferramentas funcionais independente de rede.

1.4 ESTRUTURA DA MONOGRAFIA

Este trabalho está organizado em 6 capítulos, introdução, revisão bibliográfica, método, desenvolvimento da proposta e conclusão. O primeiro capítulo apresenta a introdução, a problemática, os objetivos e a justificativa.

No capítulo 2, tem-se a revisão bibliográfica. Nela são abordados os temas pertinentes ao projeto com maior detalhes, como uma breve introdução a banco de dados, modelo de banco de dados relacional e não relacional, banco de dados embarcados, desenvolvimento e arquiteturas de softwares para dispositivos móveis.

No capítulo 3, é apresentado o método científico da proposta.

No capítulo 4, descreve-se o desenho da solução.

Em seguida, no capítulo 5, desenvolvimento e aplicação da solução proposta.

Finalmente, no capítulo 6, são descritas as conclusões obtidas e as sugestões para os trabalhos futuros.

2 REFERENCIAL BIBLIOGRÁFICO

Este capítulo tem como objeto, apresentar as tecnologias utilizadas para o desenvolvimento deste trabalho. Inicialmente é feita uma visão geral sobre banco de dados e os modelos relacionais e não relacionais. Em seguida aborda-se banco de dados embarcados e são citados alguns exemplos. Por fim, é apresentado o desenvolvimento de um software para dispositivos móveis, plataformas de desenvolvimento e arquiteturas de softwares.

2.1 BANCO DE DADOS

Segundo Elmadri; Navathe (2011), bancos de dados e sistemas de banco de dados são um componente essencial da vida na sociedade moderna, a maioria de nós encontra diariamente diversas atividades que envolvem alguma interação com um banco de dados. Exemplos do uso de banco de dados no nosso dia a dia são: quando fazemos compras no supermercado, efetuamos pagamentos com cartões, reservas em hotéis, compras em lojas de livro. Esses são apenas alguns momentos onde estamos usando algum banco de dados e nem nos damos conta.

Mas então o que é um banco de dados?

Um banco de dados é uma coleção de dados relacionados. Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significado implícito. Por exemplo, considere os nomes, números de telefone e endereços das pessoas que você conhece. Você pode ter registrado esses dados em uma agenda ou, talvez os tenha armazenados em um software como Microsoft Access ou Excel. Essa coleção de dados relacionados, com um significado implícito, é um banco de dados. (ELMASRI; NAVATHE, 2011, p. 3).

Beaulieu (2010, p.18) diz que “por armazenar dados eletronicamente, um sistema de banco de dados é capaz de recuperar e armazenar dados com mais agilidade, além de fornecer dados atualizados em tempo real para os usuários”.

Os Primeiros bancos de dados gerenciavam dados armazenados em fitas magnéticas. como normalmente existiam bem mais fitas magnéticas do que leitores de fitas, os técnicos tinham a tarefa de carregar e descarregar fitas

conforme os dados eram solicitados. como os computadores daquela época continham pouca memória, múltiplas requisições dos mesmos dados geralmente exigiam que os dados fossem lidos da fita várias vezes. apesar desses sistemas de banco de dados representarem uma melhoria significativa em relação aos bancos de dados em papel, eles são um eco distante daquilo que é possível realizar com a tecnologia atual. (BEAULIEU, 2010, p. 18).

Na próxima seção feito uma breve explicação sobre banco de dados relacionais.

2.1.1 Banco de dados relacional

O modelo de dados relacional foi introduzido pelo pesquisador Ted Codd, da IBM Research, em 1970, no artigo Relational Model of Data Large Shared Data Banks (Modelo Relacional de Grandes Bancos de Dados Compartilhados). Onde ele emprega o conceito de relação matemática para resolver algumas das inadequações dos modelos de banco de dados prevalecentes. Em 1980 o modelo relacional se torna o primeiro modelo implementado em aplicações comerciais.

Para Elmadri e Navath (2011), “o modelo relacional representa o banco de dados como uma coleção de relações. Informalmente, cada relação é semelhante a uma tabela de valores [...]”.

O modelo relacional forneceu uma maneira padrão de representar e consultar dados que poderiam ser usados por qualquer aplicativo. Desde o início, os desenvolvedores reconheceram que a principal força do modelo de banco de dados relacional estava no uso de tabelas, que era uma maneira intuitiva, eficiente e flexível de armazenar e acessar informações estruturadas. (ORACLE, 2020).

A linguagem SQL é uma linguagem utilizada para manipulação de bancos de dados relacionais. O primeiro padrão foi publicado em 1986. Com ela podemos criar, excluir, modificar e buscar tabelas, registros e todos os outros objetos que um banco de dados relacional nos permite criar.

Segundo Beaulieu (2010), foi junto com a definição de modelo relacional que Codd propôs uma linguagem chamada DSL/Alpha que, após refinamento e aprimoramentos, acabou se tornando o SQL.

Antes de se tornar o SQL, a linguagem se chamou SQUARE e SEQUEL, este último dando lugar ao atual nome.

Apesar de muita gente afirmar que SQL é uma abreviação para “Structured Query Language”, SQL na verdade não é uma abreviação de nada. Para se referir a linguagem pode-se utilizar separadamente as letras “S”, “Q” e “L” ou pronunciar “síquel”.

2.1.2 Banco de dados não relacional

Os bancos de dados não relacionais surgiram como resposta ao acelerado crescimento das tecnologias e internet, fazendo-se necessário maior velocidade em processamento de dados não estruturados.

Um banco de dados não relacional é um banco de dados que não usa o esquema de tabela de linhas e colunas encontrado na maioria dos sistemas de banco de dados tradicionais. Em vez disso, os bancos de dados não relacionais usam um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados que está sendo armazenado. Por exemplo, os dados podem ser armazenados como pares chave/valor simples, como documentos *JSON* ou como um gráfico que consiste em bordas e vértices. (MICROSOFT, 2018).

Um termo muito utilizado para banco de dados não relacional é o *NoSQL*, isso significa que não utiliza a linguagem SQL para consultas de dados, ao invés disso usa linguagens de programação e constructos.

2.1.3 Banco de dados embarcados

Um banco de dados embarcado é um banco de dados que não precisa de um servidor dedicado para rodar. Ele pode ser distribuído, integrado com a aplicação, como é o caso dos bancos de dados para dispositivos móveis.

Os bancos de dados embarcados são pensados para rodar em ambientes “fracos”, por não exigirem muito recurso do dispositivo em que estão inseridos.

Pelos bancos de dados estarem acoplados a uma aplicação, eles são totalmente gerenciados por ela.

Falaremos a seguir sobre os bancos que podem ser usados em aplicações *mobiles*, levando em consideração seus prós e contras.

- SQLite:** O SQLite é integrado a todos os dispositivos móveis. Também é conhecido por implementar a autossuficiência, sem necessidade de servidor, não precisa de configuração e é transacional. Por ter código de domínio público, é, portanto, gratuito para qualquer finalidade, seja ela privada ou comercial. Os grandes como o Facebook e Google por exemplo utilizam esse banco de dados em suas aplicações. Ao contrário da maioria dos bancos de dados SQL, o SQLite não tem um processo de servidor separado. Lê e grava diretamente em arquivos de disco comuns. A biblioteca do SQLite utilizada nas aplicações, tem um tamanho de cerca de 250KB. Um ponto importante a se frisar é que a plataforma Android disponibiliza suporte nativo ao SQLite.

- MySQL:** O MySQL também ficou extremamente popular por ser um banco de dados embarcado e empacotado com outros produtos.

- Firebird:** O Firebird é derivado do código-fonte do Borland InterBase 6.0. Não possui licença, ou seja, é gratuito. Por mais que seja compacto, é muito poderoso, possui muitos recursos e bom desempenho.

- Apache Derby:** É um banco de dados relacional, implementado em Java, e pode ser utilizado para qualquer solução baseada em Java. As vantagens de trabalhar com este banco é que ele é bem pequeno ocupando apenas 3,5 MB e já possui o driver JDBC integrado.

- SQLServer Compact:** É um banco de dados embarcado gratuito que pode ser usado tanto para sites da Web quanto aplicativos mobile e desktop. É leve, possibilita desenvolvimento fácil e tem suporte para migração rápida para SQL Server.

A seguir esta a tabela de comparação dos banco de dados levantados acima:

Tabela 1 - Comparativo Bancos de dados

Banco de dados	Desenvolvedor	Relacionamento	Suporte Nativo Android	Limite de armazenamento
SQLite	SQLite Consortium	relacional	sim	128TB
MySQL	Oracle Corporation	relacional	não	16TB
Firebird	Fundação FirebirdSql	não relacional	não	7TB
Apache Derby	Apache Software Foundation	relacional	não	-
SQLServer Compact	Microsoft	relacional	não	2GB

Fonte: Autores, 2021

Os dados dados da tabela 1 foram levantados segundo informações dos desenvolvedores.

2.2 DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

Dispositivos móveis são todos aqueles dispositivos “de bolso”, geralmente equipados com uma pequena tela e possivelmente um teclado. O desenvolvimento para dispositivos móveis ou desenvolvimento mobile, consiste no desenvolvimento de software para estes tipos de dispositivos.

No geral, software para dispositivos móveis são aplicações que exigem menos capacidade de processamento (principalmente por estarem em dispositivos menos robustos) mas conseguem entregar um valor muito grande, pois são capazes de solucionar problemas que são possibilitados devido a “mobilidade” dos dispositivos.

Atualmente, no mercado, os principais sistemas operacionais que servem como plataforma para os softwares *mobile* são: Android, iOS, Windows (Mobile) e BlackBerry.

Na grande maioria dos casos, os softwares *mobile* são feitos para smartphones. Após desenvolvimento, são disponibilizados em lojas específicas de cada sistema operacional e as lojas, por sua vez, cuidam da distribuição dos softwares.

2.2.1 Plataformas de Desenvolvimento para Dispositivos móveis

Dentre as principais plataformas, as mais utilizadas são o Android e iOS.

