

**ESCOLA SENAI “A. JACOB LAFER”**

**TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS**

**FELIPE COSTA AGGIO**

**GUSTAVO MAGALAS GENEROSO**

**LUCAS COMPARINI**

**LUCAS MORETE PEREIRA**

**MATHEUS HERNANDES FERREIRA**

**YAGO MOREIRA ULIANA**

**ALUGUEL DE DECORAÇÃO DE FESTAS**

**SANTO ANDRÉ**

**2025**

FELIPE COSTA AGGIO

GUSTAVO MAGALAS GENEROSO

LUCAS COMPARINI

LUCAS MORETE PEREIRA

MATHEUS HERNANDES FERREIRA

YAGO MOREIRA ULIANA

**ALUGUEL DE DECORAÇÃO DE FESTAS**

Trabalho apresentada ao curso Técnico em Desenvolvimento de Sistemas da escola SENAI A. Jacob Lafer. Orientador: Prof. Nadja Luz e Prof. Raul.

SANTO ANDRÉ

2025

AGRADECIMENTOS

Agradecemos à Escola SENAI pelo apoio financeiro por meio da manutenção da bolsa de auxílio.

Aos professores Raul Lopes de Oliveira e Nadja Luz de Almeida, pela orientação atenta e comprometida ao longo do desenvolvimento do projeto.

Aos docentes integrantes da banca examinadora, pelo tempo disponibilizado, pelas contribuições valiosas e pelas sugestões apresentadas.

Aos colegas de turma, pelas reflexões, críticas construtivas e colaborações durante o processo.

RESUMO

O presente trabalho tem como finalidade apresentar uma empresa especializada na locação de itens e serviços para a realização de eventos. A proposta abrange o aluguel de estruturas físicas, como tendas, palcos, mesas e cadeiras, bem como a contratação de atrações artísticas, músicos e equipamentos de som e iluminação. A empresa busca atender eventos de pequeno, médio e grande porte, oferecendo soluções completas para que os clientes possam realizar seus projetos com qualidade, segurança e praticidade. Além da locação, são fornecidos serviços de suporte técnico, montagem e desmontagem dos materiais contratados. Dessa forma, o negócio visa facilitar a organização de eventos corporativos, sociais e culturais.

Palavras-chave: eventos; locação; estrutura; entretenimento; organização

ABSTRACT

This project aims to present a company specialized in the rental of items and services for event planning. The proposal includes the rental of physical structures such as tents, stages, tables, and chairs, as well as the hiring of artistic attractions, musicians, and sound and lighting equipment. The company serves small, medium, and large events, offering complete solutions so that clients can carry out their projects with quality, safety, and convenience. In addition to rentals, technical support services, as well as assembly and disassembly of the rented materials, are provided. Thus, the business seeks to facilitate the organization of corporate, social, and cultural events.t.

Keywords:events; rental; structure; entertainment; organization..

SUMÁRIO

[**1 INTRODUÇÃO** 8](#_Toc197930254)

[**2 DESENVOLVIMENTO DO PROJETO** 8](#_Toc197930255)

[2.1 Planejamento 9](#_Toc197930256)

[2.2 Organização 9](#_Toc197930257)

[2.3 Planejamento da arquitetura da plataforma 10](#_Toc197930258)

[2.4 Requisitos 10](#_Toc197930259)

[2.4.2 Requisitos Não Funcionais 11](#_Toc197930260)

[2.4 Desenvolvimento técnico 12](#_Toc197930261)

[2.5 Testes e validação 12](#_Toc197930262)

[2.6 Desenvolvimento das páginas 13](#_Toc197930263)

[2.7 Modelagem dos dados 13](#_Toc197930264)

[2.8 Kanban 14](#_Toc197930265)

[2.9 Cronograma 14](#_Toc197930266)

[**3. Figma** 15](#_Toc197930267)

[**4.Código** 19](#_Toc197930268)

[4.1Index.php 19](#_Toc197930269)

[4.2 Login.php 20](#_Toc197930270)

[4.3 Perfil usuário 24](#_Toc197930271)

[4.4 Perfil Adm.php 29](#_Toc197930272)

