

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MARCELO MARCOS VICHAR JUNIOR

**NOVA VERSÃO DO LETSTUR - SISTEMA DE AGÊNCIAS DE PASSEIO
TURÍSTICO**

GUARAPUAVA

2023

MARCELO MARCOS VICHAR JUNIOR

**NOVA VERSÃO DO LETSTUR - SISTEMA DE AGÊNCIAS DE PASSEIO
TURÍSTICO**

New version of LetsTur - Tourism tour agencies system

Projeto de Trabalho de Conclusão de Curso de Graduação apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná, Campus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet.

Orientador: Prof. Dr. Paulo Henrique Soares

GUARAPUAVA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

RESUMO

Este projeto concentra-se na implementação de uma nova versão do sistema (LETSTUR, 2014), uma plataforma utilizada por agências de passeios turísticos na região nordeste do Brasil. Devido ao avanço das tecnologias e conceitos ao decorrer dos anos, atualmente o Letstur poderia ser muito beneficiado. Due to the advancement of technologies and concepts over the years, Letstur could currently benefit greatly from improvements regarding the technologies used and operational efficiency. O com melhorias referentes à tecnológicas utilizadas e eficiência operacional.

O foco principal deste projeto, é desenvolver uma solução que se baseie na modernização tecnológica, automação de processos operacionais e melhoria da experiência do usuário. Isso incluirá a atualização da infraestrutura tecnológica, a simplificação de fluxos de trabalho e a criação de uma interface mais fluida e responsiva.

O projeto abrangerá desde a análise inicial do sistema Letstur (2014) até a implementação prática das melhorias propostas. Será medido o impacto das mudanças implementadas, monitorando a eficiência operacional, e a satisfação dos usuários da nova versão do Letstur (2014).

Palavras-chave: letstur.

ABSTRACT

This project focuses on implementing a new version of the system Letstur (2014), a platform used by tourist tour agencies in the northeast region of Brazil. Due to the advancement of technologies and concepts over the years, Letstur could currently benefit greatly from improvements regarding the technologies used and operational efficiency.

The main focus of this project is to develop a solution that is based on modernization technology, automation of operational processes and improvement of user experience. That will include updating technological infrastructure, simplifying workflows and creating a more fluid and responsive interface.

The project will range from the initial analysis of the Letstur (2014) system to the implementation practice of the proposed improvements. The impact of innovative changes will be measured, monitoring operational efficiency and user satisfaction with the new version of Letstur (2014).

Keywords: letstur.

LISTA DE FIGURAS

Figura 1 – Tela de cadastro de vendas.	11
Figura 2 – Falha na responsividade na tela de cadastro de vendas.	12
Figura 3 – Formulário de pesquisa sobre a atual versão do Lestur	17
Figura 4 – Ilustração de quadro kanban	18
Figura 5 – Fluxo gitflow	19
Figura 6 – Pirâmide de testes	21
Figura 7 – Modelagem do banco de dados	25
Figura 8 – Protótipo da tela de autenticação	26
Figura 9 – Protótipo da dashboard das agências	26
Figura 10 – Protótipo do ponto de venda	27
Figura 11 – Protótipo da tela de cadastro de eventos	27

LISTA DE QUADROS

Quadro 1 – Requisitos funcionais do sistema	23
Quadro 2 – Requisitos não-funcionais do sistema	24

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	<i>Application Programming Interface</i> - Interface de Programação de Aplicação
CSS	<i>Cascading Style Sheets</i> - Folhas de Estilo em Cascatas
HTML	<i>HyperText Markup Language</i> - Linguagem de Marcação de Hipertexto
ORM	<i>Object-Relational Mapping</i> - Mapeamento objeto-relacional
PIB	Produto Interno Bruto
SGBD	Sistema de gerenciamento de banco de dados
SQL	<i>Structured Query Language</i> - Linguagem de consulta estruturada

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Considerações iniciais	8
1.2	Objetivos	9
1.2.1	Objetivo geral	9
1.2.2	Objetivos específicos	9
1.3	Justificativa	9
1.4	Estrutura do trabalho	10
2	LETSTUR	11
3	MATERIAIS E MÉTODOS	13
3.1	Materiais	13
3.1.1	Protótipo	13
3.1.2	API	13
3.1.3	Banco de dados	14
3.1.4	Interface	14
3.1.5	Ferramenta para testes	15
3.1.6	Versionamento	15
3.1.7	Infraestrutura	15
3.1.8	Ferramenta para kanban	16
3.2	Métodos	16
3.2.1	Avaliação da atual versão do sistema	16
3.2.2	Análise e levantamento de requisitos	16
3.2.3	Estruturação do sistema	17
3.2.4	Organização de tarefas	18
3.2.5	Fluxo de versionamento	18
3.2.6	Codificação	19
3.2.7	Testes	20
3.2.8	Documentação técnica	20
4	RESULTADOS PARCIAIS	22
4.1	Avaliação da atual versão do sistema	22
4.2	Análise e levantamento de requisitos	22

4.3	Modelagem do banco de dados	22
4.4	Prototipação visual	23
5	CONSIDERAÇÕES	28
	REFERÊNCIAS	29

1 INTRODUÇÃO

A seção de Introdução é dedicada a fornecer uma visão geral do projeto, explicando o contexto, relevância e objetivos que serão abordados ao longo do trabalho.

1.1 Considerações iniciais

A indústria do turismo desempenha um importante papel na economia brasileira, correspondendo a 8,1% do Produto Interno Bruto (PIB) nacional. O Brasil é conhecido internacionalmente por suas belezas naturais e sua costa. No entanto, apesar do potencial, esse mercado ainda não foi totalmente aproveitado. O país não ocupa uma posição de destaque significativo no cenário turístico global, registrando um fluxo de turistas que representa apenas entre 1% e 2% do total mundial (SEBRAE, 2022).

Um dos principais fatores que impedem que o Brasil se torne uma potência turística, é a falta de profissionalismo nos serviços prestados por determinadas empresas do setor, que muitas vezes oferecem serviços manuais ou com ferramentas ultrapassadas (BBC, 2022). As agências de passeios turísticos, também conhecidas como operadores turísticos, desempenham um papel fundamental na criação de boas experiências para os viajantes, que esperam agilidade e segurança quando fazem aquisição de um determinado serviço da empresa. No entanto, à medida que o setor evolui rapidamente, muitas dessas agências ainda enfrentam desafios significativos relacionados à eficiência operacional, satisfação do cliente e evolução tecnológica.

O sistema Letstur (2014), no mercado desde 2014, foi criado com o intuito de suprir as necessidades dessas agências. A plataforma oferece funcionalidades que agilizam o trabalho de operadores de passeios turísticos. Entre essas funcionalidades, destacam-se a emissão de *vouchers*, a efetuação e histórico de vendas, a gestão de cadastros de passeios, veículos, guias entre outras funcionalidades. O Letstur atua como um agente por trás das operações diárias, auxiliando as agências, com intuito de fornecer serviços de qualidade aos seus clientes, enquanto mantém um registro organizado de suas operações. Atualmente, o sistema está consolidado e conta com diversos clientes com foco na região Nordeste do Brasil, especialmente na cidade de Recife, e tem como planejamento, aumentar sua lista de agências operantes nos próximos anos.