O Android é um sistema operacional baseado em Linux, pertencente ao Google. Muitos dispositivos hoje em dia utilizam o Android - *smartwatches*, smartphones, *smarttv's*, sistemas de automóveis, etc. A principal vantagem dessa plataforma é ser um sistema operacional de código aberto.

Esse sistema é muito buscado por quem deseja sistemas de baixo custo, muito personalizáveis e que possa ser rodado em dispositivos de alta performance.

O iOS é um sistema operacional da Apple. Ele é um sistema operacional de código fechado e mais caro para que se possa desenvolver nativamente. Isso porque para desenvolver softwares para iOS, você precisa utilizar um *macbook* (notebook da Apple), e as taxas publicações na loja da Apple (Apple Store) é mais caro.

2.2.2 Arquiteturas de software

No desenvolvimento de *software* para dispositivos móveis, tem-se duas principais arquiteturas de desenvolvimento a serem utilizadas, sendo elas: modelo cliente-servidor e aplicação monolítica. Ambas arquiteturas são úteis e não necessariamente uma é melhor que a outra, contudo, a escolha entre uma ou outra depende da solução à qual se deseja atingir.

2.2.2.1 Modelo cliente-servidor

Este modelo de arquitetura consiste em uma aplicação “cliente” em uma plataforma específica para prover a interface e input e output de dados para um usuário e uma aplicação “servidor”, a qual através de webservices disponibilizará acesso a banco de dados e irá realizar os processamentos e lógicas necessárias para o funcionamento da aplicação.

A grande vantagem deste tipo de arquitetura é poder utilizar um servidor para realizar processamentos pesados e, além disso, conseguir ter um ou mais bancos de dados, centralizados em um servidor, para armazenar informações de usuários de forma segura, sem que esses dados possam ser perdidos em uma eventual desinstalação do software cliente ou encerramento da aplicação.

A grande desvantagem deste tipo de arquitetura é uma dificuldade em controlar e garantir compatibilidade entre o servidor e diferentes versões de aplicações cliente.

2.2.2.2 Modelo aplicação monolítica

A aplicação monolítica é um software desenvolvido para ser capaz de realizar todo o fluxo necessário para a conclusão de uma operação. Ela é capaz de exibir a interface ao usuário, receber o input dos dados, realizar processamentos, armazenar diretamente informações no banco de dados e recuperá-las.

3 MÉTODO DE PESQUISA

Nesta seção, é apresentado o tipo de pesquisa desta monografia, apresentando as etapas que servirão como base para a conclusão da pesquisa.

3.1 DEFINIÇÃO DA PESQUISA

Segundo Gil (2002, p. 17), “Pode-se definir pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos”.

a) Pesquisa básica: A pesquisa básica “permite articular conceitos e sistematizar a produção de uma determinada área de conhecimento”, diz Minayo (2002, p. 52, apud Zanella, 2013, p. 32). Também chamada de pesquisa teórica, Trujillo Ferrari (1982), traz que esse tipo de pesquisa busca contribuir, entender, explicar fenômenos e possivelmente gerar novas teorias.

b) Pesquisa aplicada: Sobre pesquisa aplicada, Trujillo Ferrari (1982, p. 171) diz que “não obstante a finalidade prática da pesquisa, ela pode contribuir teoricamente com novos fatos para o planejamento de novas pesquisas ou mesmo para a compreensão teórica de certos setores do conhecimento”, reforçando o conceito de pesquisa aplicada como uma forma de gerar conhecimentos acerca da solução de um problema.

c) Pesquisa quantitativa: Segundo Zanella (2013, p. 35) a pesquisa quantitativa visa coletar e tratar dados para gerar resultados passíveis de medição e análise, que consigam indicar um tipo quantificação.

d) Pesquisa qualitativa: Para Vieira (1996), a pesquisa qualitativa baseia-se em uma pesquisa que não utiliza estatística para avaliação dos dados.

No desenvolvimento deste trabalho será utilizada pesquisa aplicada, pois visa produzir um conhecimento para aplicação prática, com objetivo de solucionar o problema de falta de conexão com a internet, na utilização de aplicações que necessitam de acesso ao servidor para resgatar e armazenar dados. Como não utilizaremos estatísticas para avaliação,

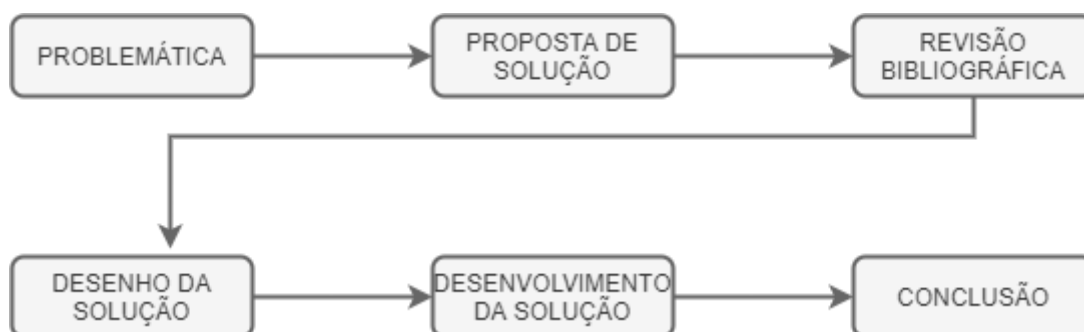
se qualifica como pesquisa qualitativa. Por ser elaborada através de referencial teórico como livros, artigos e publicações da Internet, é considerada uma pesquisa bibliográfica.

3.2 ETAPAS METODOLÓGICAS

As etapas para cumprir o objetivo deste trabalho estão organizadas da seguinte maneira:

- a) Definição da problemática: levantamento do problema que será trabalhado;
- b) Definição da proposta de solução: solução encontrada para a problemática levantada;
- c) Revisão Bibliográfica: explicação dos conceitos que serão utilizados no desenvolvimento deste trabalho.
- d) Desenho da Solução: apresentar a solução com os mínimos detalhes
- e) Desenvolvimento da solução: prática da solução aplicada
- f) Conclusões e trabalhos futuros: finalizações.

Figura 1 – Fluxograma das etapas metodológicas

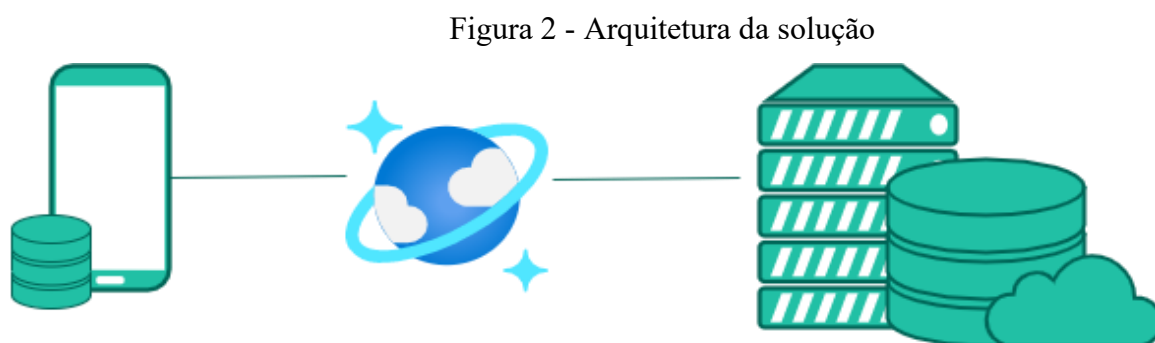


fonte: Autores, 2020.

No fluxograma da figura 1 é demonstrado de forma mais objetiva a estrutura da monografia, e fica mais clara as etapas desenvolvidas.

3.3 ARQUITETURA DA SOLUÇÃO

Na figura 2, visualiza-se a proposta de solução desenvolvida para o problema.



fonte: Autores, 2020.

O armazenamento dos dados é feito em um servidor sempre que possuir rede de internet disponível, caso contrário os dados são armazenados no próprio aparelho que possui a aplicação. Assim que a conexão for restabelecida, os dados são sincronizados com o banco de dados do servidor com apenas um clique do usuário. Desta forma a funcionalidade da aplicação não é afetada pela falta de rede.

3.4 DELIMITAÇÕES

Na elaboração propósito deste trabalho, foram definidas algumas delimitações, descritas a seguir:

- a) Consideram-se questões de segurança muito importantes, no entanto não serão abordadas, por não ser o foco deste trabalho.
- b) Será utilizado o banco de dados SQLite como banco de dados embarcado, por questões de afinidade dos autores.

c) Para o desenvolvimento da aplicação que terá o SQLite embarcado será utilizado o Android como plataforma.

d) Não serão abordados quesitos de rede e sinal.