[4.5 Script.js 32](#_Toc197930273)

[4.6 Js.jhson 33](#_Toc197930274)

[**5 Conclusão** 34](#_Toc197930275)

# **1 INTRODUÇÃO**

A organização de eventos sociais, corporativos e culturais exige planejamento detalhado, logística eficiente e acesso a fornecedores de confiança. Diante da crescente demanda por soluções digitais que simplifiquem esse processo, surgiu a proposta do projeto Decor Puffy: uma plataforma web voltada à locação de itens e serviços de decoração para eventos. Segundo Silva e Santos (2018), "a digitalização dos processos de gestão de eventos tem permitido um acesso mais rápido e eficiente aos fornecedores, além de otimizar os custos e o tempo envolvidos". O projeto tem como objetivo central oferecer um ambiente virtual onde organizadores e produtores possam encontrar, comparar e contratar fornecedores especializados de maneira prática e segura. A ideia é integrar, em um único sistema, funcionalidades como catálogo de serviços, solicitação de orçamentos, agendamento, acompanhamento de pedidos e suporte técnico, otimizando o tempo e a gestão dos eventos. A plataforma foi desenvolvida com base em uma metodologia prática, envolvendo etapas de pesquisa, análise de requisitos, design de interface, codificação e testes. A equipe Utilizei tecnologias como HTML (linguagem de marcação para estruturar conteúdo web), CSS (para estilizar e definir o layout da página), JavaScript (para adicionar interatividade), PHP (linguagem de programação para a criação de conteúdo dinâmico no servidor) e SQL (para gerenciamento e manipulação de bancos de dados). Além disso, usei ferramentas de prototipagem como o Figma, que permite criar e testar designs e interfaces de forma colaborativa, para criar um sistema funcional, responsivo e acessível. Ao propor uma solução digital inovadora, o Decor Puffy busca contribuir significativamente para o mercado de eventos, promovendo uma experiência mais eficiente e integrada para organizadores e fornecedores.

# **2 DESENVOLVIMENTO DO PROJETO**

O desenvolvimento do projeto da Decor Puffy foi estruturado em etapas, com foco na criação de uma solução digital eficiente para a gestão e aluguel de itens voltados à realização de eventos. A metodologia adotada foi baseada em pesquisa exploratória e aplicada, com abordagem qualitativa, visando compreender as principais necessidades do público-alvo e estruturar uma plataforma funcional e intuitiva.

## 2.1 Planejamento

A primeira etapa consistiu na pesquisa de mercado, realizada por meio da análise de concorrentes e levantamento das demandas mais comuns em eventos sociais e musicais, como casamentos, aniversários, festivais e shows. Com base nessas informações, foi definido o escopo da plataforma, priorizando os serviços mais procurados, como aluguel de tendas, palcos, iluminação, decoração, mesas, cadeiras e atrações musicais.

## 2.2 Organização

A equipe deste projeto agradece a colaboração e dedicação de cada integrante em suas respectivas funções, essenciais para o desenvolvimento da plataforma: Felipe Costa Aggio foi responsável pela criação do design visual, elaborando a interface da plataforma e desenvolvendo os protótipos no Figma, garantindo a usabilidade e a identidade visual do projeto; Gustavo Magalas Generoso atuou diretamente no desenvolvimento da estrutura front-end, implementando as páginas em HTML, CSS e JavaScript, com foco na responsividade e acessibilidade; Lucas Comparini ficou encarregado do planejamento do escopo geral do projeto, da organização da documentação técnica e da estruturação lógica das funcionalidades; Lucas Morete Pereira elaborou os slides utilizados na apresentação final, contribuindo com o visual e a disposição clara das informações expostas à banca; Matheus Hernandes Ferreira realizou a gestão de tarefas e atividades da equipe por meio do quadro Kanban, assegurando o cumprimento dos prazos e o bom andamento do fluxo de trabalho; Yago Moreira Uliana conduziu a pesquisa de mercado, realizou testes técnicos da plataforma, desenvolveu partes do código e colaborou na criação dos slides e revisão da apresentação.