No entanto, apesar da sua estabilidade, e do seu papel importante nas agências de passeios turísticos, o Letstur pode vir a enfrentar desafios significativos em virtude de sua longa trajetória e do rápido avanço tecnológico que ocorreu nos últimos anos. Estes desafios podem ser resumidos em algumas áreas-chave, como a migração para novas tecnologias de maior performance e eficiência e a utilização da plataforma em dispositivos móveis. Diante deste cenário, torna-se evidente a necessidade de uma reestruturação do sistema. É nesse contexto

que, neste trabalho, propõe-se a implementação de uma nova versão do sistema, bem como o desenvolvimento de novas interações e funcionalidades.

Além da modernização tecnológica, também será considerada a experiência dos operadores de vendas, usuários mais assíduos do sistema. A nova versão proposta do Letstur, irá introduzir uma interface mais amigável e responsiva, simplificando o processo de vendas de passeios, e melhorando a experiência do usuário que usa essa funcionalidade frequentemente.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo principal deste projeto é implementar uma versão mais moderna do sistema Letstur, tornando-o uma ferramenta mais eficiente e centrada no usuário, a fim de otimizar os serviços prestados para as agências de passeios turísticos.

1.2.2 Objetivos específicos

- Entrevistar administradores e operadores das agências que utilizam o sistema atualmente, a fim de identificar as principais deficiências e possíveis pontos de melhoria do produto.
- Realizar uma avaliação do sistema atual, identificando seus problemas, inconsistências e pontos de melhoria.
- Analisar e levantar os requisitos funcionais e não-funcionais da nova versão do sistema.
- Projetar o banco de dados e regras de negócio do novo sistema a fim de atender as demandas da aplicação.
- Implementar a nova versão do sistema.
- Documentar a aplicação para facilitar manutenções futuras.
- Realizar testes automatizados no sistema, com objetivo de aumentar a segurança e confiabilidade do código.

1.3 Justificativa

Embora tenha atendido inicialmente às necessidades, é importante notar que a evolução da tecnologia e o surgimento de novas soluções na área de atuação do sistema Letstur destaca-

ram a necessidade de modernização. As melhorias planejadas representam uma oportunidade significativa para aprimorar a eficiência e a qualidade dos serviços prestados.

A nova versão do sistema abordará os desafios previamente identificados, a fim de possibilitar uma infraestrutura mais ágil e de manutenção facilitada. Isso poderá resultar em respostas mais rápidas na resolução de problemas e em uma maior simplicidade na implementação de melhorias, contribuindo para uma experiência mais satisfatória das agências de passeios turísticos. A eliminação da redundância de dados deverá proporcionar informações mais precisas e confiáveis, fortalecendo a credibilidade dos serviços oferecidos.

Uma interface mais fluida e responsiva para dispositivos móveis será uma das melhorias propostas pela nova versão, permitindo um melhor aproveitamento do painel administrativo do sistema.

Portanto, a implementação da nova versão é justificada com base nas melhorias que ela trará para as operações do sistema. Ao abordar esses desafios, o projeto entregará uma solução mais eficiente, que atende às necessidades das agências de forma mais eficaz, enquanto evolui a facilidade de expansão e manutenção.

1.4 Estrutura do trabalho

Ao longo deste trabalho, serão explorados as etapas da modernização, os benefícios esperados e as estratégias para consolidar o Letstur como uma ferramenta indispensável para agências de passeio turísticos. No Capítulo 2, será abordado o funcionamento atual do sistema Letstur, e seus principais desafios encontrados atualmente. Os métodos e materiais que serão utilizados para a construção da solução, estarão descritos no Capítulo 3. Os resultados obtidos na implementação do projeto, estão dispostos no Capítulo 4. No Capítulo 5, estarão as considerações finais deste trabalho.

2 LETSTUR

O Letstur (2014), é uma aplicação web voltada para a administração de agências de passeios turísticos, e atualmente é utilizado por empresas localizadas na região nordeste do Brasil. O sistema possibilita que os usuários armazenem informações de seus passeios oferecidos, além de proporcionar vendas de lugares em eventos e emissões de *vouchers*, possibilitando que as empresas contratantes do produto possuam uma gestão e controle sobre suas atividades. A aplicação está em funcionamento desde 2014, e é um produto da empresa (LETSGROW, 2013), uma fábrica de *softwares web*, localizada em Guarapuava, no estado do Paraná.

Atualmente o sistema enfrenta algumas limitações tecnológicas, que futuramente podem inviabilizar seu crescimento e abrangência na área de operadores de passeio turístico. Um dos principais pontos de possível melhoria da aplicação, é seu painel administrativo. Ao realizar uma venda, uma das funções utilizadas com mais frequência no sistema, diversas informações irrelevantes precisam ser preenchidas (conforme pode ser visto na Figura 1), tornando o processo lento, o que pode vir a afetar a satisfação dos operadores de vendas das agências que consomem o serviço.

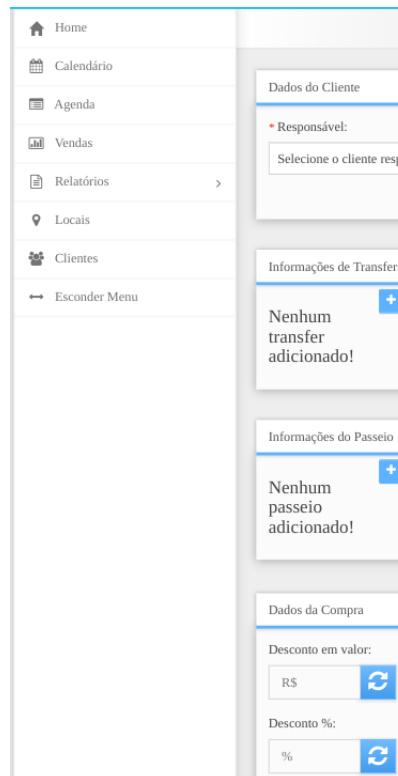
Figura 1 – Tela de cadastro de vendas.

Fonte: Letstur (2014).

Outro aspecto que pode ser otimizado no sistema é a falta de responsividade. Em um contexto no qual a maioria das empresas opta por soluções que sejam funcionais em dispositivos móveis, é importante que a plataforma seja adaptada e otimizada em *smartphones* e *tablets*. As falhas de responsividade no Letstur, torna-o um sistema limitado quando comparado a outras soluções. A Figura 2, retrata como o sistema é renderizado em dispositivos com telas menores, como celulares e tablets.

Devido ao longo período que o sistema está em uso, juntamente ao fato das tecnologias utilizadas originalmente não receberem mais uma manutenção e cuidados satisfatórios quando comparadas a tecnologias mais recentes, o Letstur está exposto a possíveis problemas relacionados ao desempenho de aplicação e busca de dados. Esta limitação pode afetar diversas

Figura 2 – Falha na responsividade na tela de cadastro de vendas.