4 DESENVOLVIMENTO DA SOLUÇÃO

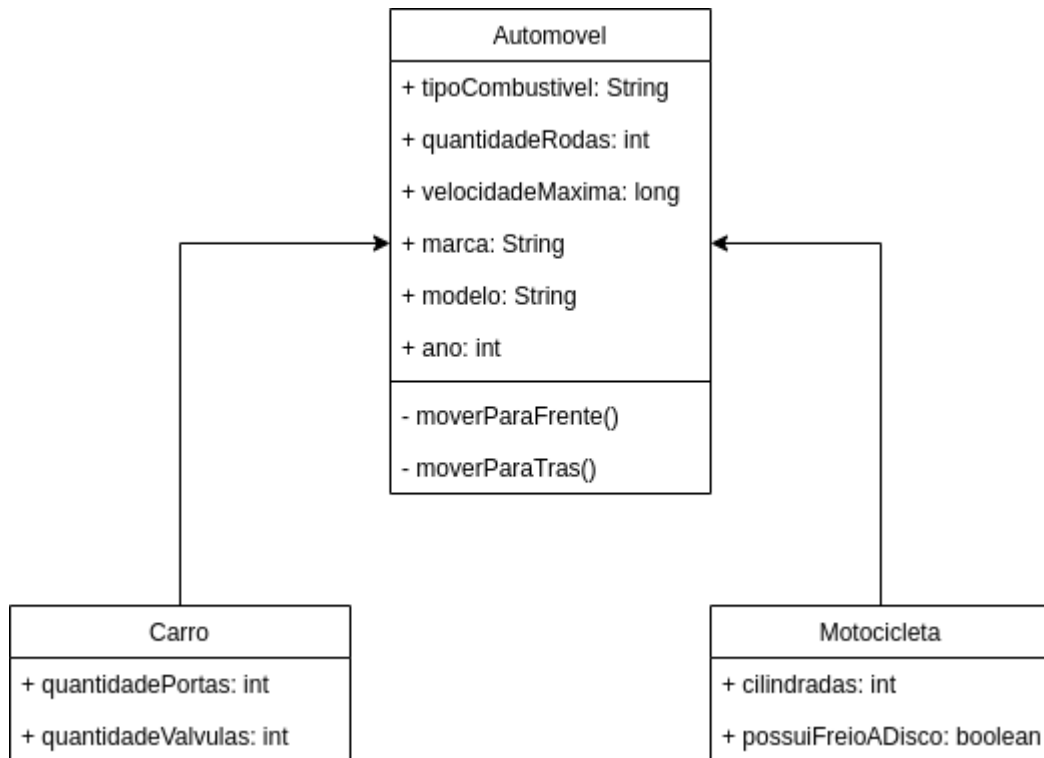
Nesta seção, será apresentada a solução aderida com objetivo de demonstrar uma arquitetura de banco de dados embarcado em dispositivos móveis. Será abordado UML, metodologia Iconix dando ênfase na primeira fase a qual será utilizada neste trabalho, especificação do protótipo, requisitos, protótipos de telas, casos de uso, modelo de domínio e por fim modelos de dados.

4.1 UML

Segundo Bezerra (2015, p. 26), “a UML é uma linguagem visual para modelar sistemas orientados a objetos. Isso quer dizer que a UML é uma linguagem que define elementos gráficos (visuais) que podem ser utilizados na modelagem de sistemas.”.

Basicamente o UML é a junção de outras notações preexistentes, com início em 1994, fruto da junção de Rurnbaugh e Booch. Após um ano de trabalho, foi lançada em 1995 a versão 0.8 do UML. Em 1996 foi lançada a versão 0.9, desta vez englobando o método OOSE.

Figura 3 - Diagrama de classes



Fonte: Elaborado pelos autores, 2021.

Existem diversos tipos de UML. O UML acima é do tipo diagrama de classes, e será utilizado para desenvolver o aplicativo deste trabalho.

4.2 ICONIX

Segundo Sbrocco e Macedo o Iconix é uma metodologia ágil, que foi criada em 1999 e publicada oficialmente em 2005 por Doug Rosenberg, fundador e presidente da Iconix Software Engineering Inc., juntamente com Matt Stephens e Mark Colling-Cope, ambos especialistas em projetos de desenvolvimento ágil, e com larga experiência em desenvolvimento, arquitetura, interação e design de softwares.

O Iconix é considerado uma metodologia ágil, simples e prática se comparada às metodologias como XP (Extreme Programming) e RUP (Rational Unified Process). A simplicidade desta metodologia não pode ser confundida com fraqueza, pois ela é muito

poderosa e eficaz quando se trata de componente de análise e representação dos problemas sólidos.

Iconix diz respeito a uma modelagem dirigida por casos de uso (use case driven), que tem como objetivo estudar e comunicar o comportamento do sistema sob o ponto de vista de um consumidor ou usuário final. Embora tenha sido criada antes da UML, utiliza-se de um de seus diagramas, que permite a compreensão da solução proposta por todas as pessoas, mesmo que não envolvidas no processo de desenvolvimento. (SBROCCO, MACEDO, p. 173, 2012).

Veremos a seguir algumas fases da metodologia Iconix e sua estrutura de funcionamento.

a) Modelagem do domínio: Nesta fase é identificado o domínio do problema e seus objetos. Para Sbrocco e Macedo (2012, p.174), “Os desenvolvedores devem identificar e abstrair os objetos do domínio de aplicação existentes no mundo real, seus relacionamentos, generalizações e agregações. Nesse momento, pode-se começar a criar um diagrama de classes de alto nível”. E então iniciar o desenvolvimento de um protótipo rápido do sistema.

Quando essa etapa terminar, um modelo razoavelmente correto do domínio deve ter sido construído, e todos os envolvidos no projeto (cliente, programadores, analistas, verificadores e mesmo usuários finais) poderão compartilhar e usar comumente o sistema prototipado. (SBROCCO e MACEDO, 2012, p.174).

O modelo do domínio continuará a evoluir e crescer, até que atinja os resultados esperados no domínio do negócio.

b) Comportamento dos requisitos por meio de casos de uso: Aqui os requisitos captados são interpretados e revisados, então se utiliza o diagrama de caso de uso da UML para sua representação. Com isso temos os benefícios de ter uma melhor visualização dos elementos como: qualquer pessoa terá facilidade em entender os processos, divisão das atividades, possibilitando a criação de diagrama de pacotes, documentação do sistema através desse diagrama, entre outros. Brocco e Macedo (2012, p.174).

c) Análise de robustez: Conforme Brocco e Macedo (2012, p.175), “Ela é feita por um diagrama de robustez que, conforme mencionado representa um diagrama de classes estereotipado. Essa fase conecta a análise ao projeto, certificando-se de que a descrição dos casos de uso esteja correta”.

Foram citadas apenas algumas fases do iconix, dando ênfase à primeira fase, Modelagem de Domínio a qual será aplicada nesta monografia, e outras fases consideradas também consideradas importantes para o desenvolvimento deste trabalho.

4.3 ESPECIFICAÇÃO DO PROTÓTIPO

Como protótipo, será desenvolvido um aplicativo *mobile* para fábricas de rastreadores, permitindo o controle de itens enviados por pedido, far-se-á uso de leitores de código de barras no telefone, para imputar os códigos únicos de cada rastreador. Serão salvas as informações no banco de dados do aparelho, quando houver conexão com a internet, o sincronismo das informações será feito no banco de dados do servidor na nuvem, com um clique do usuário no aplicativo solicitando o sincronismo.

4.3.1 Requisitos

Segundo Sommerville (2018, p.85), “os requisitos de um sistema são as descrições dos serviços que o sistema deve prestar e as restrições a sua operação”. Os requisitos atendem a um determinado propósito e necessidades do cliente, como por exemplo, fazer um pedido ou realizar uma consulta de algum dado.

Nesta etapa, são apresentados os requisitos funcionais, não funcionais e regras de negócio do protótipo proposto como solução.

4.3.2 Requisitos funcionais

“Os requisitos funcionais de um sistema descrevem o que ele deve fazer. [...] Os requisitos funcionais devem ser descritos de modo compreensível para os usuários e gerentes do sistema”. Sommerville (2018, p. 89).

Sendo assim, estão representados no Quadro 2 a seguir, os requisitos funcionais do protótipo:

Quadro 2 - Requisitos funcionais

Identificação	Requisito
RF001	O sistema deve permitir o uso sem a necessidade de autenticação
RF002	O sistema de sincronizar os pedidos do servidor sempre que for aberto
RF003	O sistema deve permitir visualizar os pedidos já processados
RF004	O sistema de permitir a inserção manual dos itens do pedido
RF005	O sistema deve permitir fazer a inserção usando um leitor de código de barras
RF006	O sistema deve salvar internamente os dados no dispositivo caso não possua conexão com a internet
RF007	O sistema deve sincronizar os dados armazenados com o servidor assim que restabelecer a conexão com a internet

Fonte: Elaborado pelos autores, 2021

Na próxima seção é feito uma breve explicação sobre requisitos não funcionais e apresentado os requisitos não funcionais do protótipo.

4.3.3 Requisitos não funcionais

Para Sommerville (2018, p. 91), “os requisitos não funcionais, [...] são aqueles que não possuem relação direta com os serviços específicos fornecidos pelo sistema aos seus usuários. Esses requisitos não funcionais normalmente especificam ou restringem as características do sistema como um todo.”

Ainda conforme Sommerville (2018) está relacionados com propriedades do próprio sistema, como confiabilidade, uso de memória e/ou tempo de resposta. Também pode

definir restrições ou limitações ao sistema, como por exemplo, a capacidade máxima de dispositivos de entrada e saída.

A seguir, no Quadro 3 estão representados os requisitos não funcionais do sistema:

Quadro 3 - Requisitos não funcionais

Identificação	Requisito
RNF001	O sistema deve funcionar mesmo sem conexão com a internet
RNF002	O sistema deve utilizar o banco de dados Sqlite como banco embarcado
RNF003	O sistema deve garantir a integridade dos dados quando houver falhas na conexão
RNF004	O sistema deve ter acesso a internet pelo menos uma vez ao dia para sincronizar os pedidos
RNF005	O sistema deve ser executado apenas no sistema operacional Android
RNF006	O sistema deve ter uma interface amigável e bem intuitiva

Fonte: Elaborado pelos autores, 2021

4.3.4 Regras de negócio

As regras de negócio estão presentes em qualquer desenvolvimento, pois se baseia nas operações e definições do negócio ou certo processo. Conforme Kamada (2006, p. 34), “uma regra que pode ser interpretada por computadores, que define ou restringe alguns aspectos de um negócio, introduzindo obrigações ou necessidades, conforme as políticas da organização”.

Foram desenvolvidas no Quadro 4 algumas regras de negócio para o protótipo

Quadro 4 - Regras de negócio

Identificação	Requisito
RN01	O usuário deve inserir os itens do pedido apenas com um leitor de código de barras
RN02	O usuário deve informar a quantidade de itens informada no pedido

4.3.5 Protótipo de tela

Nesta seção são apresentados os protótipos de tela, demonstrando de forma básica o layout de cada visualização. Os protótipos aqui presentes são de média fidelidade, ou seja, a tela definitiva poderá ter mais ou menos elementos e/ou modificações.