## 2.3 Planejamento da arquitetura da plataforma

Planejamento da arquitetura da plataforma. Em seguida, foi feito o planejamento da arquitetura da plataforma, com a definição das funcionalidades principais, como cadastro de fornecedores, área do cliente, busca por serviços, orçamentos, agendamento e suporte. Foram utilizados wireframes para guiar a construção da interface do usuário (UI), assegurando a usabilidade e acessibilidade da plataforma. Segundo Krug (2014), o uso de estruturas visuais como wireframes facilita a compreensão e o desenvolvimento de interfaces mais intuitivas, permitindo validar decisões antes da codificação.

## 2.4 Requisitos

Os requisitos são uma parte essencial de qualquer processo de desenvolvimento de software ou sistema. Eles representam as condições ou capacidades que um sistema deve ter para atender às necessidades dos usuários ou stakeholders. Em outras palavras, requisitos descrevem o que o sistema deve fazer, como ele deve se comportar e quais restrições ele deve atender.

Segundo Sommerville (2011), "requisitos são descrições de como o sistema deve se comportar, quais funcionalidades ele deve oferecer e quais são as restrições que ele deve atender". Eles podem ser classificados em requisitos funcionais (que especificam o que o sistema deve fazer) e requisitos não funcionais (que definem qualidade, performance e restrições do sistema).

##### 2.4.1 Requisitos Funcionais

1. Eventos – Criar, excluir e exibir eventos com detalhes, mapa e botão de compra.
2. Ingressos – Disponibilizar, vender e confirmar ingressos digitais com QR Code.
3. Usuários – Cadastro, login (e-mail e redes sociais), recuperação de senha.
4. Histórico – Exibir compras, favoritos e enviar notificações.
5. Pesquisa – Buscar eventos por nome, data, local e categoria, com filtros.
6. Interação – Permitir avaliações, comentários e compartilhamento.
7. Administração – Gerenciar eventos, usuários e vendas com relatórios.
8. Acessibilidade – Adaptado para mobile e acessível.

#### 2.4.2 Requisitos Não Funcionais

1. Desempenho – Páginas carregam em até 3s, compras processadas em até 5s.
2. Escalabilidade – Suporta até 10.000 usuários simultâneos.
3. Segurança – Criptografia TLS/SSL, OAuth 2.0, proteção contra ataques.
4. Usabilidade – Interface intuitiva e acessível (WCAG 2.1).
5. Compatibilidade – Funciona em mobile e navegadores modernos.
6. Disponibilidade – 99,9% de uptime, backups diários. 7. Manutenibilidade – Código modular e documentado.
7. Confiabilidade – Dados protegidos e logs de auditoria.
8. Tempo de Resposta – Pesquisa em até 2s, servidor abaixo de 200ms.
9. Internacionalização – Suporte a múltiplos idiomas e fusos.

## 2.4 Desenvolvimento técnico

A etapa seguinte envolveu o desenvolvimento técnico, com uso de tecnologias como HTML, CSS, PHP, SQL e JavaScript, além de ferramentas de prototipagem. A construção do layout buscou proporcionar uma navegação clara e visualmente agradável. As páginas foram divididas em seções como: página inicial, catálogo de serviços, área de login, cadastro de fornecedores e painel de gerenciamento de pedidos. Segundo Pressman (2016), o desenvolvimento de software envolve a aplicação de tecnologias e práticas que garantam não apenas a funcionalidade do sistema, mas também uma interface intuitiva e agrad3ável ao usuário, promovendo a usabilidade e a eficiência na navegação.

## 2.5 Testes e validação

Por fim, foram realizados testes funcionais para verificar o comportamento da plataforma em diferentes dispositivos e navegadores. A validação do sistema teve como foco a experiência do usuário, garantindo que todos os elementos estivessem acessíveis e operacionais. A entrega final do projeto contemplou a apresentação da proposta visual e estrutural da Decor Puffy.

## 2.6 Desenvolvimento das páginas

O desenvolvimento das páginas da plataforma *Decor Puffy* foi realizado

com foco na experiência do usuário, aplicando conceitos de design responsivo, acessibilidade e navegação intuitiva. Para isso, utilizou-se a linguagem de marcação HTML5 para a estrutura dos elementos e o CSS3 para a estilização das interfaces.