Fonte: Letstur (2014).

funcionalidades do sistema, comprometendo a eficiência de suas operações. Tarefas que, em um ambiente ideal, seriam conduzidas de maneira ágil e precisa, podem vir a ser afetadas por restrições tecnológicas. Como consequência, é possível estimar uma operação menos eficaz ao decorrer dos próximos anos, o que contribuirá para uma experiência insatisfatória por parte dos operadores de vendas, e administradores de agências.

Diante desses pontos de melhoria, torna-se necessário uma revisão e modernização da aplicação. O atual cenário pode ser otimizado para que o sistema não se torne defasado a médio e longo prazo.

3 MATERIAIS E MÉTODOS

Neste capítulo, serão reportados os materiais e métodos que serão utilizados para alcançar o objetivo do projeto.

3.1 Materiais

A nova versão do Letstur será dividida em duas aplicações que estarão correlacionadas, porém independentes. Será implementada uma *Application Programming Interface* - Interface de Programação de Aplicação (API), e uma aplicação de interface. Nos tópicos a seguir, constam as principais tecnologias que serão utilizadas para o desenvolvimento do sistema, separadas por áreas de programação do projeto.

3.1.1 Protótipo

Para a elaboração do design das interfaces na nova versão do Letstur, optaremos pela utilização do Figma, uma plataforma colaborativa dedicada à construção de layouts e protótipos de design de interfaces (CANALTECH, 2021).

Embora existam diversas ferramentas disponíveis para o design de interfaces, como Sketch, Adobe XD e InVision, a escolha do Figma destaca-se por suas características. O Figma é uma solução baseada em nuvem, permitindo colaboração em tempo real entre vários membros, independentemente da localização geográfica. Essa abordagem colaborativa promove a eficiência e facilita a comunicação. A escolha do Figma no contexto do Letstur é embasada em sua acessibilidade, flexibilidade e na capacidade de proporcionar um ambiente de design dinâmico.

3.1.2 API

Na construção da aplicação que rodará no servidor, será utilizada a linguagem TypeScript¹, acompanhado do Nest, uma ferramenta NodeJs², extensível, versátil e progressiva, desenvolvida para construir aplicações *backend* eficientes, confiáveis e escaláveis (NESTJS, 2021a).

Para estabelecer a comunicação entre o Nest e o banco de dados, será utilizado o Prisma, um *Object-Relational Mapping* - Mapeamento objeto-relacional (ORM) de código aberto para Nodejs e TypeScript, que é utilizado como uma alternativa à escrita de *Structured Query*

¹ <https://www.typescriptlang.org/>

² <https://nodejs.org/en/about>

Language - Linguagem de consulta estruturada (SQL) ou a utilização de outra ferramenta de acesso ao banco de dados, como construtores de consultas SQL.

O Prisma destaca-se devido às suas vantagens distintas em comparação com outros ORMs disponíveis no mercado. Ao examinar algumas das opções concorrentes, é possível identificar alternativas como Sequelize, TypeORM e Bookshelf, cada uma com suas características e nuances. Ao considerar as vantagens específicas do Prisma, podemos destacar sua eficiência na geração de consultas otimizadas e seu suporte nativo para TypeScript. A integração otimizada com o ecossistema Node.js, a facilidade de uso e a robustez na manipulação de operações relacionadas ao banco de dados também contribuem para a preferência pelo Prisma no contexto do Nest (NESTJS, 2021b).

Em resumo, embora existam diversas opções de ORMs disponíveis, a escolha pelo Prisma é respaldada por suas características, oferecendo uma solução eficiente para a camada de acesso ao banco de dados no ambiente Node.js e TypeScript.

3.1.3 Banco de dados

Para a gestão eficiente do armazenamento, o sistema adotará um banco de dados relacional, sendo o PostgreSQL a ferramenta designada para o gerenciamento. O PostgreSQL destaca-se como uma escolha sólida devido à sua robustez, desempenho e suporte extensivo a recursos avançados de SQL.

Embora existam várias opções de Sistema de gerenciamento de banco de dados (SGBD) disponíveis, como MySQL, SQLite e Microsoft SQL Server, a seleção do PostgreSQL é fundamentada em suas características distintivas. Este SGBD de código aberto oferece uma arquitetura sólida, capacidade de expansão horizontal, e uma comunidade ativa que contribui para sua constante evolução e aprimoramento (POSTGRES, 2023).

A escolha do PostgreSQL para a nova versão do Letstur é motivada pela sua capacidade comprovada de lidar com cargas de trabalho complexas, transações robustas e suporte eficaz para aplicações que requerem relacionamentos complexos, o que pode ser de suma importância no decorrer do desenvolvimento do sistema.

3.1.4 Interface

Para desenvolver a aplicação que rodará nos navegadores dos usuários, será utilizada a linguagem TypeScript, juntamente da ferramenta VueJs³. Além disso, será utilizada o framework Nuxt, responsável por criar aplicações VueJs com mais confiança e praticidade (NUXT, 2023). Para implementar as telas e elementos previamente prototipados, será utilizada a ferramenta Tailwind, uma estrutura *Cascading Style Sheets* - Folhas de Estilo em Cascatas (CSS) utilitária

³ <https://vuejs.org/>

composta de classes implementadas para construir design, diretamente na marcação *HyperText Markup Language* - Linguagem de Marcação de Hipertexto (HTML) (TAILWIND, 2023).

3.1.5 Ferramenta para testes

A ferramenta escolhida para implementação de testes automatizados na nova versão do Letstur é o Jest. O Jest destaca-se como solução para testes automatizados em projetos JavaScript, superando alternativas como Mocha e Jasmine, por proporcionar sintaxe intuitiva, simplificando a criação de testes, e proporcionando uma abordagem unificada para testes unitários, de integração e end-to-end.

A agilidade do Jest na execução de testes é outro fator importante pra escolha, já que propõe garantir respostas imediatas aos desenvolvedores e acelerando o ciclo de desenvolvimento. Em comparação com outras soluções, o Jest se destaca pela sua eficiência e pela capacidade de execução de testes em paralelo (JEST, 2023).

3.1.6 Versionamento

Para a gestão do versionamento do sistema, será adotado o Git, com repositórios hospedados na plataforma GitHub. O Git é um sistema de controle de versão distribuído amplamente reconhecido por sua eficiência e flexibilidade, enquanto o GitHub oferece uma plataforma de desenvolvimento colaborativo que hospeda projetos na nuvem.

Apesar da existência de alternativas como Bitbucket e GitLab, a escolha do Git em conjunto com o GitHub é motivada por suas particularidades. O Git, por ser descentralizado, permite um histórico de versionamento eficiente e a facilidade de trabalhar em ambientes distribuídos. O GitHub, além de servir como repositório remoto, proporciona ferramentas adicionais para colaboração, como rastreamento de problemas (issues), integração contínua e revisões de código (GITHUB, 2023).

3.1.7 Infraestrutura

Para simplificar as configurações de ambientes da aplicação, será utilizado o Docker⁴, uma plataforma *open source* que facilita a criação e administração de ambientes isolados, além do Docker Compose, uma ferramenta usada para definir e executar aplicativos de vários contêineres do Docker (DOCKERDOCS, 2023).

