

CURSO DE SISTEMAS DE INFORMAÇÃO – TCC	
(X) Pré-projeto () Projeto	Ano/Semestre:
Eixo: Desenvolvimento de Software para Sistemas de Informação	(X) Aplicado () Inovação

PROTÓTIPO DE UM APLICATIVO MÓVEL PARA GESTÃO DE OFICINAS MECÂNICAS

Iago Giuseppe Tambosi

Prof. Francisco Adell Péricas - Orientador

Aumir José Tambosi - Supervisor

1 CONTEXTUALIZAÇÃO

Atualmente, é muito comum encontrar oficinas mecânicas que contam com um software para gerenciamento interno. Muitas vezes, nem é preciso um sistema robusto, pois uma planilha com fórmulas básicas já serve. No entanto, a usabilidade da maioria dos sistemas está limitada a desktops, e existem diversos problemas que um sistema desse tipo não possui a agilidade nem praticidade suficientes para resolver com eficiência.

A empresa Oficina Mecânica Tambosi LTDA, localizada no centro da cidade de Pomerode – SC, trabalha exclusivamente com reparos de automóveis. Para tanto, conta hoje com um sistema de gestão desktop voltado para a área automotiva, permitindo o cadastro de peças, clientes, veículos, criação de ordens de serviço, entre outras funcionalidades (TAMBOSI, 2023). Porém, quando se está lidando com peças grandes e/ou pesadas, o processo de retirar a peça do estoque, levar ao computador para registrar no sistema, e somente então aplicar no veículo, acaba sendo ineficiente. Além disso, algumas peças possuem seu código de barra colados à peça em si, e nesse caso, se o mecânico precisar primeiro aplicar a peça no veículo para averiguar se o modelo da peça é o correto e posteriormente adicionar o serviço no sistema, será necessário anotar o código da peça em algum lugar para depois adicionar no sistema.

A limitação principal de um sistema baseado em computadores desktop é a falta de mobilidade. Não é possível acessar o sistema a qualquer momento nem de qualquer lugar. O mecânico não consegue adicionar uma peça na ordem de serviço sem ter que parar o que está fazendo e ir até o computador (TAMBOSI, 2023). De forma orgânica, isso foi gerando um hábito de anotar o código das peças num papel e, posteriormente, repassar do papel ao sistema. O problema disso é que um papel é facilmente perdido, danificado (principalmente numa oficina de automóveis) e não garante que os dados escritos de fato correspondam a um registro no sistema.

Outro problema é a necessidade de se consultar um serviço prestado quando não se está no escritório ou na oficina. É comum, no momento de realizar a entrega de algum automóvel, o cliente solicitar o custo final do serviço. Numa situação ideal, o mecânico já possuiria a ordem de serviço finalizada e impressa em mãos, conseguindo repassar ao cliente tudo o que foi realizado no automóvel e o custo dos serviços. No entanto, dependendo da ocasião, não é possível entregar a conta fechada junto com o automóvel, e isso pode ocorrer por vários motivos, sejam eles a falta de tempo, falta do registro de alguma peça na conta, esquecimento, entre outros (TAMBOSI, 2023). Mesmo assim, é sempre importante repassar ao cliente o que foi feito no carro, e sem a ordem de serviço impressa, esse relatório fica dependente da memória do mecânico.

Outro problema analisado é o do agendamento de serviços. Atualmente, quando algum cliente entra em contato para agendar um serviço, este é marcado numa caderneta de papel. Isso remete aos mesmos problemas mencionados anteriormente, como a falta de mobilidade: a agenda não pode ser acessada de qualquer lugar, nem a qualquer momento. Além disso, a agenda não notifica o mecânico de algum serviço que está por vir, ficando dependente de uma consulta e da memória dele (TAMBOSI, 2023).