A figura 4 demonstra a primeira tela a ser visualizada pelo usuário ao entrar no aplicativo. Nesta tela será possível escolher entre as opções de nova leitura de pedido, onde o usuário pode escolher um pedido para fazer a leitura dos códigos dos itens ou verificar pedidos, onde será possível verificar o status do pedido.

Figura 4 - Protótipo da tela inicial do aplicativo



Fonte: Elaborado pelos autores, 2021.

A figura 5 demonstra o protótipo de tela onde é possível escolher um pedido e fazer a leitura do código de cada um, incluindo itens a um pedido.

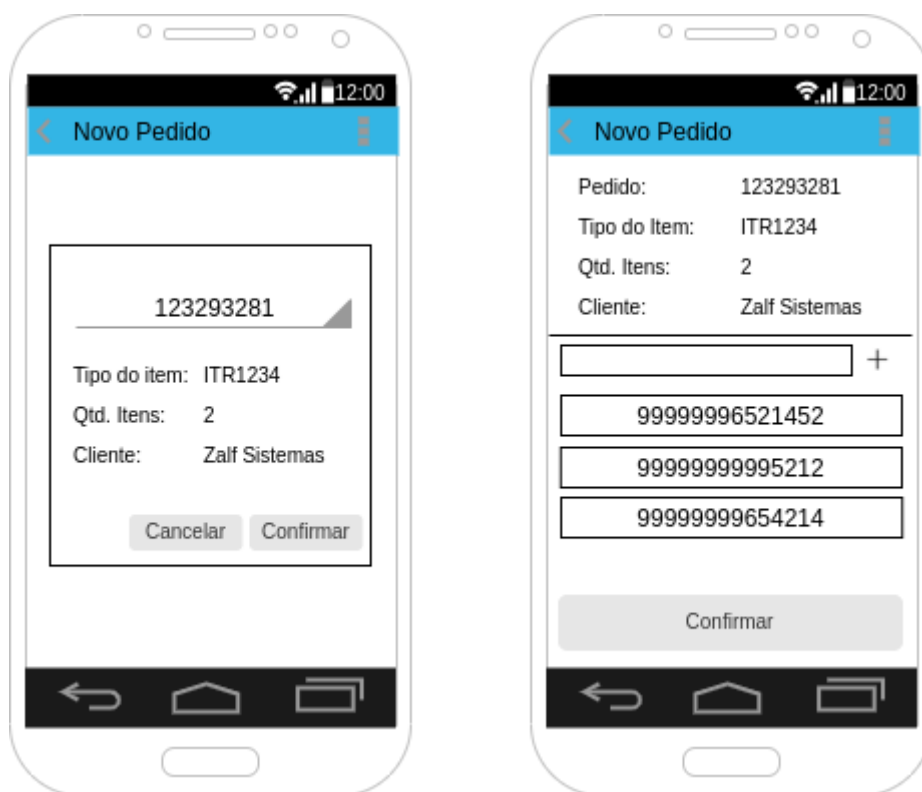
Ao abrir essa tela através da opção de *menu* da figura anterior, será exibido um *dialog*, conforme a imagem da esquerda, que permite o usuário selecionar um pedido (clicando no primeiro campo) e após selecionado, mostra algumas informações pertinentes a este pedido.

Quando confirmado, o *dialog* é escondido e os elementos de tela principais são exibidos, conforme imagem da direita.

As informações pertinentes do pedido são propagadas para esses elementos principais e exibidos um campo para input de um código de item ou leitura através de leitor de código de barras. Se for imputado o código manualmente, ao clicar no botão de “+” à direita do campo, o código de barras é armazenado. Se usado o leitor, após a leitura o próprio sistema irá “clicar” no botão de “+” e armazenar o código.

Ao confirmar, são executadas validações e se houver algum erro, o usuário recebe uma mensagem de erro descritiva. Se não houver nenhum erro, o usuário recebe uma mensagem de confirmação e é redirecionado para a tela inicial.

Figura 5 - Tela para leitura de itens de um novo pedido



Fonte: Elaborado pelos autores, 2021.

A figura 6 demonstra o protótipo da tela de acompanhamento do status dos pedidos. Nela é possível o usuário verificar se um pedido já foi sincronizado ou ainda encontra-se pendente de sincronização.

Figura 6 - Tela para acompanhamento do status dos pedidos



The image shows a smartphone screen with a status bar at the top displaying signal strength, Wi-Fi, and the time 12:00. Below the status bar is a blue header bar with a back arrow on the left and the text 'Pedidos Processados' in the center. To the right of the header is a three-dot menu icon. The main content area contains a table with three columns: 'Código', 'Qtd. Itens', and 'Status'. The table has four data rows. At the bottom of the screen is a black Android navigation bar with back, home, and recents icons.

Código	Qtd. Itens	Status
1234	10	Pendente
3214	5	Processado
5321	7	Processado
1234	14	Processado

Fonte: Elaborado pelos autores, 2021.

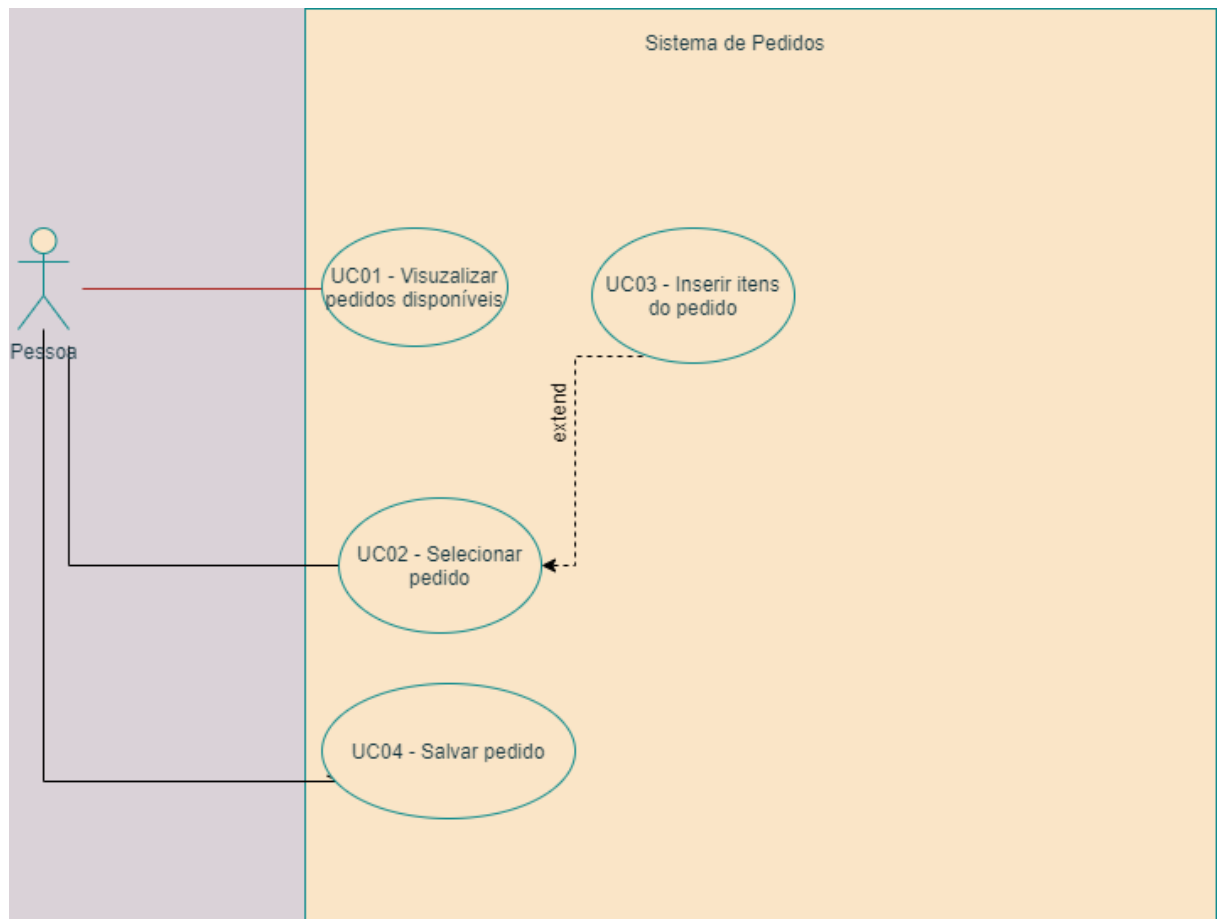
Apresentado os protótipos, a seção seguinte apresenta a definição de casos de uso que foi desenvolvido para esta monografia.

4.3.6 Casos de uso

Para Pfleeger (2004, p. 136) "os casos de uso dividem o sistema em um conjunto de partes lógicas, minimamente relacionadas; cada uma das quais descrevendo um modo de funcionamento do sistema." Ainda na visão de Pfleeger, um caso de uso descreve uma funcionalidade específica de um sistema. Pode-se usar como exemplo, descrever a interação entre um cliente de um banco e um caixa automático, para retirar dinheiro de uma conta. A entidade que interage com o sistema é chamada de ator, e pode ser um usuário, um dispositivo ou outro sistema.

Sendo assim, na Figura 7 é visualizado o caso de uso referente à funcionalidade de captação de itens de um pedido.

Figura 7 - Diagrama de caso de uso do sistema proposto



Fonte: Elaborado pelos autores, 2021

A seguir, encontra-se o Quadro 5, apresentando a descrição do caso de uso UC01.

Quadro 5 - Caso de uso UC01

UC01	Visualizar pedidos disponíveis
Objetivo	Visualizar os pedidos disponíveis
Pré-condições	O sistema deve estar sincronizado com o servidor
Pós-condições:	O ator irá visualizar os pedidos disponíveis com a condição de escolher qual pedido será iniciado
Requisitos funcionais:	RF002 - O sistema de sincronizar os pedidos do servidor sempre que for aberto
Fluxo Principal	Passo 1 - O ator abre o sistema Passo2 - O ator navega até a aba de pedidos Passo 3 - Visualiza os pedidos disponíveis

Fonte: Elaborado pelos autores, 2021

No Quadro 6, são apresentados as condições e fluxo principal do caso de uso UC02.