Embora existam outras opções para a virtualização de ambientes, como Vagrant e Kubernetes, a escolha do Docker é fundamentada em suas características específicas. O Docker proporciona a criação de contêineres leves e portáteis, garantindo consistência nos ambientes

⁴ <https://www.docker.com/>

de desenvolvimento, teste e produção. O Docker Compose, por sua vez, simplifica a orquestração de múltiplos contêineres, possibilitando uma configuração eficiente e fácil de reproduzir.

3.1.8 Ferramenta para kanban

Na busca por uma solução eficaz de Kanban, foi optado pela plataforma Jira, pois destaca-se como uma escolha estratégica. Ao eleger o Jira para a gestão Kanban, destacam-se vantagens não tão desenvolvidas em outras soluções, como Trello e Asana. O Jira se destaca pela sua robusta integração, permitindo uma visão abrangente do fluxo de trabalho, além de apresentar alta flexibilidade, um rastreamento de tarefas satisfatório para o uso na nova versão do Letstur, e uma interface intuitiva (APPMASTER, 2022).

3.2 Métodos

Nesta seção, serão apresentados as metodologias utilizadas para conduzir o desenvolvimento do sistema proposto. As atividades abrangem desde a identificação do problema até a implantação da solução.

3.2.1 Avaliação da atual versão do sistema

O primeiro objetivo do trabalho será identificar os principais pontos de melhorias do sistema atual. Para isso, será disponibilizado aos operadores das agências de passeio, um formulário com perguntas sobre a atual versão do sistema, a fim de coletar informações sobre possíveis deficiências e pontos de melhoria percebidos no fluxo atual. A Figura 3 ilustra um modelo do formulário que será utilizado para tal tarefa.

Para complementar esta etapa, serão realizados testes funcionais e uma análise documental na atual versão do sistema, buscando um melhor entendimento das melhorias a serem implementadas na nova versão do Letstur.

3.2.2 Análise e levantamento de requisitos

Nesta etapa, será utilizada a metodologia de levantamento de requisitos. Trata-se do processo de compreensão e identificação das necessidades que devem ser solucionadas pelo sistema que será desenvolvido, definindo a função que o software vai desempenhar. Os requisitos são divididos entre funcionais, que tratam das funcionalidades que o sistema deve ter, e não funcionais, que são recursos não considerados funcionalidades, mas características do sistema, como restrições, segurança, confiabilidade, velocidade, validações, entre outros (MARQUES, 2018).

Figura 3 – Formulário de pesquisa sobre a atual versão do Letstur

Letstur - Melhorias futuras

Por favor, preencha os campos e indique pontos de melhorias que facilitem seu dia a dia no uso do sistema Letstur

marcos.marcelo43@gmail.com [Alternar conta](#)
☁

✉ Não compartilhado

* Indica uma pergunta obrigatória

Qual sua função dentro da agência de passeios? *

Administrador

Operador

Quais operações você mais realiza dentro do sistema atualmente? *

Venda de eventos

Cadastro de eventos

Cadastro de passeios

Cadastros de motoristas, veículos, etc

Emissão de voucher

Deixe sugestões do que pode melhorar no sistema, nas atividades que você realiza no dia a dia

Sua resposta

Enviar
Limpar formulário

Fonte: Autoria própria (2023).

3.2.3 Estruturação do sistema

Após a conclusão do levantamento de requisitos do sistema, procede-se à etapa seguinte do ciclo de desenvolvimento, que consiste na modelagem do banco de dados da aplicação. Essa fase é crucial para a estruturação eficiente e organização dos dados que serão manipulados pelo sistema. A definição física de tabelas, relacionamentos e atributos proporciona uma base para o armazenamento e recuperação de informações, a fim de otimizar a eficácia operacional e garantir a integridade dos dados ao longo do ciclo de vida do software.

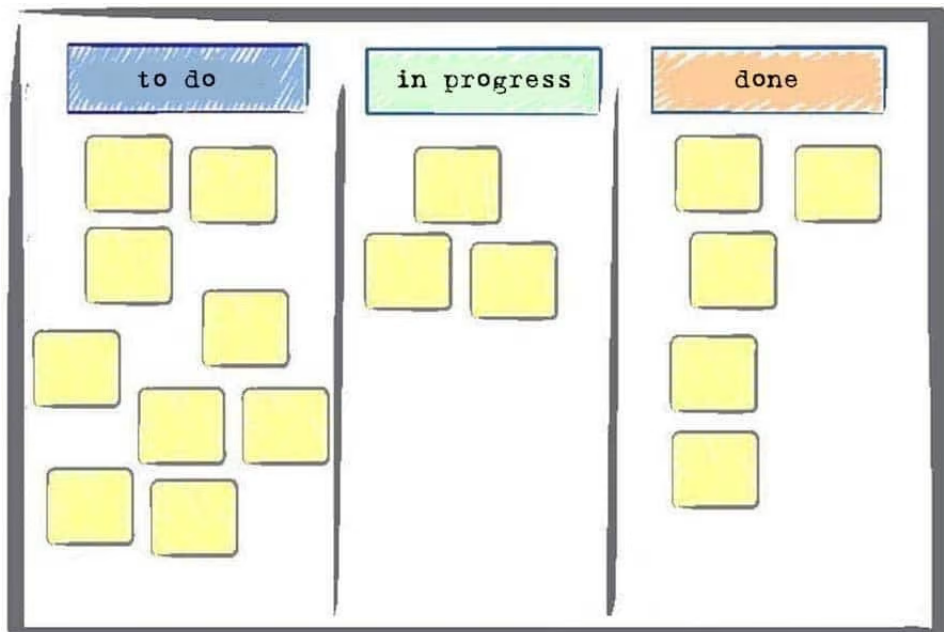
Simultaneamente à modelagem do banco de dados, será realizada a prototipação das interfaces gráficas do sistema. Essa etapa visa fornecer uma representação visual do *layout* e fluxo de interação do usuário com o software. A prototipação das telas permite antecipar e refinar a experiência do usuário antes mesmo da implementação completa do sistema, proporcionando a oportunidade de ajustes e melhorias com base no *feedback* inicial. Essa abordagem acelera o desenvolvimento, e contribui para a identificação precoce de possíveis desafios de usabilidade.

3.2.4 Organização de tarefas

Para controlar a prioridade e andamentos das tarefas nas etapas de desenvolvimento do projeto, será utilizado um quadro *Kanban*. Um quadro *Kanban* é uma ferramenta de gerenciamento de projeto ágil que auxilia na visualização de tarefas, limitação de trabalho em andamento e maximização de eficiência (ou fluxo). Ele pode ajudar equipes a estabelecer ordem no processo diário. Os quadros *Kanban* usam cartões, colunas e melhorias contínuas para ajudar as equipes de tecnologia e serviço a se comprometerem com a quantidade certa de trabalho e finalizarem suas tarefas no tempo adequado. Um quadro *Kanban* na sua forma mais básica, possui 3 colunas: *To do*, para tarefas que devem ser realizadas, *In progress*, onde estão localizadas as tarefas que estão em andamento, e *Done*, para exibir tarefas que já foram finalizadas (MIRO, 2023).