O objetivo principal deste projeto é desenvolver um aplicativo para dispositivo móvel que complemente as funcionalidades do sistema já utilizado pelo estabelecimento, tendo em mente estendê-lo para contemplar novas funções visando a sua agilidade e acessibilidade. Já os objetivos específicos são: providenciar funcionalidades básicas de gestão de uma oficina, como cadastro de clientes, veículos, peças e ordens de serviço; permitir o controle de recursos do estabelecimento de qualquer lugar, exigindo do usuário apenas o acesso à internet; e disponibilizar uma agenda intuitiva e prática, que notifique o usuário dos serviços agendados quando estiver próximo do horário.

2 BASES TEÓRICAS

A seguir, na subseção 2.1, serão abordadas as tecnologias e fundamentos utilizados como base para criação do aplicativo. Já na subseção 2.2, serão mencionados outros trabalhos que contribuíram de alguma forma com este projeto.

2.1 REVISÃO BIBLIOGRÁFICA

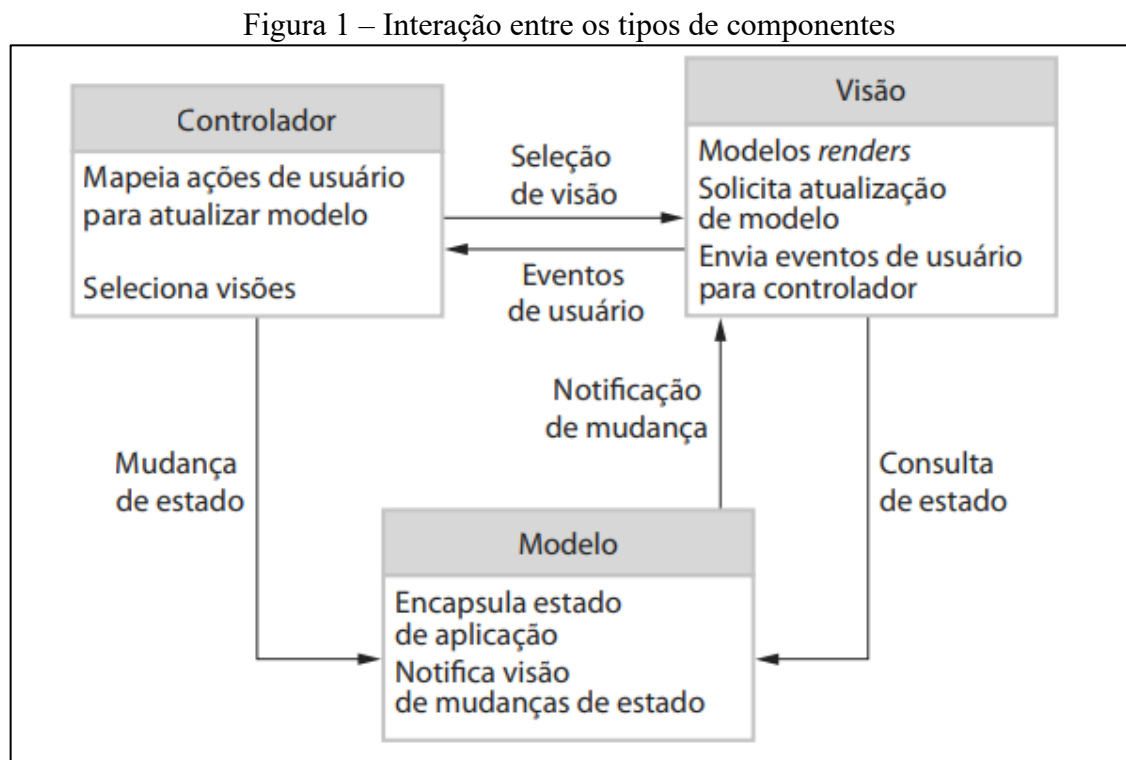
O primeiro tópico a ser explorado é o padrão Model-View-Controller (MVC), que regerá o *design* de construção do aplicativo. O segundo tópico irá abordar as tecnologias escolhidas para a construção do *frontend* e *backend* do projeto. Por fim, o terceiro tópico irá elaborar sobre

a utilização dos serviços da Amazon Web Services (AWS) como plataforma de infraestrutura para hospedagem do *backend*.

