
Tour Experience

Tour Experience
Arquitetura de Software
Versão <1.6>

Histórico de Revisões

Data	Versão	Descrição	Autor
13/06/2023	1.0	Criação do documento	João Victor, Matheus, Pedro, Talles Eduardo.
28/06/2023	1.1	Elaboração dos requisitos e partes interessadas	João Victor, Matheus, Pedro, Talles Eduardo.
02/07/2023	1.2	Implementação de diagramas e recursos visuais	João Victor, Matheus, Pedro, Talles Eduardo.
15/07/2023	1.3	Elaboração dos pontos de vista das partes envolvidas	João Victor, Matheus, Pedro, Talles Eduardo.
24/07/2023	1.4	Revisão do documento	João Victor, Matheus, Pedro, Talles Eduardo.
02/08/2023	1.5	Detalhamento das decisões arquiteturais	João Victor, Matheus, Pedro, Talles Eduardo.
04/08/2023	1.6	Revisão do documento, versão final.	João Victor, Matheus, Pedro, Talles Eduardo.

Sumário

1. Introdução.....	4
1.1. Descrição.....	4
1.2. Finalidade.....	4
1.3. Arquitetura (Resumo).....	4
1.4. Pessoas-chave.....	5
1.5. Escopo.....	5
1.6. Definições, acrônimos e abreviações.....	5
1.7. Visão Geral.....	5
2. Requisitos.....	5
2.1. Requisitos Funcionais.....	5
2.2. Requisitos Não Funcionais.....	6
3. Contexto da Arquitetura.....	6
3.1. Funcionalidades e restrições Arquiteturais.....	6
3.2. Atributos de Qualidades Prioritários.....	7
4. Representação da Arquitetura.....	8
5. Ponto de vista dos Casos de Uso.....	8
5.1. Descrição.....	8
5.2. Visão de Casos de Uso.....	9
6. Decisões arquiteturais.....	10
6.1. Requisitos arquiteturais.....	10
6.2. Alternativas consideradas.....	10
6.3. Decisões tomadas.....	13
6.4. Impacto nas partes interessadas.....	14
7. Ponto de Vista do Projetista.....	16
7.1. Visão Geral.....	16
7.2. Visão de Componentes.....	16
7.3. Detalhamento das Camadas.....	17
8. Ponto de Vista do Desenvolvedor.....	17
8.1. Visão Geral.....	17
8.2. Visão Lógica.....	18
8.3. Visão de Segurança.....	18
8.3.1. Detalhamento da Segurança.....	18
9. Ponto de Vista do Implantador.....	19
9.1. Visão Geral.....	20
9.2. Visão Física.....	20
9.2.1. Detalhamento dos nós.....	20

Lista de figuras

Figura 1 - Representação da arquitetura.....	8
Figura 2 - Diagrama de Casos de uso 1.....	9
Figura 3 - Diagrama de Casos de uso 2.....	9
Figura 4 - Arquitetura de Implantação.....	19
Figura 5 - Diagrama de Banco de Dados.....	19

1. Introdução

1.1. Descrição

A Tour Experience é uma aplicação WEB que busca solucionar problemas recorrentes na venda de produtos de grande valor agregado, além de trazer alguns diferenciais para seu negócio no que tange os anúncios e marketing. Se o seu cliente não possui disponibilidade de horário, se ele reside em outra localidade ou se questões climáticas - como chuva ou até mesmo o sol escaldante - e a logística te atrapalham na hora de apresentar o seu produto de grande valor agregado como imóveis, apartamentos, casas ou até mesmo veículos, aqui está sua solução. A Tour Experience busca através da tecnologia e da inovação facilitar e melhorar exponencialmente a forma como você apresenta seu catálogo ou produto. A nossa proposta é ser uma plataforma de Tours Virtuais 360°, na qual você pode personalizar como quiser, inserir áudios, vídeos e todas as informações que achar relevante em seu anúncio diretamente nas imagens de alta qualidade. Além disso, a nossa plataforma busca se diferenciar das outras já existentes trazendo exclusividade e uma interface totalmente diferente, tendo a usabilidade e facilidade de uso como um dos pontos principais. A plataforma também possui suporte ao uso de óculos de realidade virtual e possui um catálogo de diferentes serviços, como a criação do tour e até mesmo tirar as fotos 360°, caso você precise. Então se seu negócio precisa de maior conversão e seus produtos necessitam de coisas além de imagens estáticas e vídeos sem vida e interação, se eles precisam de maior riqueza de detalhes, imersão, disponibilidade (seja onde for) e se seu cliente precisa de maiores informações, opções, qualidade de anúncio e uma experiência fantástica e diferenciada, a Tour Experience pode te ajudar.

1.2. Finalidade

O objetivo deste documento é apresentar a arquitetura de software para uma aplicação de tour virtual, a TourExperience, servindo de base para projetistas, desenvolvedores e qualquer outro interessado na arquitetura deste sistema. A aplicação permitirá a exploração de ambientes específicos através de imagens 360° e personalização dos tours.

1.3. Arquitetura (Resumo)

A aplicação será Web. Teremos o Front-end baseado em React e Three.js para renderização das imagens em 360 e para aplicação das diferentes funcionalidades visuais. O Back-end será baseado em Node, utilizando Next.js, funcionando basicamente como um MVC. Para armazenamento dos dados e imagens, por se tratar de um grande quantidade de dados seria utilizado um servidor próprio de armazenamento.