A Figura 4, ilustra um quadro *kanban* básico, com as colunas *to do*, *in progress* e *done*. Durante o desenvolvimento do sistema, um quadro muito semelhante ao da figura será utilizado, inicialmente sem necessidade de acréscimo de colunas adicionais para organização.

Figura 4 – Ilustração de quadro kanban



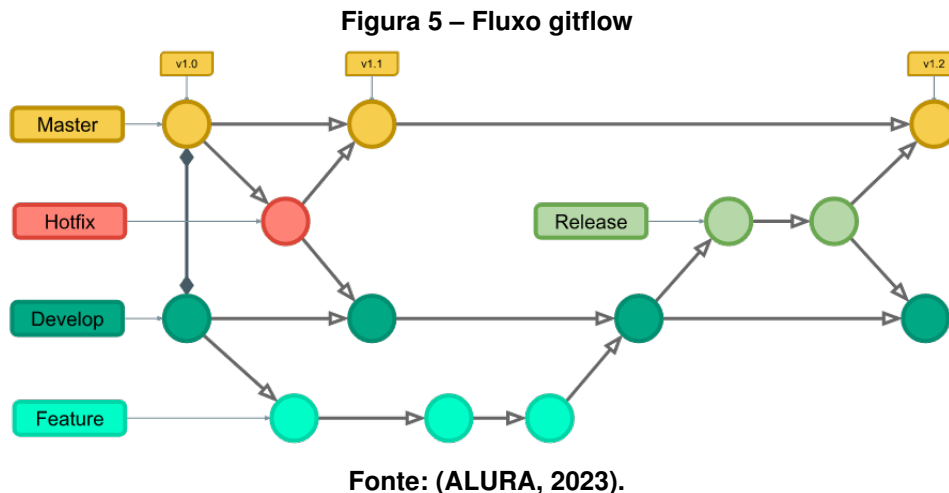
Fonte: (T2INFORMATIK, 2023).

3.2.5 Fluxo de versionamento

Para gerenciar as versões do sistema, e garantir que erros sejam corrigidos sem prejudicar outras atividades, será utilizado o *Gitflow*. O *Gitflow* é um modelo de fluxo de trabalho do *Git*, que define um padrão de ramificação com base nos lançamentos das versões do projeto.

O *gitflow* atribui às *branches* principais do sistema, funções durante o lançamento do projeto (ATLASSIAN, 2023).

A Figura 5, ilustra um fluxo básico que o *gitflow* utiliza para a organização do projeto. Durante o desenvolvimento da nova versão do Letstur, esse mesmo esquema de branches apresentado na figura será utilizado, com as tarefas sendo realizadas a partir da *branch develop*, e atribuindo a *branch master*, a responsabilidade de ser o ramo mais estável durante o desenvolvimento e lançamento.



3.2.6 Codificação

O processo de codificação do sistema se divide em duas etapas: a codificação do *backend* e do *frontend*. Essa segmentação permite uma abordagem mais específica e eficiente na implementação do software, garantindo uma divisão clara de responsabilidades entre o lado do servidor e a interface do usuário.

Na etapa de codificação do *backend*, será realizada a implementação das funcionalidades essenciais que operam no servidor. Serão aplicados princípios consolidados de boas práticas de programação, como os princípios SOLID, a fim de criar uma API robusta e flexível. O SOLID, conjunto de princípios da programação orientada a objetos, orienta o desenvolvedor na criação de código mais coeso, escalável e de fácil manutenção. Cada letra do acrônimo SOLID representa um princípio específico (Responsabilidade Única, Aberto/Fechado, Substituição de Liskov, Segregação de Interface e Inversão de Dependência), os quais são aplicados de maneira a assegurar a modularidade e a extensibilidade do código, facilitando a manutenção e evolução do sistema ao longo do tempo (ARAGÃO, 2017).

Paralelamente, na codificação do *front-end*, será implementada a interface do usuário, a fim de garantir uma experiência fluida e intuitiva para o usuário final. A aplicação de conceitos modernos de desenvolvimento front-end, como frameworks JavaScript, facilita a criação de interfaces dinâmicas e responsivas. Ademais, a utilização de práticas de codificação consistentes,

como a adoção de padrões de projeto e a modularização do código, contribui para a legibilidade e a escalabilidade do sistema.

A comunicação entre o backend e o frontend é essencial para o funcionamento do sistema, e a adoção de boas práticas de programação em ambas as camadas do desenvolvimento é fundamental para a coesão global do projeto. A abordagem orientada a objetos, respaldada pelos princípios SOLID, não apenas promove a manutenibilidade e a extensibilidade do código, mas também facilita a colaboração entre desenvolvedores e a compreensão do sistema como um todo.

3.2.7 Testes

Ambas as aplicações, terão testes automatizados, a fim de manter as funcionalidades do sistema confiáveis e de fácil manutenção futura. Para implementar testes satisfatórios e controlar o custo operacional, o sistema deverá contar com 3 tipos de testes:

- **Testes de unidade:** são os testes realizados na menor parte testável de uma aplicação, independentemente da sua interação com outras partes do código.
- **Testes de integração:** Têm como objetivo testar um conjunto de unidades interagindo entre si. Alguns casos comuns de cobertura de testes de integração são testes realizados na comunicação com o banco de dados, comunicação de interfaces, micro-serviços, etc.
- **Testes de Ponta a Ponta (E2E):** Têm como objetivo principal testar o comportamento do usuário na aplicação, entre outros casos. São testes que simulam o ambiente real, e tentam garantir que operações completas estejam com o comportamento esperado.

A relação entre esses tipos de testes, pode ser representada graficamente através de uma pirâmide de testes. Quanto mais próximo do topo da pirâmide, maior será a cobertura do teste, e conseqüentemente, maior o custo operacional (TRAN, 2022).

A Figura 6, ilustra uma pirâmide de testes, exibindo os diferentes níveis de tipos de testes. Na nova versão do letstur, os testes serão implementados seguindo essa metodologia, tanto em sua API, quanto em sua interface.

3.2.8 Documentação técnica

Dado que o Letstur se configura como um sistema empresarial, com perspectivas de ser mantido e evoluído por uma equipe de desenvolvedores no futuro, destaca-se como prática essencial no desenvolvimento da sua nova versão a elaboração de uma documentação técnica abrangente.

Figura 6 – Pirâmide de testes

Fonte: Autoria própria (2023).

A escrita de documentação técnica desempenha um papel crucial no processo de desenvolvimento e manutenção do software, fornecendo uma fonte centralizada e compreensível de informações. Ao documentar detalhadamente as funcionalidades e as regras de negócio da aplicação, cria-se um recurso valioso que possibilita a disseminação consistente do conhecimento entre os membros da equipe de desenvolvimento, tanto atuais quanto futuros.

A documentação técnica atua como um guia integral, oferecendo uma visão panorâmica do sistema e facilitando a compreensão das camadas funcionais e estruturais da aplicação. Isso não apenas acelera o processo de integração de novos desenvolvedores à equipe, mas também assegura uma boa transição entre diferentes fases do ciclo de vida do software, como manutenções corretivas, evolutivas ou adaptativas.