2.1.1 Padrão MVC

O MVC é um padrão de projeto, que, de acordo com Gamma e Salgado (2000, p. 20), “Um padrão de projeto nomeia, abstrai e identifica os aspectos-chave de uma estrutura de projeto comum para torná-la útil para a criação de um projeto orientado a objetos reutilizável”. A utilização do padrão MVC irá facilitar a manutenção futura do aplicativo e permitir maior reutilização das classes de interface.

Este padrão irá separar os componentes do aplicativo em três grandes grupos. O primeiro grupo, Controller, contempla as classes que recebem requisições e entradas do usuário e repassam ao próximo grupo, Model. O Model, por sua vez, diz respeito às classes que serão responsáveis pelas regras de negócio e pelo processamento geral das entradas do usuário. Por fim, o grupo View é quem faz a disposição e organização dos dados de saída. A Figura 1 ilustra de forma mais completa como que essas três categorias de componentes se comunicam dentro do padrão estabelecido



Fonte: Sommerville (2011).

2.1.2 Frontend e Backend

Para o desenvolvimento do *frontend*, que será o aplicativo móvel em si instalado no dispositivo do usuário, foi escolhido o *framework* Flutter. Esta tecnologia, criada pela empresa Google, é baseada na linguagem Dart, muito utilizada para o desenvolvimento de aplicativos e

sistemas voltados para dispositivos móveis. O Flutter ajuda a garantir que o aplicativo continue funcionando corretamente ao ser utilizado em outros tipos de dispositivos, como outros modelos de *smartphone* ou sistemas operacionais diferentes, sem comprometer a performance. Além disso, o *framework* conta com uma documentação vasta e didática, indicando tratar-se de uma tecnologia fácil de se obter dicas e suporte.

Para o *backend* (serviço que faz o intermédio entre *frontend* e um banco de dados e opera fora do ambiente do usuário), foi escolhido o *framework* “ASP.NET Core”, desenvolvido pela empresa Microsoft. O “ASP.NET Core é uma estrutura de software livre multiplataforma, para a criação de aplicativos modernos habilitados para nuvem e conectados à Internet” (ROTH, 2023). Além disso, o *framework* permite que o projeto seja compilado para funcionar nos principais sistemas operacionais (Windows, MacOS e Linux) sem perda de compatibilidade (ROTH, 2023). Esta tecnologia será utilizada para a criação de uma API REST (Representational State Transfer - REST), servindo como um intermediário entre o *frontend* e o banco de dados.

Ainda nas entrelinhas do *backend*, é necessário mencionar a escolha do serviço de banco de dados. Para tal, foi escolhido o MySQL, que, de acordo com sua documentação oficial, é um banco de dados relacional, de código aberto, rápido, confiável e escalável (ORACLE, s.d.). A propriedade “escalável” é importante dentro do contexto atual, pois permite que o projeto possa se expandir futuramente, sem ter que realizar grandes mudanças na sua arquitetura.

2.1.3 Hospedagem em nuvem com AWS

Para que o sistema seja acessível de qualquer lugar, bastando que haja conexão com a internet, foi escolhido o uso do Amazon Web Services (AWS). Isso porque a empresa fornece uma plataforma de infraestrutura escalável, confiável e de baixo custo (AMAZON, 2020 apud BRIENZE JÚNIOR, 2022). Além disso, a empresa fornece também alguns planos gratuitos, o que irá ajudar a viabilizar o desenvolvimento da aplicação deste projeto.