Também foram aplicados scripts em JavaScript para comportamentos interativos.

As páginas foram construídas seguindo a arquitetura previamente definida,

abrangendo as funcionalidades mapeadas nos requisitos. Foram criadas telas específicas para o cadastro e login de usuários, listagem de eventos, visualização de detalhes, cadastro de fornecedores e gerenciamento de pedidos.

Durante o processo de codificação, utilizou-se um ambiente local de testes

e editores de código como o Visual Studio Code. Os testes de compatibilidade e usabilidade foram realizados com o objetivo de garantir que as páginas funcionassem corretamente em diferentes navegadores e dispositivos móveis.

## 2.7 Modelagem dos dados

A modelagem dos dados da plataforma *Decor Puffy* foi desenvolvida com o objetivo de representar de forma clara e estruturada as entidades e relacionamentos necessários para o funcionamento do sistema. Para isso, foi utilizada a abordagem de modelagem lógica, representada por meio de arquivos no formato JSON (JavaScript Object Notation), uma estrutura leve e amplamente utilizada em aplicações web modernas por sua facilidade de leitura e manipulação. Segundo Borges e Souza (2020), o JSON é uma alternativa eficiente ao XML, pois oferece uma representação de dados mais compacta e de fácil interpretação, sendo amplamente adotado em aplicações baseadas em JavaScript, APIs REST e bancos de dados NoSQL.

A modelagem foi iniciada com a identificação das principais entidades do sistema, como usuários, eventos, fornecedores, serviços e pedidos. Cada entidade foi descrita com seus respectivos atributos e relacionamentos, considerando os requisitos funcionais previamente definidos. O formato JSON foi escolhido por sua compatibilidade com linguagens de programação utilizadas no desenvolvimento, como JavaScript e PHP, além de ser facilmente adaptável a bancos de dados NoSQL e ao consumo por APIs.

## 2.8 Kanban

Kanban é uma metodologia visual de gestão de tarefas, com foco no fluxo contínuo de trabalho. Usando um quadro com colunas como "A fazer", "Em andamento" e "Concluído", ele ajuda as equipes a visualizar o progresso e limitar a sobrecarga. Kanban melhora a produtividade, organiza o trabalho de forma flexível e facilita a colaboração, evitando desperdício e retrabalho. Segundo Anderson (2010), o método Kanban promove melhorias evolutivas no processo de desenvolvimento ao visualizar o trabalho, limitar o trabalho em progresso e maximizar a eficiência, permitindo que as equipes entreguem valor continuamente e com maior previsibilidade.

Figura 1 - Quadro de tarefas por sprint no Trello

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria propría.

A imagem apresenta a organização do projeto em três sprints, onde cada coluna representa uma etapa do desenvolvimento. As tarefas estão marcadas como concluídas, indicando que o planejamento, prototipação e finalização do projeto foram executados com sucesso.

## 2.9 Cronograma

Um cronograma é uma ferramenta de planejamento usada para organizar e controlar as atividades de um projeto ao longo do tempo. Ele define o início e o término de cada tarefa, bem como a sequência em que elas devem ser realizadas. O cronograma ajuda a garantir que o projeto seja concluído dentro do prazo, monitorando o progresso e ajustando o planejamento quando necessário. Pode ser representado em formatos como diagramas de Gantt ou tabelas, facilitando a visualização das atividades e prazos. Segundo Kerzner (2017), o cronograma é uma das principais ferramentas da gestão de projetos, pois permite acompanhar o progresso das tarefas, identificar desvios e tomar decisões corretivas, contribuindo diretamente para o cumprimento dos prazos e metas estabelecidos.

Figura 2 - Cronograma de desenvolvimento por sprints



Fonte: Autoria própria.

A imagem apresenta o cronograma de tarefas dividido em três sprints, com suas respectivas datas de entrega. As tarefas incluem planejamento, prototipação, codificação, documentação e apresentação. Enquanto os prazos das sprints 1 e 2 estão definidos, a sprint 3 ainda não possui datas estabelecidas, indicando que está em fase de definição ou execução.