Quadro 6 - Caso de uso UC02

UC02	Selecionar pedidos
Objetivo	Selecionar um pedido
Pré-condições	O ator precisa visualizar e escolher um pedido
Pós-condições:	O ator precisa clicar no pedido escolhido
Requisitos funcionais:	RF008 - O sistema deve permitir selecionar um pedido para inserir os itens
Fluxo Principal	Passo 1 - O ator Escolhe o pedido Passo 2 - O ator seleciona um pedido

Fonte: Elaborado pelos autores, 2021

No Quadro 7, são apresentados as condições, fluxo principal e fluxo alternativo do caso de uso UC03.

Quadro 7 - Caso de uso UC03

UC03	Inserir itens do pedido
Objetivo	Informar os itens que pertencem ao pedido
Pré-condições	O ator precisa ter selecionado um pedido e ter os produtos a serem inseridos

Pós-condições:	bipar ou preencher manualmente os itens
Requisitos funcionais:	RF004 - O sistema de permitir a inserção manual dos itens do pedido RF005 - O sistema deve permitir fazer a inserção usando um leitor de código de barras
Fluxo Principal	Passo 1 - O ator disponibiliza sobre uma mesa os itens do pedido Passo2 - O ator clica no campo de adicionar item Passo3 - O ator lê o código de barras de cada item Passo4 - O sistema inseri em sequência em uma lista
Fluxo alternativo A	Passo1 - No passo 1 do fluxo principal, o ator pega item por item Passo2 - O ator clica no campo de adicionar Passo3- O ator digita manualmente os itens Passo 4 - O sistema inseri em sequência em uma lista

Fonte: Elaborado pelos autores, 2021

No Quadro 8, são apresentados as condições, fluxo principal e fluxo alternativo do caso de uso UC04

Quadro 8 - Caso de uso UC04

UC04	Salvar pedido
Objetivo	Salvar os itens inseridos do pedido
Pré-condições	O ator precisa ter lido os itens
Pós-condições:	O ator deve clicar em "confirmar"
Requisitos funcionais:	RF006 - O sistema deve salvar internamente no dispositivo os dados caso não possua conexão com a internet RF007 - O sistema deve sincronizar os dados armazenados com o servidor assim que restabelecer a conexão com a internet

Fluxo Principal	<p>Passo 1 - O sistema verifica se há conexão com a internet</p> <p>Passo2 - O sistema envia os dados do pedido para o servidor</p> <p>Passo3 - O sistema retorna uma mensagem de sucesso</p> <p>Passo4 - O sistema redireciona para página inicial</p>
Fluxo alternativo A	<p>Passo1 - No passo 1 do fluxo principal, o sistema identifica que não há conexão com a internet</p> <p>Passo2 - O sistema salva no banco local (Sqlite).</p> <p>Passo3- O sistema retorna mensagem de sucesso</p> <p>Passo 4 - O sistema redireciona para página inicial</p> <p>Passo5 - O sistema em paralelo verifica se a conexão com a internet foi restabelecida e sincroniza os pedidos inseridos no servidor</p>

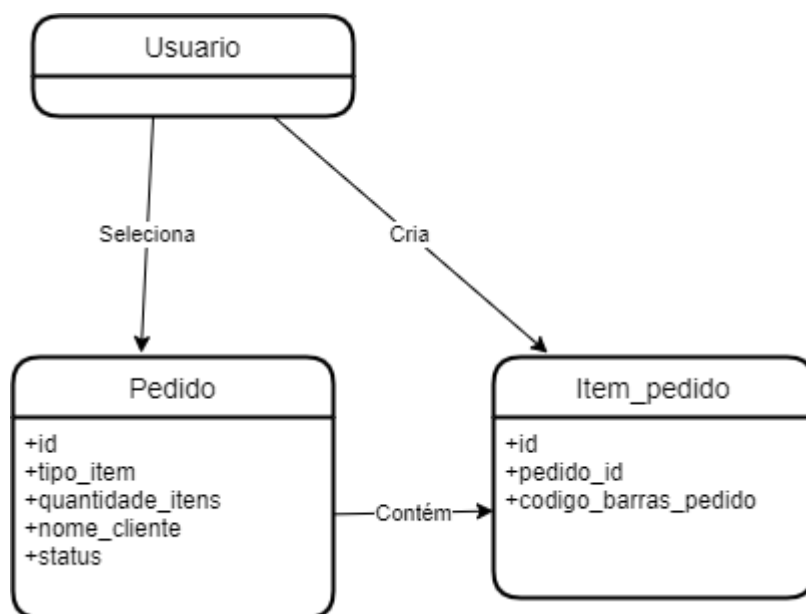
Fonte: Elaborado pelos autores, 2021

4.3.7 Modelo de domínio

Sommerville (2011) diz que o principal interesse do protótipo é identificado pelos modelos de domínio.

Para o protótipo apresentado nos tópicos anteriores, foi criado o modelo de domínio apresentado na figura 8.

Figura 8 - Domínio problema



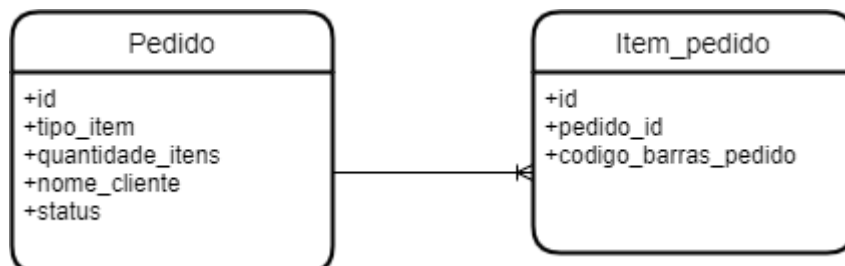
Fonte: Elaborado pelos autores, 2021

Na próxima seção é apresentado o modelo de dados proposto para o banco de dados local.

4.4 MODELO DE DADOS

A figura 9 demonstra o diagrama ER, com o modelo das tabelas do banco de dados local do aplicativo.

Figura 9 - Diagrama Entidade Relacionamento do banco de dados local



Fonte: Elaborado pelos autores, 2021.

No próximo capítulo, inicia-se o desenvolvimento da solução proposta nesta monografia.

5 DESENVOLVIMENTO

Neste capítulo, são apresentadas as ferramentas e linguagens utilizadas para o desenvolvimento do protótipo, tal como histórico do que foi feito e do resultado final do mesmo. Em síntese é demonstrado avaliação da experiência do usuário

5.1 FERRAMENTAS

Para o desenvolvimento do protótipo, os autores optaram por ferramentas comuns, conhecidas no mercado, que facilitam o desenvolvimento e que já são de conhecimentos dos mesmos. Nesta seção são apresentadas as ferramentas utilizadas durante a execução do protótipo.

Na figura 10 estão presentes as logomarcas das ferramentas utilizadas e que estão descritas logo a-baixo.

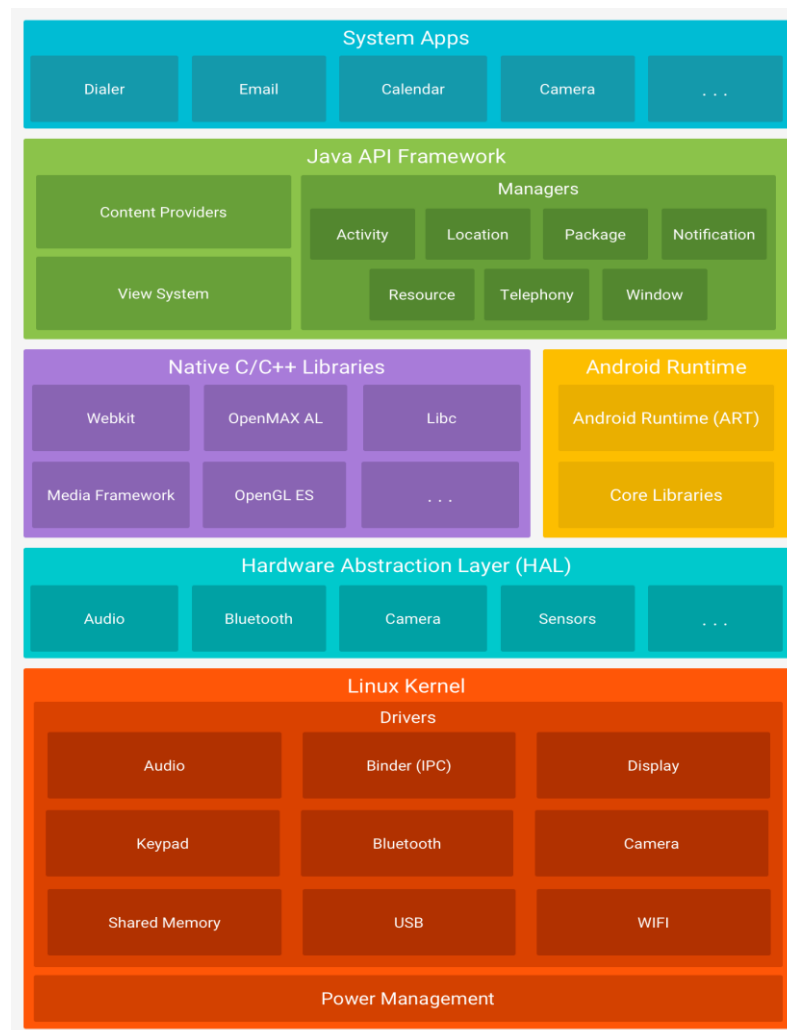
Figura 10 - Ferramentas



Fonte: Elaborado pelos autores, 2021

- Android: O Android é uma plataforma de código aberto, baseado em Linux e criado para atender uma grande variedade de dispositivos. Na figura 11 a seguir mostra os principais componentes da plataforma.