O AWS será utilizado para a hospedagem do *backend* e do banco de dados. Neste caso, estes serviços irão funcionar de forma independente entre si e do *frontend*. Isso quer dizer que, se devido a erros ou bugs algum serviço parar de funcionar, os outros componentes não serão comprometidos.

2.2 CORRELATOS

O Quadro 1 ilustra alguns trabalhos que contribuíram de alguma forma para a construção e formulação do projeto atual. Todos foram encontrados utilizando a fonte de pesquisa Google Acadêmico (ou Google Scholar). Os trabalhos correlatos foram escolhidos levando alguns

fatores em consideração, sendo eles a similaridade entre os problemas abordados, as tecnologias utilizadas e as soluções propostas. Dentre eles, vale a pena mencionar alguns trabalhos que mais influenciaram na projeção do sistema sugerido, conforme será destacado e avaliado na seção seguinte.

Quadro 1 - Síntese dos trabalhos correlatos selecionados

Assunto	Filtro	Referência
Agenda de eventos	Aplicativo móvel de agenda	Ruchel (2017)
Sistema para gestão de oficinas	Aplicação desktop oficina mecânica	Silva (2013)
		Moser (2014)
		Menezes (2016)
Implementação de computação em nuvem para <i>backend</i> em Java	Aplicação <i>backend</i> nuvem	Brienze Júnior (2022)

Fonte: elaborado pelo autor.

3 JUSTIFICATIVA

Esse trabalho possui algumas similaridades com os trabalhos correlatos elencados na seção 2.2. O Quadro 2 mostra em especial uma comparação entre dois dos trabalhos correlatos cujos problemas mais se assemelham ao trabalho atual, mas utilizando tecnologias distintas, cada um com suas peculiaridades. É válido mencionar que ambos os trabalhos correlatos providenciam funcionalidades básicas para gestão interna da oficina. Dentre eles, podem ser citados o cadastro de clientes, veículos, peças e ordens de serviço (OS). A aplicação móvel a ser desenvolvida não será muito diferente neste quesito, também contemplando tais funcionalidades, mas com alguns diferenciais que podem aproveitar ao máximo da característica móvel da aplicação.

Quadro 2 – Comparativo entre os principais trabalhos correlatos selecionados

Correlatos Características	Aplicação RIA para gestão de oficinas Silva (2013)	Sistema desktop para gestão de serviços de manutenção automotiva Moser (2014)
Agenda de serviços	Não	Não
Hospedagem dos serviços	Internet	Local
Banco de Dados	MySQL	MySQL
Linguagem de programação	Java	Object Pascal
Aplicação Desktop/Web/Móvel	Web	Desktop
Padrão de arquitetura	Padrão MVC	Não menciona

Fonte: elaborado pelo autor.

Por se tratar de trabalhos relativamente antigos, existem algumas diferenças a serem pontuadas, principalmente sobre o contexto em que os sistemas foram implantados e hospedados. Moser (2014, p. 62) afirma que sua aplicação foi desenvolvida para a plataforma desktop, pois é “voltado para empresas que, em geral, são de pequeno porte e muitas vezes sem acesso adequado, em termos de largura de banda e estabilidade, à Internet”. Esta, no entanto, não é mais exatamente a realidade. De acordo com a Pesquisa Nacional por Amostra de

Domicílios (BRASIL, 2022, n.p), “em 2021, o número de domicílios com acesso à internet no Brasil chegou a 90,0%”. Consequentemente, o acesso à uma conexão de qualidade à internet também ficou mais viável, tanto para a população em geral, quanto para empresas de todos os portes.

Como o contexto da época era diferente, tanto os trabalhos de Moser (2014) quanto o de Silva (2013) foram projetados para serem implantados numa rede ou computadores locais. Por se tratar de uma Aplicação Rica para Internet (RIA), que é uma espécie de aplicação web com funcionalidades e características de softwares convencionais para desktop, o trabalho de Silva (2013), do jeito que foi desenvolvido, poderia ter sido hospedado na internet, tornando-o acessível de qualquer computador com acesso a tal. Isso significa que até smartphones poderiam acessar o sistema, uma vez que esse tipo de dispositivo, na maioria dos casos, também possui um navegador para processamento de páginas web.