1.4. Pessoas-chave

As pessoas-chave envolvidas na aplicação são o fornecedor, no caso as empresas donas de imóveis - corretoras, imobiliárias - e/ou empresas do ramo automobilístico como concessionárias e revendas ou pessoas físicas que pretendem vender imóveis e/ou veículos. Também pode-se destacar como pessoa-chave, aquela que deseja fazer a compra de algum dos produtos citados acima, que neste caso seria o cliente.

1.5. Escopo

O sistema permitirá que os usuários realizem visitas virtuais a imóveis, apartamentos, casas, locais específicos e até mesmo explorem veículos através de imagens 360°, servindo assim de repositório e mostruário de bens de grande valor agregado. Além disso, será possível a utilização de óculos de realidade virtual e personalização dos anúncios.

1.6. Definições, acrônimos e abreviações

Tour Virtual - experiência interativa que permite que os usuários explorem ambientes virtuais através de imagens 360° e modelos 3D.

Ponto de Interesse - um local específico dentro do ambiente virtual que possui informações adicionais associadas.

Recursos de mídia - elementos como imagens, vídeos ou áudios que podem ser incorporados aos pontos de interesse.

1.7. Visão Geral

A aplicação consistirá em um aplicativo web que permitirá em um primeiro momento os usuários visualizarem e compartilhar tours virtuais relacionados a bens de grande valor agregado e no futuro acrescentar a funcionalidade de criar tours virtuais. A arquitetura será baseada em uma abordagem cliente-servidor, com componentes específicos para a criação e visualização dos tours.

2. Requisitos

2.1. Requisitos Funcionais

Autenticação de usuários: permitir que os usuários criem contas, façam login e gerenciem suas informações pessoais.

Navegação interativa: permitir que os usuários naveguem pelos tours virtuais, incluindo a rotação de visualização, zoom e movimentação por toda a área do produto.

Informações sobre o produto: incluir detalhes, descrições, vídeos, áudios e metadados relevantes sobre cada produto.

Compartilhamento de tours: permitir que os usuários compartilhem os tours com outras pessoas por meio de links ou mídias sociais.

Upload de imagens 360°: permitir que os vendedores e profissionais relevantes possam carregar as imagens em 360°.

Criação de tours virtuais: permitir que os usuários criem seus próprios tours virtuais, conectando as imagens em 360° para criar uma experiência imersiva.

Integração com outras plataformas: possibilitar a integração da aplicação com outras plataformas de mesmo nicho, como sites e portais de venda, para facilitar a inclusão de tours nas listagens.

2.2. Requisitos Não Funcionais

Desempenho: garantir que a aplicação seja rápida e responsiva ao carregar e navegar pelos tour virtuais, proporcionando uma experiência fluida ao usuário

Escalabilidade: garantir que a aplicação seja capaz de lidar com um grande número de usuários e tour virtuais simultaneamente, sem degradar o desempenho.

Compatibilidade com outros dispositivos: certificar-se de que a aplicação seja compatível com uma variedade de dispositivos, incluindo desktops, smartphones e dispositivos de realidade virtual.

Segurança: proteger as informações dos usuários, implementando medidas de segurança, como criptografia de dados e autenticação adequada.

Disponibilidade: assegurar que a aplicação esteja disponível para acesso e utilização a maior parte do tempo, minimizando o tempo de inatividade e interrupções.

Usabilidade: projetar uma interface intuitiva e fácil de usar, com instruções claras e elementos de navegação simples, garantindo que os usuários possam aproveitar ao máximo a experiência de tour virtual.

Qualidade de imagem: garantir que as imagens em 360° sejam de alta qualidade e apresentem uma representação precisa dos imóveis, para fornecer uma experiência imersiva e realista aos usuários.

3. Contexto da Arquitetura

3.1. Funcionalidades e restrições Arquiteturais

As principais funcionalidades do sistema de Tour Virtual serão:

- Visualização de tours virtuais em um ambiente 360° interativo.

- Busca de anúncios de interesse do usuário.
- Compartilhamento dos tours virtuais com outros usuários.
- Interação entre os usuários que buscam anúncios de seu interesse e os usuários que disponibilizam os anúncios.
- Criação de tours virtuais, permitindo a adição de pontos de interesse e informações relacionadas ao produto de interesse.

As principais restrições arquiteturas incluem:

- Acesso a recursos gráficos e de processamento para a renderização dos ambientes 360° e 3D.
- Acesso a câmeras 360° de alta qualidade e que permitam a visualização do ambiente.
- Armazenamento dos dados dos tours virtuais (imagens, vídeos, áudios, metadados) de forma segura e escalável.
- Suporte a múltiplos dispositivos, incluindo computadores, tablets, smartphones e óculos de realidade virtual.

3.2. Atributos de Qualidades Prioritários

Os atributos de qualidade prioritários para o sistema de tour virtual são:

- Usabilidade: o sistema deve ser fácil de usar, oferecendo uma experiência intuitiva na criação e visualização dos tours.
- Escalabilidade: o sistema deve ser capaz de ser expandido de forma rápida, segura e com a menor complexidade possível.
- Desempenho: o sistema deve ser capaz de renderizar os ambientes de forma eficiente, proporcionando uma experiência fluida ao usuário.
- Portabilidade: o sistema deve ser acessível em diferentes dispositivos e navegadores.