Figura 11 - Componentes do Android



Fonte: Google, 2021

No mais baixo nível está o Kernel do Linux como base do Android, permite que o Android aproveite recursos segurança, e facilitando o desenvolvimento de drivers por se tratar de algo conhecido. Na camada de abstração de hardware fornece interfaces padrões que expõem os recursos de hardware do dispositivo para a estrutura de API Java de nível superior. Consiste em vários módulos de biblioteca, cada um dos quais implementa uma interface para um tipo específico de componente de hardware, como a câmera ou o módulo *bluetooth*. Para a camada de Android Runtime, possibilita que cada aplicativo seja executado em sua própria instância do Android Runtime, trazendo alguns benefícios como, coleta de lixo otimizada, melhor suporte de depuração entre outros. Também é composto por algumas bibliotecas nativas da linguagem C/C++. Todo o conjunto de recursos do sistema operacional Android está disponível para desenvolvedores por meio de APIs escritas na linguagem Java. E por fim o

Android vem com um conjunto de aplicativos básicos para e-mail, mensagens SMS, calendários, navegação na Internet, contatos e muito mais. (Google, 2021).

- GitHub: É uma plataforma utilizada para manter repositórios de códigos de forma remota. Esta plataforma é mundialmente utilizada por mais de 65 milhões de desenvolvedores, devido suas facilidades em compartilhar e contribuir projetos tanto públicos (de código aberto), quanto privados ou corporativos. (GitHub, 2021).
- IntelliJ IDEA: É um ambiente integrado para desenvolvimento que traz diversas facilidades na hora de desenvolver softwares. Desenvolvido pela empresa JetBrains e escrito na linguagem Java, foi utilizado nesta monografia para desenvolver o código do servidor responsável por receber as alterações do aplicativo. (JetBrains, 2021).
- Android Studio: Ferramenta de desenvolvimento integrado oficial para o sistema operacional Android, construído no software IntelliJ IDEA e projetado especificamente para o desenvolvimento Android com um editor de código inteligente, sistema de construção flexível e emuladores em tempo real. (Google, 2021).
- GIT: é um sistema de versionamento de código, distribuído, gratuito e de código aberto, projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência. (GIT, 2021).
- Heroku: Heroku é uma plataforma de nuvem como serviço que suporta várias linguagens de programação. (Heroku, 2021). Foi utilizada para hospedar o servidor que recebe os dados do protótipo desenvolvido.
- Heroku Postgres: Banco de dados confiável e poderoso como serviço baseado em PostgreSQL. (Heroku, 2021). Este banco de dados foi utilizado exclusivamente para manter os pedidos que serão sincronizados com o protótipo.
- SQLite: SQLite é uma biblioteca em linguagem C que implementa um mecanismo de banco de dados SQL **pequeno , rápido , independente , de alta confiabilidade e recursos completos** . SQLite é o mecanismo de banco de dados **mais usado** no mundo. O SQLite está integrado em todos os telefones celulares e na maioria dos computadores, e vem empacotado em inúmeros outros aplicativos que as pessoas usam todos os dias (SQLite, 2021). Por todos os pontos citados acima os autores desta monografia, optaram por utilizar o Sqlite como banco de dados embarcado.

Na próxima seção encontram-se as linguagens utilizadas para execução do protótipo.

5.2 LINGUAGENS

Nesta monografia foram utilizadas linguagens robustas e de conhecimento dos autores. A seguir é apresentada a linguagem Java e linguagem SQL.

5.2.1 Java

Uma das linguagens de programação mais amplamente usadas, Java é usada como a linguagem do lado do servidor para a maioria dos projetos de desenvolvimento de back-end, incluindo aqueles que envolvem big data e desenvolvimento Android (Oracle University, 2019).

Figura 12 - Logomarca Java



Fonte: Brandslogos, 2021

O Java é uma linguagem de programação de alto nível, baseada em classes e orientada a objetos.

5.2.2 SQL

O SQL é uma linguagem usada para executar comando em bancos de dados relacionais e possui algumas características da álgebra relacional.

5.3 HISTÓRICO DO DESENVOLVIMENTO

A proposta dos autores desta monografia veio a partir de uma palestra onde os mesmos participaram em 2019, chamado Floripa Tech Day, e um dos temas palestrados foi sobre banco de dados. Então o palestrante levantou a seguinte questão: Qual o banco de dados mais utilizado no mundo? Todos os participantes apostaram nos grandes nomes, como Oracle ou Mysql, mas a resposta foi o banco de dados SQLite. E daí surgiu o interesse sobre esse banco onde se pode considerar que grande parte da população carrega no bolso. Os autores decidiram se aprofundar nas questões de banco de dados para embarcado e desenvolver um protótipo demonstrando o funcionamento e utilidade do modo offline para os usuários.

Eram amplas as opções de desenvolvimentos para esta solução de falta de internet e necessidade de armazenamento. O protótipo escolhido foi criar um aplicativo *mobile* para fábricas de rastreadores, permitindo o controle de itens enviados por pedido. Como definição de arquitetura, no primeiro momento é realizado o salvamento no banco de dados do aparelho e por fim o usuário com um clique faz o sincronismo com o banco de dados na nuvem, caso não haja internet é necessário esperar até que a conexão com a rede retorne, e então fazer o sincronismo.

Para iniciar o projeto do protótipo, foi levantada toda a modelagem de dados necessária como, os requisitos funcionais, requisitos não funcionais, casos de uso e o modelo de domínio referente à proposta da funcionalidade sugerida pelos autores. E então iniciou-se a codificação do aplicativo, devido curto tempo, algumas definições foram alteradas de última hora e ficaram como melhorias e trabalhos futuros.

Como já era de conhecimento dos autores as tecnologias sugeridas para o protótipo, ficou simples a codificação por mais que tenha sido trabalhosa. Os principais problemas

encontrados neste processo como um todo, foram redefinições de algumas partes do escopo pela falta de tempo.

Para apresentar o protótipo, alguns levantamentos foram feitos, como por exemplo, o fato de os possíveis usuários que testariam o aplicativo, não possuírem as ferramentas necessárias para ler os códigos de barras (leitor de código de barras e cabo adaptador). E por este motivo a forma de demonstração foi após o aplicativo e servidor prontos, foi criado um vídeo de demonstração da funcionalidade e do uso no geral.

5.3.1 Telas e pontos chaves no desenvolvimento do código

As principais telas que atendem a funcionalidade do sistema serão apresentadas nos próximos tópicos, juntamente com os pontos chaves necessários para o funcionamento da função, como apresentação dos métodos responsáveis pelos salvamentos e sincronismos com os bancos de dados.

5.3.1.1 Configuração do Room

Para o desenvolvimento do protótipo foi utilizada a biblioteca Room para manipulação do banco de dados.

O Room é uma biblioteca que faz parte do JetPack do Android, responsável por auxiliar os desenvolvedores a trabalhar com o SQLite durante o desenvolvimento de aplicativos mobile.

Basicamente a ferramenta funciona criando uma abstração do banco de dados, através do uso de três camadas: entity, dao e database.

A entity é uma classe que representa uma tabela do banco de dados. Já a dao (Data Access Object) é uma abstração em forma de interface que permite a criação de clausulas para o banco de dados, como por exemplo select, update, delete e insert. Por fim o

database é uma classe que representa o banco de dados como um todo dentro do Java. A seguir é possível analisar a classe de configuração do Room.

Figura 13 - Classe de configuração do Room

```
package com.unisul.leitor.database;

import ...

@Database(entities = {PedidoEntity.class, ItemPedidoEntity.class}, version = 1, exportSchema = false)
public abstract class AppDatabase extends RoomDatabase {
    private static final String TAG = AppDatabase.class.getSimpleName();
    private static final Object LOCK = new Object();
    private static final String DATABASE_NAME = "leitordb";
    private static AppDatabase sInstance;

    public static AppDatabase getInstance(Context context) {
        if (sInstance == null) {
            synchronized (LOCK) {
                Log.d(TAG, msg: "Creating new database instance");
                sInstance = Room.databaseBuilder(context.getApplicationContext(),
                    AppDatabase.class, AppDatabase.DATABASE_NAME)
                    .allowMainThreadQueries()
                    .build();
            }
        }
        Log.d(TAG, msg: "Getting the database instance");
        return sInstance;
    }

    public abstract PedidoDao pedidoDao();

    public abstract ItemPedidoDao itemPedidoDao();
}
```

Fonte: Elaborado pelos autores, 2021

Na classe de configuração é definido quais as entidades existentes no projeto, nome do bando de dados entre outras configurações adicionais.

5.3.1.2 Tela de inicial

Não foi definido nenhum método de autenticação, levando em consideração que o objetivo deste trabalho é apresentar a funcionalidade de armazenamento de dados offline. Ao entrar no aplicativo o usuário tem a tela com a lista de pedidos e seus status, podendo rolar

para baixo e sincronizar com o servidor caso haja mais pedidos pendentes. Na figura 14 abaixo, esta representando o resultado final.