O fato acima torna o trabalho de Silva (2013) o mais próximo de uma aplicação móvel, em comparação com os demais trabalhos. Mesmo assim, utilizar tal sistema por meio de um dispositivo móvel não seria ideal nem prático, uma vez que toda interface gráfica foi projetada para se adequar a um computador desktop, e não a uma tela de smartphone. Portanto, é possível afirmar que nenhum dos trabalhos citados atende o fator de mobilidade e praticidade simultaneamente.

Outra questão que nenhum dos trabalhos atende é a agenda de serviços. Isso, talvez, se dê ao fato de que uma agenda em uma plataforma desktop não seria muito diferente do que uma agenda de papel, no que se refere a mobilidade. Inclusive, poderia ser até pior, uma vez que uma agenda de papel pode ser transportada com facilidade, e um computador desktop não.

Então, é neste ponto que entra a aplicação móvel proposta nesse trabalho. Se tratando de um *backend* hospedado em nuvem, qualquer pessoa com o aplicativo instalado, contanto que tenha um usuário e acesso à internet, poderá acessar o sistema. Dentro do sistema, o usuário terá as funcionalidades básicas de gestão da oficina (cadastro de entidades mencionadas anteriormente), e poderá utilizar-se de uma agenda virtual para registrar e consultar eventos ou serviços marcados, recebendo uma notificação no dispositivo quando o evento estiver iminente.

Tendo em vista o projeto acima, pode-se afirmar que o trabalho atual se encaixa no eixo de Desenvolvimento de Software para Sistemas de Informação, uma vez que foi avaliada a necessidade de um sistema de informação, abordado o motivo de outras soluções não atacarem o problema atual e especificado as tecnologias necessárias para seu desenvolvimento. Além disso, este trabalho se torna relevante, de um ponto de vista tecnológico, ao avaliar um cenário que envolve um conjunto de rotinas já estabelecidas, e tentar resolver problemas de ineficiência com a inserção de uma tecnologia móvel nessas rotinas. Se estima também, como uma

contribuição social, uma melhora na qualidade dos serviços prestados tanto para o cliente, quanto para os funcionários.

4 METODOLOGIA

Nesta seção serão apresentados os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF). Além disso, serão especificadas as etapas de análise e de desenvolvimento que irão compor a realização do projeto.

Quanto aos requisitos, a aplicação móvel proposta deve:

- a) permitir o acesso ao sistema, caso as informações de usuário e senha sejam preenchidas corretamente - RF;
- b) permitir o cadastro e controle de entidades básicas (cliente, veículo, peças, fornecedores, funcionários e ordens de serviço) - RF;
- c) permitir a emissão do recibo do serviço (um resumo do trabalho realizado no automóvel) - RF;
- d) disponibilizar uma agenda virtual para criação e controle de eventos e serviços marcados - RF;
- e) por meio da agenda virtual, notificar o usuário sobre eventos ou serviços iminentes - RF;
- f) ter o *backend* construído utilizando a linguagem C#, com o *framework* ASP.NET Core - RNF;
- g) ter o *frontend* construído utilizando a linguagem Dart, com o *framework* Flutter - RNF;
- h) utilizar o serviço de banco de dados MySQL - RNF;
- i) ter o *backend* e banco de dados hospedados em nuvem, por meio do AWS – RNF.