4. Representação da Arquitetura

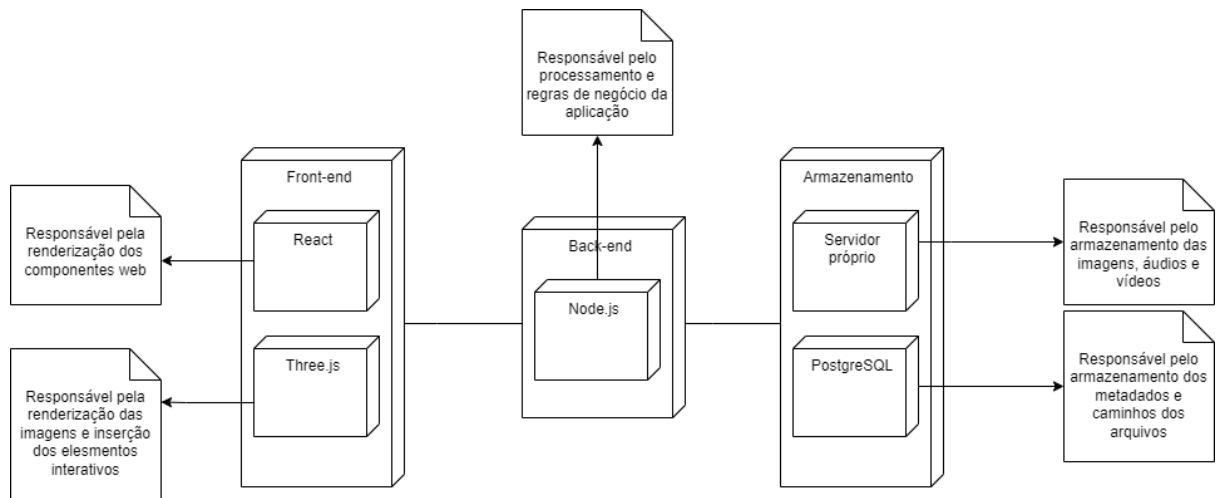


Figura 1 - Representação da arquitetura

5. Ponto de vista dos Casos de Uso

5.1. Descrição

A aplicação será responsável por ser um repositório para os comerciantes que desejam vender seus imóveis e veículos. Assim, o usuário comprador irá visitar o site em busca de encontrar o produto desejado, podendo fazer pesquisas ou entrar através de links diretos. Ele poderá visualizar os anúncios e fazer interações com os tours virtuais disponíveis e compartilhá-los se quiser. Será disponibilizado também uma forma de o comprador entrar em contato com o responsável pelo produto através do e-mail, por exemplo. Além disso, o comprador poderá fazer login e editar seus dados cadastrais. Em um futuro próximo, a aplicação deve ser capaz de permitir que os vendedores possam criar seus próprios tours, sem a necessidade de que a TourExperience faça esse trabalho.