Além disso, a documentação técnica desempenha um papel vital na estabilidade e na atualização contínua do sistema. Ao manter a documentação alinhada com as mudanças implementadas, garante-se que os desenvolvedores tenham acesso a informações precisas e atualizadas sobre as funcionalidades e o comportamento do sistema. Isso é particularmente crucial em um contexto empresarial, onde a estabilidade e a consistência das operações são imperativas para o sucesso contínuo das atividades comerciais (GUIADEV, 2023).

4 RESULTADOS PARCIAIS

Este capítulo apresenta o que foi obtido como resultado do trabalho, enquanto este projeto foi escrito.

4.1 Avaliação da atual versão do sistema

Após a análise do fluxo da plataforma, e das respostas obtidas do formulário fornecido aos usuários da atual versão do sistema, foi possível constatar que a área de vendas e emissão de *vouchers*, são as funcionalidades mais acessadas no sistemas, e que a redundância de dados no registro de novas vendas, é o principal ponto de melhoria que os usuários desejam. Além disso, foi possível coletar sugestões sobre o desempenho do sistema, e constatado que o tempo de resposta das operações pode ser otimizado, acelerando os processos atuais.

4.2 Análise e levantamento de requisitos

Após a avaliação da versão atual do sistema, tornou-se necessário o levantamento dos requisitos que guiarão o desenvolvimento da próxima versão do Letstur. Esse processo de análise é crucial para assegurar que a nova iteração do sistema seja concebida de maneira estratégica, alinhada aos objetivos específicos de satisfação dos requisitos identificados.

O Quadro 1, apresentado a seguir, detalha de forma abrangente os requisitos funcionais da aplicação. Os requisitos elencados, constituem o núcleo do planejamento de desenvolvimento, fornecendo uma base sólida para a concepção de funcionalidades que atendam às expectativas e demandas dos usuários finais.

Já o Quadro 2 se destina a catalogar e especificar os requisitos não-funcionais da aplicação, ampliando a abrangência da análise para além das funcionalidades. Esses requisitos, muitas vezes imperceptíveis ao usuário final, desempenham um papel crucial na definição de parâmetros como desempenho, segurança, escalabilidade e usabilidade. Ao incorporar esses elementos não-funcionais, busca-se garantir não apenas a eficácia operacional do sistema, mas também sua funcionalidade em aspectos que transcendem a execução de tarefas, a fim de garantir a qualidade do sistema.

4.3 Modelagem do banco de dados

A modelagem do banco de dados é uma fase crucial no desenvolvimento da nova versão do sistema, desempenhando um papel fundamental na organização e estruturação dos dados que serão armazenados e gerenciados pela aplicação. Após o levantamento e análise de dados, foi implementado uma modelagem física de dados, que posteriormente será traduzida em um

Quadro 1 – Requisitos funcionais do sistema

Requisito	Usuário	Descrição
RF001	Todos	Autenticação com email e senha
RF002	Todos	Edição de perfil e senha
RF003	Administrador geral	Cadastro de usuários
RF004	Administrador geral	Cadastro de agências
RF005	Agência	Agenda de passeios
RF006	Administrador de agência	Edição de dados da agência
RF007	Administrador de agência	Cadastro de operadores
RF008	Administrador de agência	Cadastro de eventos
RF009	Administrador de agência	Cadastro de passeios
RF010	Administrador de agência	Cadastro de motoristas
RF011	Administrador de agência	Cadastro de guias
RF012	Administrador de agência	Cadastro de veículos
RF013	Administrador de agência	Relatório de vendas
RF014	Operador de vendas	Ponto de vendas de eventos
RF015	Operador de vendas	Emissão de voucher
RF016	N/A	Envio de voucher por email

Fonte: Autoria própria (2023).

banco de dados, cujo a API irá gerenciar através de regras de negócios. A modelagem física de dados do sistema proposta inicialmente, está ilustrada na Figura 7.

4.4 Prototipação visual

A prototipação visual desempenha um papel essencial no processo de design da interface da nova versão do Letstur, proporcionando uma representação das ideias concebidas durante a fase inicial do desenvolvimento do software. Trata-se da criação de representações visuais que simulam a aparência do produto final, permitindo que sejam validados e experimentados os conceitos de design antes da implementação efetiva. A Figura 8, é uma representação de como ficará a área de autenticação da aplicação, primeira tela que será acessada pelos usuários do sistema.

A tela de *dashboard* do Letstur, exibirá aos administradores das agências, informações pertinentes sobre vendas efetuadas e eventos próximos. A Figura 9 retrata o *layout* que essa área do sistema possuirá.

Quadro 2 – Requisitos não-funcionais do sistema

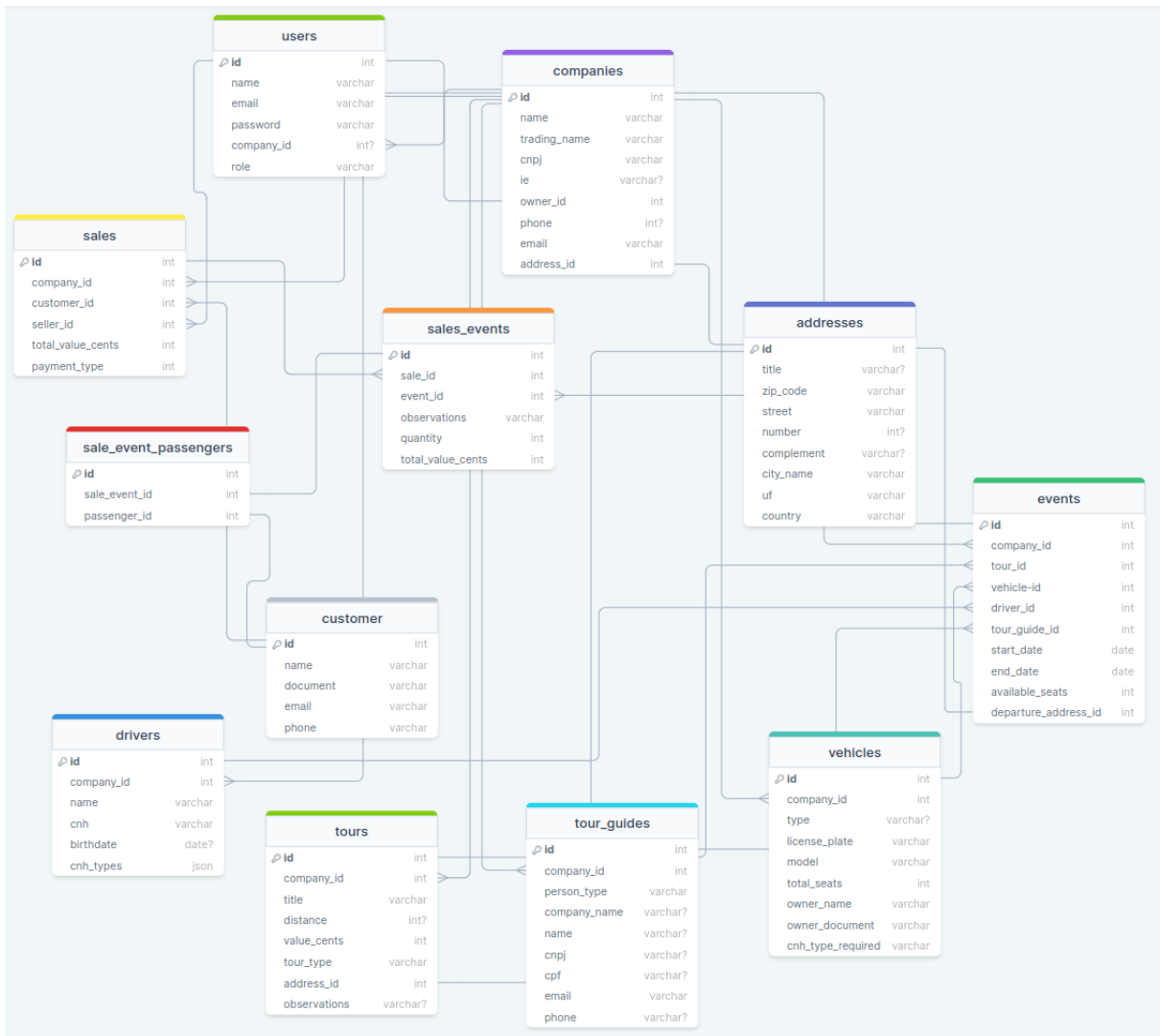
Requisito	Descrição
RNF001	A interface do usuário deve ser intuitiva
RNF002	A infraestrutura do sistema deve ser constituída com contêiner do docker
RNF003	A API deve ser constituída utilizando Node e Typescript
RNF004	A interface deve ser constituída utilizando VueJs
RNF005	A interface do usuário deve ser responsiva
RNF006	A aplicação deve suportar pelo menos 1000 acessos simultâneos
RNF007	O tempo de resposta das requisições não devem ultrapassar 3 segundos
RNF008	Nenhuma agência pode visualizar informação de outras
RNF009	O sistema deve ser compatível com os principais navegadores.
RNF010	Deve ser possível acessar o sistema em dispositivos móveis.
RNF011	O sistema deve ser capaz de ser implantado em ambientes de nuvem.