Quanto ao desenvolvimento do projeto, este deve seguir as seguintes etapas:

- a) análise dos processos atuais: analisar e detalhar os processos que serão afetados com o uso da aplicação proposta;
- b) estudo do *framework* Flutter: estudar a linguagem Dart, bem como o *framework* Flutter, utilizando como base de estudo a documentação oficial disponibilizada na internet;
- c) levantamento dos requisitos: juntamente do usuário, especificar e reavaliar os requisitos necessários para a construção da aplicação;
- d) documentação da aplicação: documentar, em um arquivo em formato *markdown* localizado no repositório do projeto, a relação entre os componentes e como fazer seu uso por meio de diagramas de classes, de caso de uso, de persistência de dados;

- e) construção da solução: desenvolver o *backend* e *frontend* da aplicação utilizando, respectivamente, os *frameworks* ASP.NET Core e Flutter, e os Ambientes de Desenvolvimento Integrado (do inglês, Integrated Development Environment - IDE) Visual Studio 2022 e Visual Studio Code; e realizando, em paralelo, testes locais de usabilidade;
- f) implementação: hospedar o *backend* e banco de dados em nuvem, instalar o aplicativo móvel nos dispositivos dos usuários e treiná-los sobre como utilizar;
- g) acompanhamento do uso: avaliar a recepção do aplicativo, por parte dos usuários, verificando e corrigindo eventuais erros, e questionando sobre possíveis melhorias;
- h) avaliação de resultados: validar com os usuários sobre a adaptação das rotinas anteriores com a introdução do aplicativo, avaliando os resultados obtidos e averiguando se os objetivos foram alcançados.

REFERÊNCIAS

- BRASIL, Pesquisa Nacional por Amostra de Domicílios. **90% dos lares brasileiros já tem acesso à internet no Brasil**, aponta pesquisa. Publicado em: 19/09/2022. Disponível em: <https://www.gov.br/casacivil/pt-br/assuntos/noticias/2022/setembro/90-dos-lares-brasileiros-ja-tem-acesso-a-internet-no-brasil-aponta-pesquisa#:~:text=Isto%20.> Acesso em: 20 abr. 2023
- BRIENZE JÚNIOR, Luis Felipe Sabadoto. **Análise da implementação de tecnologias da nuvem Amazon Web Services para aplicação backend em Java**. Universidade Estadual Paulista (Unesp), 2022. Disponível em: <http://hdl.handle.net/11449/217101>. Acesso em: 18 abr. 2023.
- GAMMA, E.; SALGADO, L. A. M. **Padrões de projeto**: soluções reutilizáveis de software orientado a objetos. [S.l.] Grupo A - Bookman, 2000.
- MENEZES, Lucas Martins. **Análise de desenvolvimento de software para uma oficina mecânica**, 2016. Trabalho de conclusão de curso (Curso de Tecnologia em Análise e Desenvolvimento de Sistemas) - Faculdade de Tecnologia de Americana, Americana, 2016.
- MOSER, Adilson José. **Sistema desktop para gerenciamento de serviços de manutenção automotiva**. 2014. 64 f. Trabalho de Conclusão de Curso (Graduação) - Universidade Tecnológica Federal do Paraná, Pato Branco, 2014.
- ORACLE. **MySQL :: MySQL 8.0 Reference Manual :: 1.3.1 What is MySQL?** Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. Acesso em: 18 abr. 2023
- ROTH, D.; ANDERSON, R.; LUTTIN, S. **Visão geral do ASP.NET Core**. Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>. Acesso em: 19 abr. 2023
- RUCHEL, L. V. **Aplicação móvel para automação de uma agenda**: uma forma rápida e prática para acompanhar eventos. 2017. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/15667>. Acesso em: 17 abr. 2023.
- SILVA, C. A. R. DA. **Aplicação RIA para gerenciamento de oficina mecânica**. 2013. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/15583>. Acesso em: 15 abr. 2023.
- SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo, Brasil: Pearson, 2011.

TAMBOSI, Aumir José. Problemas com o atual sistema de gestão da oficina. Entrevista concedida a Iago Giuseppe Tambosi. Pomerode, 23 abr. 2023. Entrevista Pessoal.