5.2. Visão de Casos de Uso

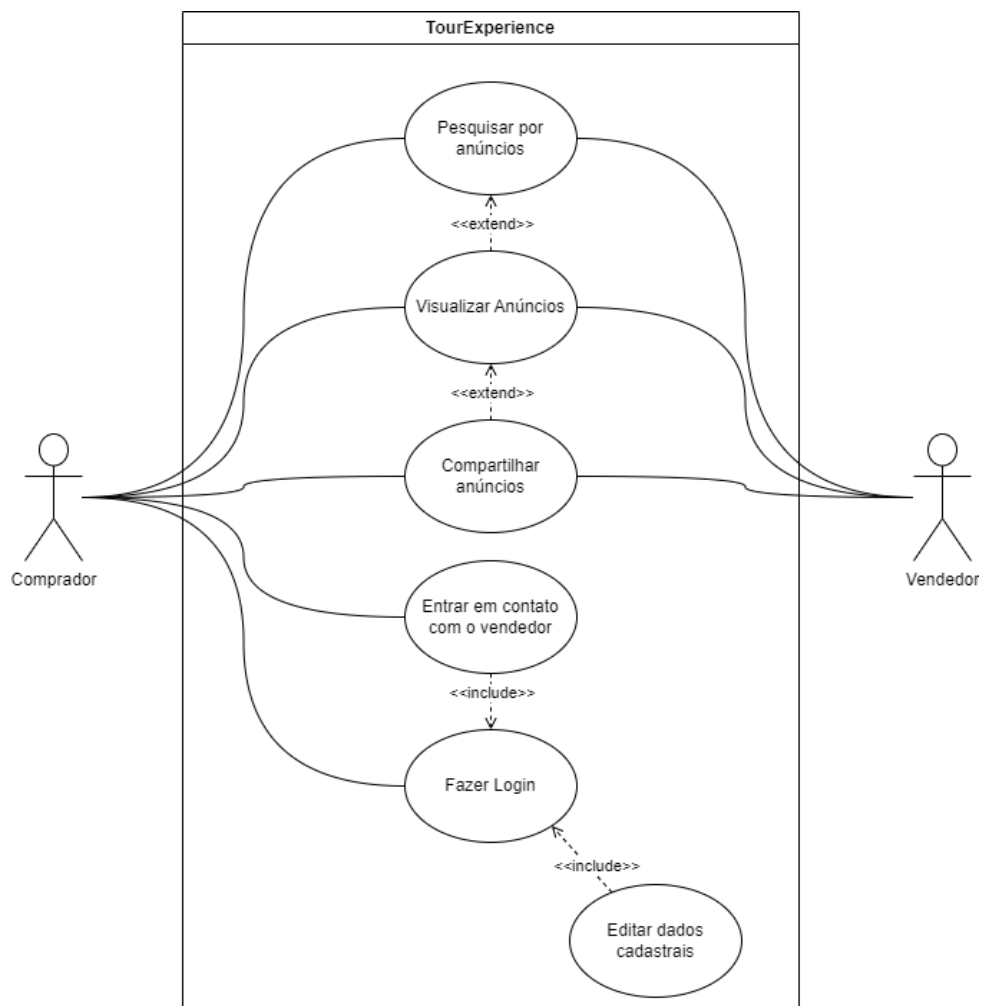


Figura 2 - Diagrama de Casos de uso 1

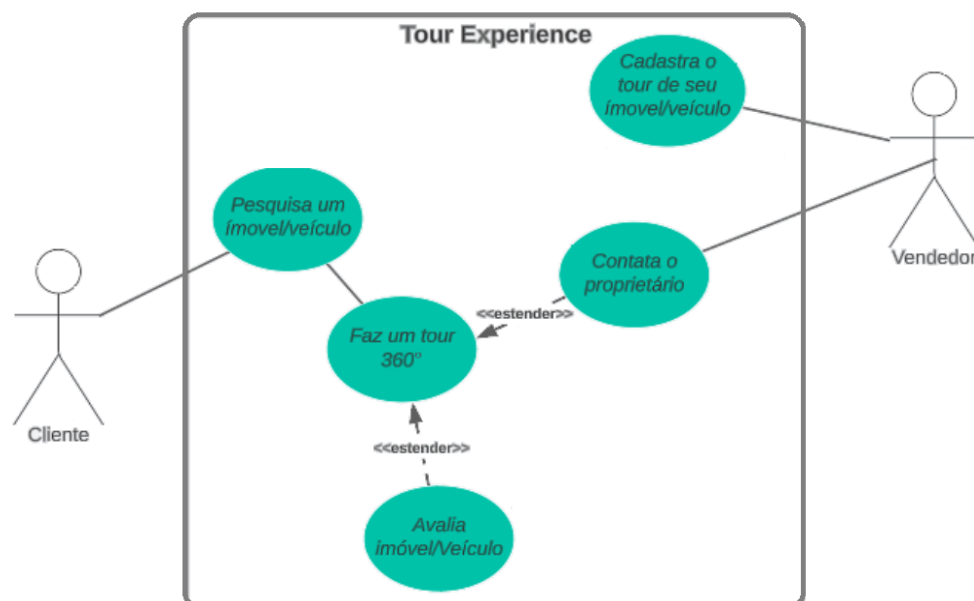


Figura 3 - Diagrama de Casos de uso 2

6. Decisões arquiteturais

6.1. Requisitos arquiteturais

As decisões tomadas nesse projeto levam em consideração os requisitos não funcionais e funcionais, de forma que sejam atendidas da melhor forma possível.

- **Requisitos Funcionais**

Os requisitos funcionais incluem autenticação de usuários, navegação interativa, informações sobre o produto, compartilhamento de tours, upload de imagens 360°, criação de tours virtuais, integração com outras plataformas, que podem ser atendidos através do uso do Node.js no Back-end, React e Three.js no Front-end e armazenamento Firebase.

- **Requisitos Não Funcionais**

Os requisitos não funcionais consistem em garantir um bom desempenho, escalabilidade, compatibilidade com outros dispositivos, segurança, disponibilidade, usabilidade e qualidade de imagem, que podem ser garantidos com a potência de processamento adequada para lidar com as solicitações dos usuários e renderização dos ambientes virtuais 360°, memória RAM suficiente para armazenar dados temporários e caches, espaço de armazenamento para a aplicação, banco de dados e arquivos de mídia e conexão de rede de alta velocidade.

6.2. Alternativas consideradas

Durante o processo de tomada de decisões arquiteturais, várias alternativas foram consideradas para atender aos requisitos funcionais e não funcionais do projeto TourExperience. Aqui estão algumas das principais alternativas avaliadas:

- **Tecnologia de Renderização 3D:**

- **Alternativa 1: Three.js**

- Uso do Three.js para renderização 3D no navegador.
- Framework bem estabelecido e amplamente utilizado para gráficos 3D.
- Facilita a criação de ambientes virtuais imersivos.
- Boa documentação e suporte da comunidade.

- **Alternativa 2: A-Frame**
 - Utilização do A-Frame, um framework para criação de experiências de realidade virtual (VR) no navegador.
 - Abstrai complexidades de renderização 3D, sendo mais amigável para desenvolvedores menos experientes em gráficos 3D.
 - Pode reduzir a flexibilidade e personalização em comparação com o Three.js.
 - **Alternativa 3: JQuery**
 - Utilização de JQuery para o desenvolvimento da parte Web da aplicação.
 - Facilita a manipulação de elementos Web.
 - Requer experiência.
 - Dificuldade de integração com outras ferramentas.
- **Plataforma de Armazenamento:**
- **Alternativa 1: Firebase**
 - Utilização do Firebase para armazenamento de dados, autenticação e compartilhamento de tours.
 - Integração simplificada de recursos como autenticação de usuários e armazenamento de imagens.
 - Escalabilidade e disponibilidade gerenciadas pelo Firebase.
 - Restrição a planos e custos do Firebase.
 - **Alternativa 2: Implantação Própria**
 - Configuração e gerenciamento de servidores próprios para armazenamento e autenticação.
 - Maior controle sobre infraestrutura e custos.
 - Necessidade de recursos adicionais para garantir escalabilidade e disponibilidade.
 - **Alternativa 1: PostgreSQL**
 - Utilização do Postgresql SQL para armazenamento de dados, autenticação e compartilhamento de tours.
 - Integração simplificada de recursos como autenticação de usuários e armazenamento de imagens.
 - Permite uma fácil utilização.