Fonte: Autoria própria (2023).

A área de vendas de lugares em eventos da nova versão de sistema, deverá ter menos informações, a fim de realizar uma venda de forma mais rápida. O protótipo dessa tela, propõe um formulário de vendas dividido em passos e sem informações não essenciais para a realização da venda, conforme a Figura 10.

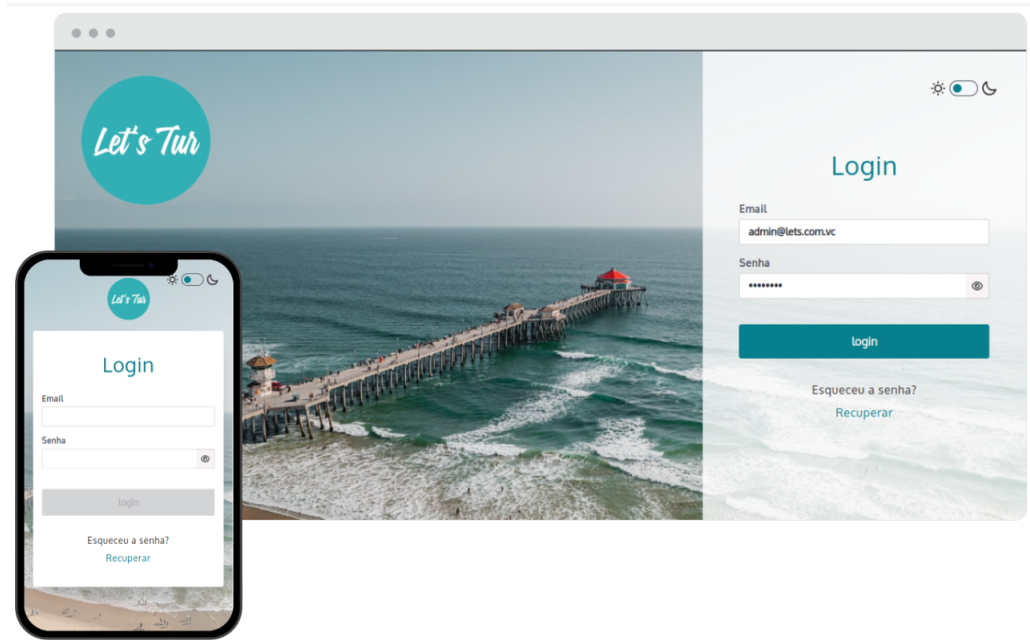
A nova versão do Letstur contará com algumas telas de cadastros, como de usuários, eventos, agências, etc. Essas áreas do sistema serão de suma importância para o controle das agências que utilizarão o sistema. A Figura 11 ilustra a tela de cadastro de eventos do sistema, com as funções de listagem de eventos, além de cadastro, edição e exclusão de um evento específico.

Figura 7 – Modelagem do banco de dados



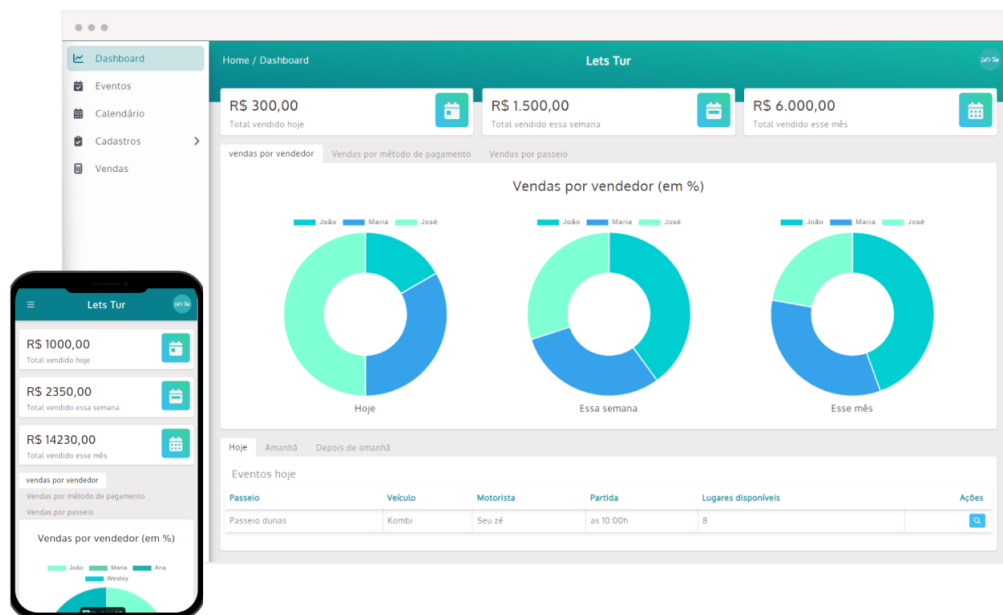
Fonte: Autoria própria (2023).