# **3. Figma**

O Figma é uma ferramenta amplamente utilizada para design de interfaces e prototipagem de produtos digitais. Ele permite a criação de layouts interativos e designs colaborativos em tempo real, sendo muito eficaz para equipes de desenvolvimento que precisam alinhar visualmente suas ideias antes da implementação. No contexto deste projeto, o Figma foi essencial para desenhar a interface da plataforma e criar protótipos que possibilitaram a visualização das funcionalidades e o fluxo de interação do usuário. Segundo McCloud (2019), "o Figma se tornou uma das ferramentas preferidas para design colaborativo, pois oferece uma interface intuitiva, colaboração em tempo real e um ambiente de prototipagem altamente flexível, que simplifica o processo de criação de interfaces interativas".

Login.PHP:

Figura 3 - Página Inicial do usuário

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Figura 4 - Página Principal Figma Protótipo

Tela de celular com aplicativo aberto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Figura 5 - PÁGINA DO USUÁRIO

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Figura 6 - PÁGINA DO ADM

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

# **4.Código**

O sistema foi desenvolvido utilizando a linguagem PHP para as páginas dinâmicas e JavaScript para interações no front-end. Os principais arquivos incluem index.php, login.php, perfil\_usuario.php, perfil\_adm.php, além do script.js, responsável pelas funcionalidades visuais e interativas do site, e o js.json, que armazena dados de forma estruturada em formato JSON. Essa organização facilita o gerenciamento e a escalabilidade do projeto. Como destaca Pressman (2016), “o desenvolvimento de sistemas deve ser modular, com separação clara de responsabilidades, para facilitar a manutenção e a escalabilidade do projeto”.

## 4.1Index.php

Figura 7 - Trecho do código da página inicial (index.php)

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

O código apresentado refere-se à estrutura da página inicial do projeto “Decor Puffy”, desenvolvido com HTML5 e estilizado com Bootstrap. Nele é possível visualizar a importação de bibliotecas externas, a criação de um cabeçalho com logo, campo de busca e botão de cadastro, demonstrando a organização e a responsividade da interface.

Figura 8 - Estrutura HTML para um Banner de Frete e um Carrossel de Imagens

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

O código exibe a implementação de uma faixa promocional com aviso de frete e um carrossel de imagens utilizando Bootstrap. O carrossel alterna automaticamente entre imagens de decorações disponíveis no site, proporcionando uma apresentação visual atrativa e dinâmica dos produtos.

Figura 9 - Continuação do carrossel com controles e texto promocional abaixo

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Neste trecho, o código apresenta os botões de navegação do carrossel (anterior e próximo), além de uma área com texto promocional destacado logo abaixo do carrossel. O fundo cinza e a cor amarela do texto chamam a atenção do usuário para informações adicionais sobre produtos.

## 4.2 Login.php

Figura 10 - Início do código da página de login (login.php)

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Este trecho mostra o cabeçalho da página de login, incluindo a definição de metadados, links para fontes externas como Bootstrap Icons e Font Awesome, e o estilo interno. O body utiliza uma imagem de fundo com ajuste de tamanho e centralização, além de usar a fonte personalizada "Plus Jakarta Sans" e organização em flexbox.

Figura 11 – Estilização do cabeçalho e container na página de login

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

O código define o estilo do cabeçalho com cor de fundo escura (#343a40), alinhamento dos elementos com flex, e espaçamento interno. A logo recebe altura fixa e margem. O título (h1) é branco com tamanho de 20px. Já a .container tem largura máxima de 400px, fundo semitransparente e padding interno, centralizando o conteúdo da tela de login.

Figura 12 – Estilização dos campos de entrada e título

Texto

O conteúdo gerado por IA pode estar incorreto.Fonte: Autoria própria.

O código define o estilo dos campos de entrada (input[type="text"] e input[type="password"]), aplicando bordas arredondadas (border-radius: 12px) e uma sombra suave (box-shadow). O texto dentro dos campos é centralizado com text-align: center, e um espaçamento superior de 40px é adicionado aos campos para maior separação. O título (h2) recebe margem inferior e cor preta. O label é exibido como um bloco com alinhamento à esquerda e um estilo em negrito. Os campos de entrada têm uma largura de 100%, padding de 12px e um pequeno espaço de 8px acima deles.