Figura 14 - Tela inicial



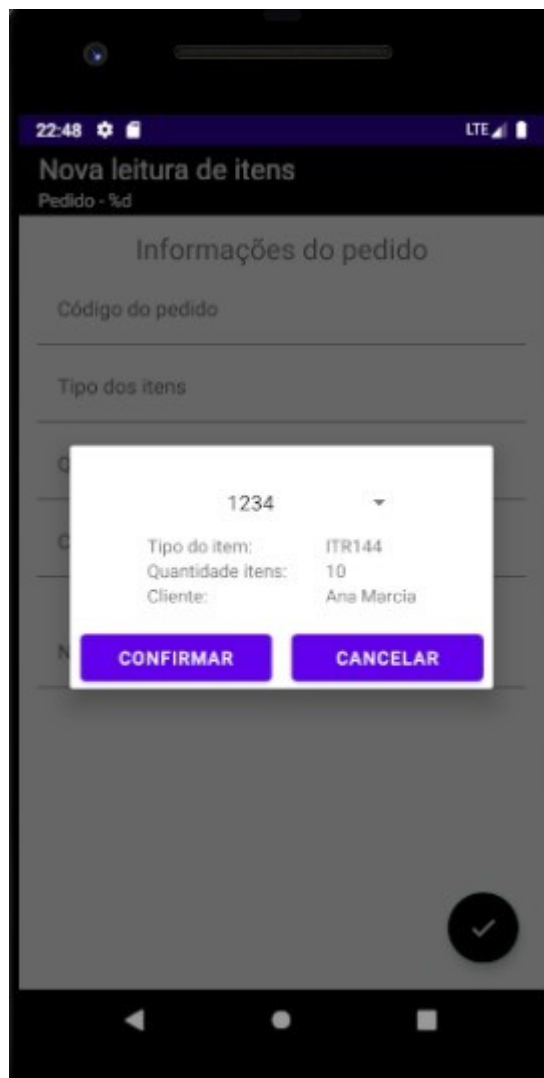
Fonte: Elaborado pelos autores, 2021

É possível visualizar os pedidos expedidos em uma lista com as seguintes colunas: número do pedido, quantidade de itens, status de andamento podendo ser **pendente** para pedidos ainda não lidos, ou **preenchido** para os que já foram lidos. Por fim o botão circular de sincronização, quando ainda não tiver sido feita a sincronização. Sendo substituído pelo ícone de checado na cor verde, quando a sincronização tiver sido feita com sucesso. Nesta mesma tela possui um botão com ícone de mais, para fazer a leitura dos códigos de barras, abrindo a próxima tela que será apresentada no próximo tópico.

5.3.1.3 Modal de seleção de pedidos pendentes

Este modal é acionado com o clique no botão com o ícone de mais da tela inicial. Neste momento é possível selecionar os pedidos pendentes e visualizar o detalhamento dos mesmos, e com a opção de cancelar e confirmar. Conforme a figura 15.

Figura 15 - Modal pedidos pendentes



Fonte: Elaborado pelos autores, 2021

Após selecionar um pedido e clicar em confirmar, abre a próxima tela para adicionar os itens com o leitor de códigos de barras.

5.3.1.4 Tela de nova leitura de pedido

Nesta tela é possível adicionar os itens do pedido no mesmo, utilizando um leitor de código de barras para ler os itens com mais facilidade. Basta selecionar o campo “Novo código de barras” e apontar o leitor para o item. A figura 16 a seguir possui mais detalhes da tela.

Figura 16 – Tela iniciar nova leitura do pedido



Fonte: Elaborado pelos autores, 2021

Ao finalizar a leitura dos itens, o usuário precisa clicar no botão de confirmar no canto inferior direito da tela, que automaticamente salva as informações obtidas no banco de dados SQLite e redireciona para tela inicial.

Ao retornar para tela inicial o pedido fica com status preenchido, aguardando a sincronização manual do usuário. Caso o usuário tente enviar sincronizar o pedido sem conexão com a internet aparece a seguinte mensagem na tela: sem conexão com a internet. Neste momento o usuário pode sair do aplicativo e retornar novamente quando houver uma conexão com a rede.

5.4 AVALIAÇÃO DO PROTÓTIPO

Nesta seção é apresentado o método de coleta de dados para avaliação do aplicativo e as perguntas elaboradas e feitas aos entrevistados.

O método de avaliação escolhido foi uma análise a partir de entrevistas com pessoas que utilizariam o aplicativo no seu dia a dia, e também desenvolvedores que conhecem sobre banco de dados e aplicativos móveis. Para demonstração do aplicativo, os autores criaram um vídeo da utilização do aplicativo e descrevendo o funcionamento. Foram formadas cinco perguntas, levando em consideração a problemática e objetivos deste trabalho, descritos a seguir:

Pergunta nº 1. Com o protótipo apresentado é possível adicionar os dados dos pedidos sem estar conectada a rede de internet?

Pergunta nº 2. Com a abordagem do sistema, é possível evitar retrabalhos, como anotar em planilhas e depois repassar para o sistema em casos de falta de internet?

Pergunta nº 3. Você já teve problemas para anotar informações importantes em aplicativos por falta de conexão com a internet?

Pergunta nº 4. Em se tratando da solução proposta pelos autores, de utilizar bancos de dados embarcados em aplicativos móveis para suprir o uso offline, a solução é bem-vinda e aplicável em outros cenários?

Pergunta nº 5. Qual cenário você levaria consideração no momento de desenvolver um aplicativo que possibilitasse a utilização off-line?

O total de entrevistas coletadas foram oito. Sendo possível avaliar se protótipo desenvolvido atendeu ao objetivo geral e se foi possível resolver as problemáticas. As entrevistas foram coletadas de forma que os entrevistados pudessem responder de forma aberta, podendo ser conferido a descrição de cada entrevistado e respostas no apêndice A.

5.5 ANÁLISE DOS RESULTADOS

Tendo em vista a problemática de falta de conexão de internet durante o uso de aplicativos, com o questionário é possível afirmar que o cenário utilizado e protótipo

desenvolvido cumpre os objetivos desta monografia de permitir o uso das funcionalidades mesmo sem acesso a internet e garantindo a consistência dos dados.

Com base na análise dos resultados do questionário aplicado fica definido que:

Na pergunta nº 1, foi possível definir que é possível adicionar os dados dos pedidos sem estar conectada a rede de internet.

Na pergunta nº 2, fica justificado que com o uso de banco de dados embarcados, é possível evitar retrabalhos, como anotar em planilhas e depois repassar para o sistema.

A pergunta nº 3, justifica a problemática e comprova que as pessoas tem dificuldade de utilizar funcionalidade dos aplicativos quando estão offline.

As respostas da pergunta nº 4, afirma-se que o uso de banco de dados embarcados pode ser aplicado em diversos cenários.

Na pergunta nº 5, obteve-se algumas ideias de cenários para utilizar banco de dados embarcados em aplicativos móveis, como por exemplo a resposta do Everton Silva Rosa, Diretor P&D Hardware e Operacional: “APP de transportadora, onde o entregador confirma a entrega realizada e coloca os dados de quem recebeu, se ele estiver em um local com sinal ruim de internet não impactaria o seu trabalho”.

Com a análise dos resultados da pesquisa, foi possível identificar que o uso de bancos de dados embarcados em aplicações para dispositivos móveis é capaz de solucionar problemas de conexão a internet.

6 CONCLUSÕES E TRABALHOS FUTUROS

Nesta seção é apresentada a conclusão desta monografia, e possíveis trabalhos futuros.

6.1 CONCLUSÃO

Após realizar as pesquisas sobre banco de dados embarcados como solução para suprir a falta de conexão durante o uso de aplicativos e aplicá-la em um sistema com o intuito de exemplificar a ajuda provida para um caso real, foi possível concluir que usando um banco de dados embarcados é possível o armazenamento de dados em qualquer momento, permitindo que esses dados sofram manipulações posteriores, quando for o melhor momento.

Além disso, a pesquisa realizada foi crucial para a análise e identificação que a solução proposta é eficaz e permite com que os usuários utilizem os aplicativos mesmo com imprevistos, podendo estes serem a falta de conexão com a internet ou a falta de energia elétrica impedindo o uso de WIFI, que muitas vezes podem impedir o uso do aplicativo para sua funcionalidade principal. As respostas também permitiram o entendimento de que o uso de banco de dados embarcados poderia solucionar problemas não só de falta de internet, mas também como falta de bateria ou outros problemas temporários de um dispositivo móvel.

Tendo em vista esses problemas e a importância de armazenamento de dados, a conclusão adotada pelos autores foi que utilizar um banco de dados embarcado seria a solução mais adequada. Portanto, alguns bancos de dados foram analisados, e de todos, foi escolhido o banco de dados SQLite para colocar a solução em desenvolvimento juntamente com a plataforma Android, e assim criado um protótipo que disponibiliza o acesso de pedidos incompletos para expedição e permitindo que sejam preenchidos com os códigos de barras dos itens mesmo sem o uso de internet e posteriormente sincronizado com o servidor, tornando-os aptos para expedição.

Com base nos resultados do questionário aplicado, foi possível identificar que todos entrevistados que fazem uso de aplicativos em dispositivos móveis sofrem ou sofreram

com a impossibilidade e/ou dificuldade do uso do sistema por conta de problemas com conexão.

Além disso, foi possível concluir que o protótipo apresentado consegue demonstrar que é possível fazer uso do aplicativo em sua função principal, que neste caso é adicionar itens em um pedido, fazendo a posterior sincronização dos dados, quando houver uma boa conexão da internet.

Na entrevista também foi abordada a possibilidade dessa mesma solução ser utilizada em outros cenários, além da leitura de itens de pedidos para expedição, e conseguimos entender com os resultados de que há sim outros cenários onde essa solução com bancos de dados embarcados seriam úteis.

Por fim, concluímos que existem alguns cenários que seriam importantes de serem considerados durante o desenvolvimento de um aplicativo que use bancos de dados embarcados para funcionamento offline.

6.2 TRABALHOS FUTUROS

Como melhorias no protótipo ficam definidas as seguintes questões:

- Complementar e melhorar layouts das telas pensando no UI/UX;
- Implementar o Lombok nas classes de modelo do aplicativo;
- Adicionar validades de quantidade de itens lidos versus quantidade de itens do pedido;
- Implementar o uso de databinding em todas as views do projeto;
- Fazer a sincronização automática quando houver conexão com a internet
- Implementar testes unitários para as classes e métodos utilitários do projeto;

7 REFERÊNCIAS

BANKMYCELL. How many smartphones are there in the world? October 2020 mobile user statistics: find out the number of phones in the world and smartphone penetration by country or region. **BankMyCell**, 2020. Disponível em: [https://www.bankmycell.com/blog/how-many-phones-are-in-the-](https://www.bankmycell.com/blog/how-many-phones-are-in-the-world#:~:text=According%20to%20GSM%20real%2Dtime,population%20has%20a%20mobile%20device)

[world#:~:text=According%20to%20GSM%20real%2Dtime,population%20has%20a%20mobile%20device](https://www.bankmycell.com/blog/how-many-phones-are-in-the-world#:~:text=According%20to%20GSM%20real%2Dtime,population%20has%20a%20mobile%20device). Acesso em: 04 out. 2020.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. - 3. ed. - Rio de Janeiro : Elsevier, 2015. Disponível em: https://www.academia.edu/35326870/Principios_De_Analise_E_Projeto_Eduardo_Bezerra_pdf. Acesso em: 10 Abril 2021.