Figura 8 – Protótipo da tela de autenticação



Fonte: Autoria própria (2023).

Figura 9 – Protótipo da dashboard das agências



Fonte: Autoria própria (2023).

Figura 10 – Protótipo do ponto de venda

Fonte: Autoria própria (2023).

Figura 11 – Protótipo da tela de cadastro de eventos

Id	Passeio	Veículo	Lugares disponíveis	Partida	Chegada	Ações
1	Passeio dunas	ABC-1234 - Fiat Uno	3	10/09/2021 10:00	10/09/2021 12:00	[ícone]
2	Passeio praia	ABC-3434 - Mini Van	10	10/09/2021 10:00	10/09/2021 12:00	[ícone]
3	Passeio lagoa	ABC-3434 - Mini Van	10	12/09/2021 10:00	12/09/2021 12:00	[ícone]
4	Passeio praça	ABC-1111 - Ônibus	44	13/09/2021 10:00	14/09/2021 10:00	[ícone]

Na parte inferior esquerda, há uma visualização em smartphone da mesma tela de eventos, mostrando o formulário de 'Novo evento' com campos para ID, Passeio, Veículo, Lugares disponíveis, Partida, Chegada e Ações.

Fonte: Autoria própria (2023).

5 CONSIDERAÇÕES

A modernização do sistema Letstur, pode trazer uma série de vantagens, incluindo uma maior agilidade na execução dos processos pelos usuários, e conseqüentemente um melhor desempenho e avaliação do *software*. É importante reconhecer que, embora essa modernização possa trazer vantagens substanciais, também é fundamental considerar as possíveis desvantagens. Um dos principais pontos a serem abordados, é a dificuldade de adaptação de maneira imediata dos usuários, considerando que alguns fluxos do painel administrativo deverão ser modificados, tanto para melhoria de performance, quanto para facilitar a usabilidade.

Portanto, após análise das possíveis vantagens e desvantagens que a nova versão do Letstur fornecerá, conclui-se que a implementação do projeto proposto é necessária, e tem potencial para proporcionar significativas melhorias em relação ao sistema atual.

REFERÊNCIAS

- ALURA. **Git Flow: entenda o que é, como e quando utilizar**. 2023. Disponível em: <https://www.alura.com.br/artigos/git-flow-o-que-e-como-quando-utilizar>. Acesso em: 05 Nov. 2023.
- APPMASER. **O que é JIRA? Visão geral e guia completo**. 2022. Disponível em: <https://appmaster.io/pt/blog/o-que-e-jira>. Acesso em: 11 Dez. 2023.
- ARAGÃO, T. **SOLID — Princípios da Programação Orientada a Objetos**. 2017. Disponível em: <https://medium.com/thiago-aragao/solid-princ%C3%ADpios-da-programa%C3%A7%C3%A3o-orientada-a-objetos-ba7e31d8fb25#:~:text=SOLID%20s%C3%A3o%20princ%C3%ADpios%20ou%20boas,a%20qualquer%20linguagem%20de%20programa%C3%A7%C3%A3o>. Acesso em: 06 Out. 2023.
- ATLASSIAN. **Saiba tudo sobre o Gitflow Workflow**. 2023. Disponível em: <https://www.atlassian.com/br/git/tutorials/comparing-workflows/gitflow-workflow#:~:text=O%20que%20%C3%A9%20Gitflow%3F,por%20Vincent%20Driessen%20no%20nvie>. Acesso em: 05 Nov. 2023.
- BBC. **4 fatores que impedem que Brasil vire potência no turismo apesar do potencial**. 2022. Disponível em: <https://www.bbc.com/portuguese/brasil-63671736>. Acesso em: 08 set. 2023.
- CANALTECH. **O que é Figma**. 2021. Disponível em: <https://canaltech.com.br/software/o-que-e-figma/>. Acesso em: 05 Out. 2023.
- DOCKERDOCS. **Docker Compose overview**. 2023. Disponível em: <https://docs.docker.com/compose/>. Acesso em: 05 Out. 2023.
- GITHUB. **O que é github**. 2023. Disponível em: <https://ebaonline.com.br/blog/o-que-e-github>. Acesso em: 05 Out. 2023.
- GUIADEV. **Documentação Técnica**. 2023. Disponível em: <https://guia.dev/pt/pillars/software-architecture/technical-documentation.html>. Acesso em: 11 Dez. 2023.
- JEST. **Documentação Jest**. 2023. Disponível em: <https://jestjs.io/pt-BR/>. Acesso em: 11 Dez. 2023.
- LETSGROW. **Fábrica de softwares web**. 2013. Disponível em: <https://lets.com.vc/>. Acesso em: 16 sep. 2023.
- LETSTUR. **Sistema de gerenciamento de agências turísticas**. 2014. Disponível em: <http://www.lets.tur.br/>. Acesso em: 07 sep. 2023.
- MARQUES, R. **Como realizar o Levantamento de Requisitos no desenvolvimento de software**. 2018. Disponível em: <https://www.cedrotech.com/blog/levantamento-de-requisitos-e-desenvolvimento-de-softwares/#:~:text=De%20forma%20simples%2C%20o%20levantamento,que%20o%20software%20vai%20desempenhar>. Acesso em: 06 Out. 2023.
- MIRO. **O que é Kanban**. 2023. Disponível em: <https://miro.com/pt/agile/o-que-e-quadro-kanban/>. Acesso em: 05 Nov. 2023.

NESTJS. **Documentação**. 2021. Disponível em: <https://nestjs.com/>. Acesso em: 05 Out. 2023.

NESTJS. **Prisma**. 2021. Disponível em: <https://docs.nestjs.com/recipes/prisma>. Acesso em: 05 Out. 2023.

NUXT. **The intuitive web framework**. 2023. Disponível em: <https://nuxt.com/>. Acesso em: 05 Out. 2023.

POSTGRES. **O que é postgres**. 2023. Disponível em: <https://www.postgresql.org/>. Acesso em: 05 Out. 2023.

SEBRAE. **A importância dos atrativos turísticos do Brasil**. 2022. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/a-importancia-dos-atrativos-turisticos-do-brasil,4db2a30bd0f13810VgnVCM100000d701210aRCRD#:~:text=O%20turismo%20no%20Brasil%20corresponde,turistas%20venham%20visitar%20o%20pa%C3%ADs.&text=O%20turismo%20%C3%A9%20uma%20das,no%20Brasil%20n%C3%A3o%20%C3%A9%20diferente>. Acesso em: 07 set. 2023.

T2INFORMATIK. **What is a Taskboard?** 2023. Disponível em: <https://t2informatik.de/en/smartpedia/taskboard/>. Acesso em: 05 Nov. 2023.

TAILWIND. **Rapidly build modern websites without ever leaving your HTML**. 2023. Disponível em: <https://tailwindcss.com/>. Acesso em: 05 Out. 2023.

TRAN, M. **Unit, Integration, and End-to-End Testing: What's the Difference?** 2022. Disponível em: <https://www.twilio.com/blog/unit-integration-end-to-end-testing-difference>. Acesso em: 15 Nov. 2023.