- A equipe já possui certa experiência com a ferramenta.
- **Linguagem e Framework de Desenvolvimento Back-end:**
 - **Alternativa 1: Node.js com NextJS**
 - Uso do Node.js com o framework NextJS para o backend.
 - Eficiência na manipulação de solicitações de rede.
 - Grande quantidade de módulos e pacotes disponíveis.
 - Alta escalabilidade.
 - Facilidade de manutenção e uso.
 - Requer familiaridade com JavaScript.
 - **Alternativa 2: Outras Linguagens e Frameworks**
 - Exploração de alternativas como Python (Django), Ruby (Ruby on Rails), Java (Spring), entre outros.
 - Escolha baseada na experiência da equipe e nas necessidades específicas do projeto.
- **Serviço de Hospedagem:**
 - **Alternativa 1: Hospedagem na Nuvem (por exemplo, AWS, Azure, Google Cloud)**
 - Utilização de serviços de nuvem para hospedar a aplicação.
 - Escalabilidade automática e alta disponibilidade.
 - Possível curva de aprendizado e gerenciamento de configurações na nuvem.
 - **Alternativa 2: Hospedagem Tradicional**
 - Implantação em servidores físicos ou virtuais gerenciados internamente.
 - Maior controle sobre a infraestrutura, mas com responsabilidade adicional de manutenção.

As decisões finais foram baseadas em uma análise detalhada das alternativas, levando em consideração o equilíbrio entre desempenho, escalabilidade, custos, facilidade de desenvolvimento e requisitos específicos do projeto TourExperience.

6.3. Decisões tomadas

- **Front-end**

Foi decidido por utilizar React, uma biblioteca de JavaScript, juntamente com o framework NextJS para o desenvolvimento da parte Web da aplicação. Isso garante para o projeto um desenvolvimento com maior velocidade, mais assertivo e com uma escalabilidade e manutenção garantidas, além de proporcionar uma ótima gama de ferramentas, atingindo assim a usabilidade. Além disso, para a criação e utilização dos Tours Virtuais foi escolhida a biblioteca ThreeJS, pois é uma biblioteca relativamente simples e que não demandaria dos desenvolvedores utilizarem outros meios para criação das funcionalidades, ficando somente com a parte de código, excluindo assim, a necessidade de se utilizar ferramentas como Unity.

- **Back-end**

Para o Back-end a aplicação deve utilizar NodeJS e seu framework NestJS. Essa escolha se dá pela facilidade e versatilidade de desenvolvimento. Além disso, o NestJS proporciona que a aplicação seja desenvolvida como uma Rest API, funcionando basicamente como um MVC sem a View. Sendo assim, a aplicação seria composta por microsserviços, tendo grande escalabilidade e facilidade na manutenção;

- **Armazenamento**

Para o armazenamento de dados, foi decidido utilizar uma combinação de soluções. O armazenamento principal dos PostgreSQL. Além disso, para armazenamento de arquivos de mídia, como imagens 360°, a aplicação fará uso de um serviço de armazenamento local, por meio de um servidor próprio, isso inicialmente. No futuro a aplicação deve utilizar serviços em nuvem para armazenar essas informações.

- **Infraestrutura**

A infraestrutura da aplicação será hospedada em uma plataforma de nuvem, como a Amazon Web Services (AWS) ou Microsoft Azure. Essa escolha se deve à capacidade de escalabilidade, gerenciamento simplificado e alta disponibilidade oferecidos por essas plataformas. A aplicação será implantada em containers Docker, o que facilita o

empacotamento e a implantação consistente em diferentes ambientes.

- **Componentes de Hardware**

Para a execução da aplicação, os componentes de hardware incluirão servidores virtuais na infraestrutura de nuvem escolhida. Esses servidores serão dimensionados conforme as necessidades de processamento, memória e armazenamento da aplicação. O uso de máquinas virtuais ou instâncias de contêineres dependerá da preferência e da capacidade da plataforma de nuvem selecionada. O dimensionamento correto desses componentes garantirá um desempenho ideal da aplicação e uma experiência satisfatória para os usuários.

Essas decisões arquiteturais visam garantir uma aplicação TourExperience escalável, de alto desempenho, com uma interface rica e interativa para os usuários explorarem locais virtuais de forma imersiva.

6.4. Impacto nas partes interessadas

As decisões arquiteturais mencionadas para o software "Tour Experience" têm impactos significativos em diferentes partes interessadas. Vamos detalhar como cada uma delas pode influenciar diferentes grupos envolvidos:

- **Front-end baseado em React e Three.js:**

Usuários finais: Os usuários do software serão beneficiados com uma experiência de usuário interativa e imersiva. O uso do React permitirá a criação de uma interface responsiva e amigável, enquanto o Three.js possibilitará a renderização de imagens em 360 graus, proporcionando uma sensação mais realista durante a visualização das tours virtuais.

Back-end baseado em Node com Next.js:

Desenvolvedores e equipe de TI: A escolha do Node como tecnologia para o back-end permitirá o uso da mesma linguagem (JavaScript) tanto no front-end quanto no back-end, facilitando o compartilhamento de código e conhecimento entre os desenvolvedores. O Next.js, que é baseado no Node, fornecerá recursos avançados para otimização de páginas, como renderização do lado do servidor (SSR) e geração de páginas estáticas (SSG), melhorando o desempenho e a experiência do usuário.

- **Funcionamento como um MVC:**

Desenvolvedores e equipe de TI: A adoção do padrão MVC (Model-View-Controller) ajudará a organizar o código em camadas distintas, facilitando a manutenção e a evolução do software. O modelo permite que diferentes partes da equipe trabalhem em suas respectivas áreas (Modelo, Visualização e Controle) sem afetar diretamente as outras, melhorando a colaboração.