BRANDSLOG. **Java Logo PNG (1)**. Disponível em: <https://brandslogos.com/j/java-logo-1/>. Acesso em: jun, 2021.

COSTA, Drielli Peres; SANTOS, Marilde Terezinha Prado. **Comparativo Entre Gerenciadores de Banco de Dados Para Aplicação Android**. Revista T.I.S. São Carlos, v. 4, n. 1, p. 20, jan-abr 2015.

COUTINHO, Gustavo Leuzinger. **A Era dos Smartphones**: Um estudo Exploratório sobre o uso dos Smartphones no Brasil. 2014. p. 5. Monografia (Publicidade e Propaganda) - Universidade de Brasília, UnB, Brasília, 2015.

ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.

BEAULIEU, Alan. **Aprendendo SQL**. São Paulo: Novatec, 2010.

GARRIDO, Katiuscia Andrade de Sousa. **Dispositivos móveis a evolução do celular: o mundo em nossas mãos**. 2013. p. 2. Monografia (Marketing e Comunicação Digital) - Instituto de Educação Superior de Brasília, IESB, Brasília, 2013.

GILL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. E-book. Disponível em:

http://www.uece.br/nucleodelinguasitaperi/dmdocuments/gil_como_elaborar_projeto_de_pesquisa.pdf. Acesso em: 10 dez. 2020.

GIT. **Git and Software Freedom Conservancy**. Disponível em: <<https://git-scm.com/sfc>>. Acesso em: 04 jun, 2021.

GITHUB. **Github**. Disponível em: <<https://github.com/>>. Acesso em: 04 jun, 2021.

GOOGLE. **Platform Architecture**. Disponível em: <https://developer.android.com/guide/platform>. Acesso em 04 jun, 2021.

GUERRA, Jhonatas. A História dos Aplicativos – Quem usa e quem vive de desenvolver. **Tribuna da Imprensa Livre**. Disponível em: <https://tribunadaimpressalivre.com/a-historia-dos-aplicativos-quem-usa-e-quem-vive-de-desenvolver/#:~:text=O%20avan%C3%A7o%20tecnol%C3%B3gico%20possibilitou%20o,praticidade%20ao%20cotidiano%20das%20pessoas>. Acesso em 04 out. 2020.

HEROKU. **Polyglot JVM Platform for modern apps**. Disponível em: <https://www.heroku.com/java#why-heroku>. Acesso em: jun, 2021.

JETBRAINS. **IntelliJ IDEA**. Disponível em: <https://www.jetbrains.com/idea/>. Acesso em: 04 jun, 2021.

KAMADA, Aqueo. **Execução de Serviços Baseada em Regras de Negócio**. 2006. 222 f. Tese (Doutorado) - Curso de Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2006. Disponível em: <http://repositorio.unicamp.br/bitstream/REPOSIP/260976/1/Kamada_Aqueo_D.pdf>. Acesso em: 20 mai, 2021.

MEDEIROS. Davi. Operadoras começam a testar 5G no Brasil; veja os locais contemplados. **Olhar Digital**, jul. 2020. Disponível em: <https://olhardigital.com.br/noticia/operadoras-comecam-a-testar-5g-no-brasil-veja-os-locais-contemplados/104079>. Acesso em: out. 2020.

MICROSOFT. **Dados não relacionais e NoSQL**. 2018. Escrito por Zoiner Tejada. Disponível em: <https://docs.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>. Acesso em: 06 dez. 2020.

ORACLE. **O que É um Banco de Dados Relacional?** Disponível em: <https://www.oracle.com/br/database/what-is-a-relational-database/>. Acesso em: 06 dez. 2020.

SBROCCO, J.H.T.D.C.; MACEDO, P.C.D. **Metodologias Ágeis - Engenharia de Software sob Medida**. São Paulo Editora Saraiva, 2012. 9788536519418. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536519418/>. Acesso em: 11 Apr 2021

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução Luiz Claudio Queiroz; revisão técnica Fabio Levy Siqueira – 10ª Ed. – São Paulo – Person Education do Brasil. 2018.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SQLITE. **What Is SQLite?** Disponível em: <https://www.sqlite.org/index.html>. Acesso em: jun, 2021.

TRUJILLO FERRARI, Alonso. Metodologia da pesquisa científica. São Paulo: McGraw-Hill do Brasil, 1982.

VALENTE, Jonas. Mais de 5 bilhões de pessoas usam aparelho celular, revela pesquisa. **Agência Brasil**, Brasília, set. 2019. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2019-09/mais-de-5-bilhoes-de-pessoas-usam-aparelho-celular-revela-pesquisa>. Acesso em: 04 out. 2020.

VIEIRA, Marcelo Milano Falcão Vieira. A comparative study on quality management in the brazilian and the Scottish prison service. 1996. Tese [Doutorado PhD on Business Studies] – Scotland, University of Edinburg, Edimburgo, 1996.

ZANELLA, Liane Carly Hermes. **Metodologia de Pesquisa**. 2. ed. – Florianópolis: Departamento de Ciências da Administração/ UFSC, 2013. E-book. Disponível em: http://arquivos.eadadm.ufsc.br/EaDADM/UAB_2014_2/Modulo_1/Metodologia/material_didatico/Livro%20texto%20Metodologia%20da%20Pesquisa.pdf. Acesso em 10 dez. 2020.

APÊNDICES

APÊNDICE A – IDENTIFICAÇÃO DOS ENTREVISTADOS, PERGUNTAS E RESPOSTAS DAS ENTREVISTAS

IDENTIFICAÇÃO
Nome: Diogenes Vanzela
Profissão: Analista de Integrações
Respostas:
Pergunta nº 1: Sim
Pergunta nº 2: Sim
Pergunta nº 3: Sim
Pergunta nº 4: Sim
Pergunta nº 5: Consistência da informação, uma vez que outro operador pode estar realizando a leitura do mesmo produto. Ao sincronizar, possivelmente a quantidade de itens lidos seria errada.

IDENTIFICAÇÃO
Nome: Douglas Andrei de Lima
Profissão: Application Development Analyst
Respostas:
Pergunta nº 1: Sim
Pergunta nº 2: Sim
Pergunta nº 3: Sim
Pergunta nº 4: Sim, podemos ter vários cenários de uso. Ex: App transportadora, lista de tarefas, etc.
Pergunta nº 5: São N cenários que dá para pensar, um exemplo os APP de transportadora, onde o entregador confirma a entrega realizada e coloca os dados de quem recebeu, se ele estiver em um local com sinal ruim de internet não impactaria o seu trabalho, sendo assim

que a conexão voltar, o conteúdo poderia ser sincronizado de forma automática.

IDENTIFICAÇÃO

Nome: Miguel Freitas Kuckoski

Profissão: Desenvolvedor de software

Respostas:

Pergunta nº 1: Sim

Pergunta nº 2: Sim

Pergunta nº 3: Sim

Pergunta nº 4: Sim

Pergunta nº 5: Garantir que os dados não serão perdidos

IDENTIFICAÇÃO

Nome: Marcelo Walther da Silva

Profissão: Desenvolvedor de software

Respostas:

Pergunta nº 1: Sim

Pergunta nº 2: Sim

Pergunta nº 3: Sim

Pergunta nº 4: Sim

Pergunta nº 5: Custo

IDENTIFICAÇÃO

Nome: Guilherme Steinert de Azevedo

Profissão: Desenvolvedor

Respostas:

Pergunta nº 1: Sim

Pergunta nº 2: Sim

Pergunta nº 3: Sim

Pergunta nº 4: Sim

Pergunta nº 5: reatividade do projeto

--

IDENTIFICAÇÃO Nome: Everton Silva Rosa Profissão: Diretor P&D Hardware e Operacional
Respostas: Pergunta nº 1: Sim Pergunta nº 2: Sim Pergunta nº 3: Sim Pergunta nº 4: Sim Pergunta nº 5: Entrega de cargas/mercadorias. (Sedex, Mercado Livre, etc)

IDENTIFICAÇÃO Nome: Luiz Felipe Pereira Profissão: CTO
Respostas: Pergunta nº 1: Sim Pergunta nº 2: Sim Pergunta nº 3: Sim Pergunta nº 4: Sim Pergunta nº 5: Se o uso do aplicativo ocorrerá em locais de falta de Internet ou se a operação feita pelo aplicativo é crítica e não pode deixar de funcionar mesmo sem rede.

IDENTIFICAÇÃO Nome: Marcus Vinícius Nascimento Oliveira Filho Profissão: Coordenador Operacional
Respostas: Pergunta nº 1: Algumas vezes Pergunta nº 2: Conforme apresentado, sim. Pergunta nº 3: O uso de um banco de dados inserido em aplicativos poderia sim, manter informações seguradas em casos de falhas de conexão com a rede. Talvez até em casos de falhas de alimentação ou bateria dos dispositivos usados.

Pergunta nº 4: Sendo leigo em desenvolvimento mobile digo que o uso de BD deveria ser obrigatório em aplicativos que exigem cadastros de informações ou registros de idéias e documentos. Desde que não interfira no desempenho do aparelho.

Pergunta nº 5: Penso que o melhor uso dessa idéia seria em aplicativos de uso industrial. Principalmente em operações de manipulação de estoque.