Servidor próprio de armazenamento para dados e imagens:

Usuários finais: Com um servidor dedicado para armazenamento, os usuários podem esperar um acesso mais rápido às imagens e conteúdos do software. Isso reduzirá o tempo de carregamento e melhora a experiência geral do usuário durante a navegação nas tours virtuais.

Equipe de TI: Ao gerenciar o próprio servidor de armazenamento, a equipe terá controle total sobre a infraestrutura e poderá ajustá-la conforme necessário para atender às demandas de armazenamento de dados e imagens. Isso permitirá escalabilidade e maior controle sobre o gerenciamento de recursos.

- **Servidor de Armazenamento:**

Equipe de TI: O armazenamento e o gerenciamento de grandes quantidades de dados representam um desafio significativo para a equipe de TI. A implementação de boas práticas de design de banco de dados, indexação adequada e otimização de consultas será essencial para garantir um desempenho eficiente da aplicação.

Usuários finais: O tratamento eficiente dos dados permitirá uma experiência de usuário mais fluida e rápida durante a navegação pelas tours, evitando atrasos ou lentidão na exibição das informações.

Em resumo, as decisões arquiteturais tomadas para o "Tour Experience" têm o objetivo de fornecer uma experiência imersiva e responsiva para os usuários finais, ao mesmo tempo que facilitam o desenvolvimento e a manutenção da aplicação para a equipe de TI. O servidor próprio de armazenamento permite um controle mais efetivo sobre o desempenho e escalabilidade, embora a gestão de uma grande quantidade de dados exige uma abordagem cuidadosa para garantir o bom funcionamento do sistema.

7. Ponto de Vista do Projetista

7.1. Visão Geral

Como projetista a aplicação TourExperience, o foco é criar uma estrutura cólica e modular que atenda os requisitos funcionais e não funcionais do software. O foco é garantir a usabilidade, escalabilidade, portabilidade e desempenho da aplicação, permitindo que ela evolua de forma eficiente conforme novas funcionalidades são integradas e alterações sejam feitas. A aplicação será Web, funcionando através de Rest API.

7.2. Visão de Componentes

A aplicação será dividida em diversos componentes que colaboram entre si para fornecer as funcionalidades do sistema. São eles: frontend, backend, microsserviços, bibliotecas e módulos compartilhados e serviços terceiros.

- **Frontend**
 - Descrição: responsável pela interface do usuário e montagem dos tours. Permite que os usuários interajam com a aplicação e explorem os locais virtuais de forma imersiva.
 - Tecnologias: HTML, CSS, JavaScript, React, Next.JS, PANOLENS, Three.js.
- **Backend**
 - Descrição: responsável por processar as requisições do frontend, gerenciar os dados, as regras de negócio e integração com outros sistemas.
 - Tecnologias: Node.JS, Nest.JS, Firebase (servidor próprio).
- **Microsserviços**
 - Descrição: componentes independentes e especializados que fornecem funcionalidades específicas da aplicação.
 - Tecnologias: arquitetura de microsserviços, containers (Docker), etc.
- **Bibliotecas e módulos compartilhados**
 - Descrição: conjunto de bibliotecas e/ou módulos compartilhadas entre componentes, visando a reutilização de código e padronização.
 - Tecnologias: NPM e/ou outros gerenciadores de pacote.
- **Serviços de terceiros**
 - Descrição: integração com serviços externos, como APIs de redes sociais ou outros sites, serviços de mapa para exibir a localização dos produtos, dentre outros.

7.3. Detalhamento das Camadas

A arquitetura do sistema será organizada em camadas, seguindo um modelo cliente-servidor. Serão as seguintes: camada de interface de usuário (frontend), camada de aplicação (backend), camada de microsserviços e camada de dados.

- **Camada de Interface de Usuário**
 - Responsável por apresentar a interface visual da aplicação aos usuários.
 - Recebe e processa as integrações com o usuário.
 - Monta e gerencia os tours.
 - Comunica-se com o backend para obter e enviar dados.
- **Camada de aplicação**
 - Responsável por receber as requisições do frontend e processá-las.
 - Contém a lógica de negócio da aplicação.
 - Realiza chamadas a microsserviços e serviços de terceiros.
- **Camada de microsserviços**
 - Composta por vários micro serviços especializados em funções.
 - Realizam tarefas como o gerenciamento de usuários, autenticação, integração social e outros.
 - Independentes, podem ser escalados e atualizados separadamente.
- **Camada de dados**
 - Responsável por armazenar os dados da aplicação.
 - Utiliza bancos de dados e sistemas de armazenamento adequados.

8. Ponto de Vista do Desenvolvedor

8.1. Visão Geral

A aplicação TourExperience é um software desenvolvido para que os explorem locais virtuais de forma imersiva. Ela é projetada para proporcionar uma experiência realista e interativa, permitindo que os usuários naveguem por ambientes virtuais 3D e imagens 360°, interajam com objetos virtuais e acessem tours guiados por especialistas, integração de informações históricas e culturais, opções de personalização do ambiente virtual e suporte para integrações sociais como um canal de comunicação entre compradores e vendedores.

8.2. Visão Lógica

A arquitetura será baseada em um modelo cliente-servidor, onde o cliente é responsável por renderizar os ambientes virtuais, lidar com a integração do usuário e se comunicar com o servidor para obter informações adicionais. O servidor será responsável por gerenciar os dados dos locais virtuais, dos anúncios e dos usuários, irá fornecer recursos de autenticação e autorização, além de possibilitar a integração entre usuários. A aplicação também poderá utilizar novas tecnologias de renderização para melhorar a qualidade dos recursos, melhorar o desempenho e permitir sua utilização em diferentes dispositivos.

8.3. Visão de Segurança

A segurança da aplicação irá abordar a proteção dos dados dos usuários, a autenticação e autorização adequadas, além de prevenir acessos não autorizados.

8.3.1. Detalhamento da Segurança

Para a implementação da funcionalidade citadas no tópico anterior será necessário abordar os seguintes tópicos:

- Criptografia dos dados sensíveis: a transmissão de dados entre o cliente e o servidor deve possuir um certo grau de segurança, utilizando até mesmo criptografia.
- Autenticação de usuários: os usuários devem ser identificados de maneira única através de login e senha.
- Controles de acesso: é necessário que os usuários tenham acesso somente a recursos destinados para o seu perfil.
- Proteção contra ataques: medidas para proteger a aplicação contra ataques conhecidos e comuns, como ataques de injeção de código, negação de serviço e de força bruta.
- Realização de testes de segurança: devem ser realizados testes para identificar e corrigir erros e vulnerabilidades, além de serem realizados testes de penetração.
- Treinamento e conscientização: os desenvolvedores deverão ser treinados e conscientizados com relação às boas práticas de segurança, como a validação adequada de entrada de dados, prevenindo assim, ataques de injeção de código e vulnerabilidades relacionadas.
- Monitoramento: monitoramento de atividades suspeitas e detecção de possíveis ataques, além de possuir logs do sistema.

9. Ponto de Vista do Implantador

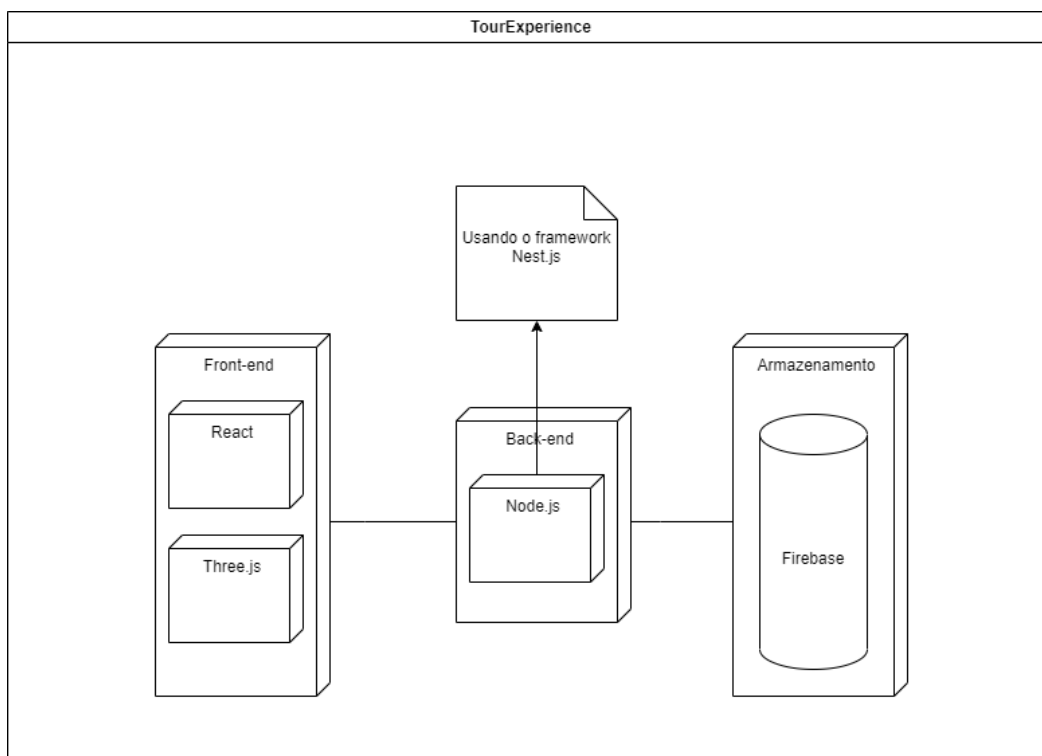


Figura 4 - Arquitetura de Implantação

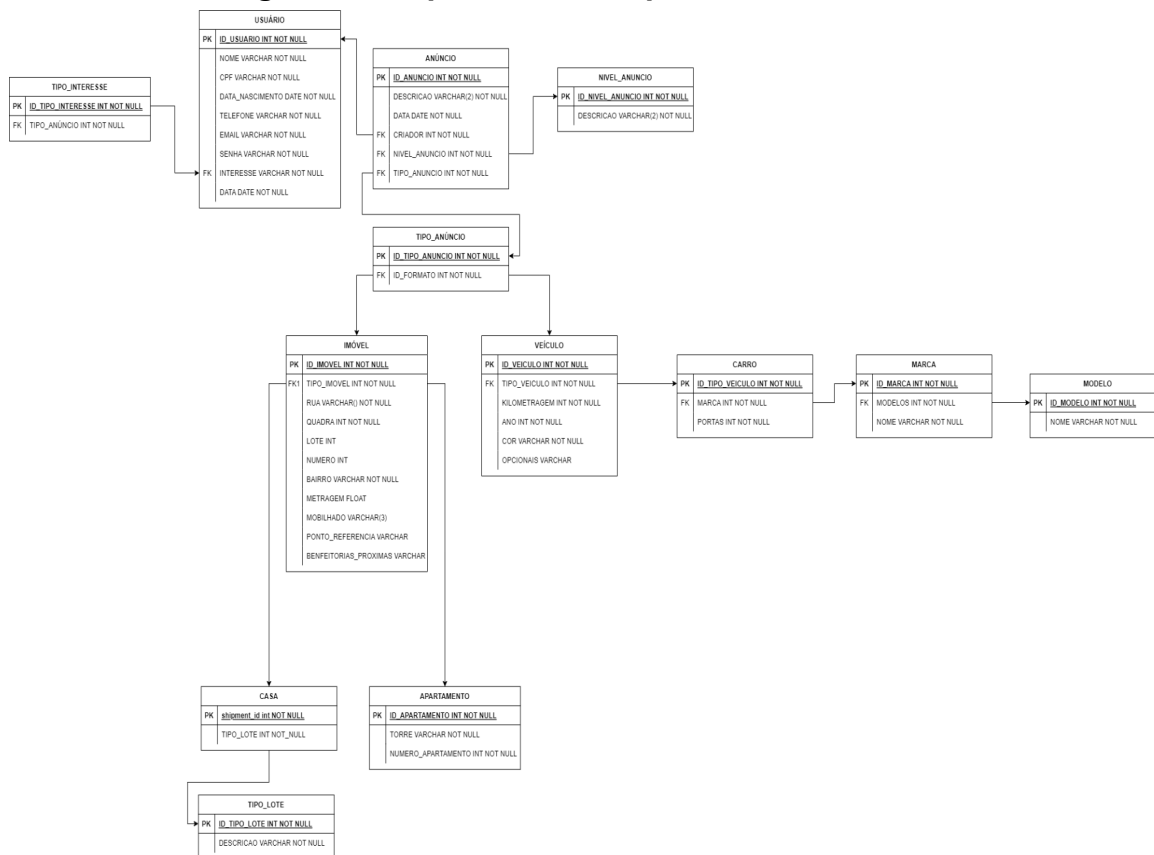


Figura 5 - Diagrama de Banco de Dados

Diagrama de banco de dados: [aqui](#)

9.1. Visão Geral

O ponto de vista do implantador tem como objetivo fornecer uma visão geral da infraestrutura física necessária para implantar e operar a aplicação Tour Experience. Nesta perspectiva, serão detalhados os componentes de hardware, as dependências externas e os requisitos de ambiente para garantir a disponibilidade, escalabilidade e desempenho adequado do sistema.

9.2. Visão Física

A infraestrutura física do sistema Tour Experience será composta por diferentes nós-físicos, que abrigam os componentes do hardware e software necessários para a execução da aplicação. Esses nós-físicos podem ser distribuídos geograficamente para melhorar a eficiência e garantir a alta disponibilidade.

9.2.1. Detalhamento dos nós

Servidores Web:

Descrição:

Nós responsáveis por hospedar a aplicação web e disponibilizar a interface do usuário.

Requisitos:

- Potência de processamento adequada para lidar com as solicitações dos usuários e renderização dos ambientes virtuais 360°.
- Memória RAM suficiente para armazenar dados temporários e caches.
- Espaço de armazenamento para a aplicação, banco de dados e arquivos de mídia.
- Conexão de rede de alta velocidade para garantir a resposta rápida ao tráfego de usuários.
- Certificado SSL para garantir a segurança na transmissão de dados sensíveis.

Servidores de Banco de Dados:

Descrição:

Nós responsáveis pelo armazenamento dos dados da aplicação, incluindo informações dos usuários, tours virtuais, mídias e metadados.

Requisitos:

- Potência de processamento para manipulação eficiente de consultas e operações no banco de dados.
- Grande capacidade de armazenamento para acomodar a grande quantidade de dados dos tours virtuais.
- Sistema de armazenamento resiliente e com backups regulares para garantir a recuperação de dados em caso de falhas.

Servidores de Armazenamento de Mídia:**Descrição:**

Nós responsáveis por armazenar as imagens 360°, vídeos e áudios utilizados nos tours virtuais.

Requisitos:

- Grande capacidade de armazenamento para acomodar arquivos de mídia em alta qualidade.
- Taxas de transferência de rede elevadas para possibilitar o rápido acesso aos recursos de mídia.

Balanceador de Carga:**Descrição:**

Nó responsável por distribuir o tráfego de usuários entre os servidores web para garantir uma carga equilibrada.

Requisitos:

- Capacidade para gerenciar e distribuir eficientemente o tráfego de usuários em tempo real.
- Tolerância a falhas para garantir a disponibilidade contínua do serviço.

Servidores de Backup e Recuperação:**Descrição:**

Nós responsáveis por manter cópias de segurança dos dados e configurações do sistema para possibilitar a recuperação em caso de desastres.

Requisitos:

- Capacidade de armazenamento para acomodar as cópias de segurança de todos os servidores.
- Sincronização regular com os servidores principais para garantir a consistência dos dados.

Servidores de Monitoramento e Log:

Descrição:

Nós responsáveis por monitorar o desempenho do sistema e coletar registros de atividades para análise e resolução de problemas.

Requisitos:

- Capacidade de armazenamento para os registros de atividades de todos os servidores.
- Ferramentas de monitoramento para acompanhar o desempenho dos componentes do sistema em tempo real.

Firewall e Roteadores:

Descrição:

Nós responsáveis por garantir a segurança da rede e controlar o tráfego de entrada e saída do sistema.

Requisitos:

- Políticas de segurança configuradas para bloquear tráfego malicioso.
- Filtragem e inspeção de pacotes para garantir a integridade dos dados.

Data Centers ou Serviços de Cloud:

Descrição:

Nós físicos que abrigam toda a infraestrutura do sistema Tour Experience.

Requisitos:

- Ambiente controlado com resfriamento e energia redundante para garantir a disponibilidade contínua.
- Conexão de internet de alta velocidade e tolerância a falhas para assegurar a disponibilidade e a escalabilidade.