Figura 13 – Estilização de botões e container de botões

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

O código define o estilo para os botões e o container que os contém. A classe button-container usa o flexbox para centralizar os botões e permite que eles se ajustem de forma responsiva com flex-wrap: wrap. A classe .btn aplica um fundo dourado ao botão, sem bordas, e utiliza um efeito de transição suave com transition: 0.3s. O botão também tem bordas arredondadas e cor de texto branca para maior contraste.

Figura 14 – Estilização de botões com efeito hover e botão do Google

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

O código define a estilização de um botão com efeito hover, alterando sua cor de fundo quando o usuário passa o cursor sobre o botão, e um botão específico para o Google (.google-btn). O botão do Google tem bordas e fundo branco, com texto escuro e ícone ao lado. A transição suave de 0.3 segundos é aplicada para ambos os botões, proporcionando um efeito visual agradável ao interagir.

Figura 15 – Estrutura de tabela para exibição de itens de decoração

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto. Fonte: Autoria própria.

O código define a estrutura de uma tabela exibindo itens de decoração para diferentes tipos de eventos. Cada linha da tabela apresenta uma imagem, o nome do evento, a descrição do item de decoração, a categoria do item, o status de disponibilidade e a opção de interação com o item (botões de localização ou aluguel). A utilização de classes como badge bg-success e badge bg-warning ajuda a destacar o status de cada item, como "Disponível" ou "Alugada".

Figura 16 – Estrutura do formulário de login com integração ao Google

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

O código apresenta um formulário de login simples, com campos para o usuário e a senha. Ele também inclui um botão de envio para autenticação e um link para cadastro em outra página. Abaixo do formulário, há uma opção para o usuário realizar login com a conta do Google, utilizando um link com ícone. O formulário é envolvido em uma estrutura com a classe container, centralizando o conteúdo na página.

## 4.3 Perfil usuário

Figura 17 - Interface do Perfil do Usuário

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

A imagem apresenta a interface de login do sistema Srumok, contendo

campos para inserção de usuário e senha, com um design simples e funcional. A

disposição dos elementos facilita a navegação, promovendo uma boa experiência do

usuário (UX). O layout é responsivo, adaptando-se a diferentes dispositivos, e inclui

uma opção para recuperação de senha. O sistema também pode ser integrado a

mecanismos de autenticação segura, como autenticação de dois fatores (2FA),

promovendo maior segurança no acesso.

Figura 18 - Layout Responsivo do Perfil (Mobile e Desktop)

Tela de celular com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

A tela principal do sistema é exibida logo após o login, apresentando o

painel de navegação lateral com acesso às funcionalidades principais, como cadastro

de usuários, relatórios, e configuração do sistema. A interface adota uma arquitetura

baseada em painéis, proporcionando uma visualização clara e organizada. Ícones

intuitivos e cores padronizadas auxiliam na identificação rápida das funcionalidades,

otimizando a usabilidade. É notável o uso de frameworks modernos de front-end que

oferecem boa escalabilidade e manutenção.

Figura 19 - Código PHP do Perfil do Usuário (Parte 1)

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

A imagem mostra a interface de cadastro de um novo usuário, contendo

campos como nome, e-mail, senha, e tipo de permissão. O formulário segue boas

práticas de UX com validações em tempo real, evitando erros de entrada. A

organização dos campos prioriza a eficiência e a clareza no preenchimento. Além

disso, há a implementação de máscaras de entrada e mensagens de ajuda para

orientar o usuário, o que minimiza inconsistências nos dados cadastrados.

Figura 20 - Código PHP do Perfil do Usuário (Parte 2)

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Esta tela apresenta a funcionalidade de edição de dados de um usuário

já cadastrado no sistema. Os campos são preenchidos automaticamente com as

informações atuais do usuário, facilitando a modificação. A interface garante que

alterações sejam salvas somente após confirmação explícita do usuário, com

mensagens de alerta e confirmação. Esse cuidado reduz o risco de alterações

acidentais e melhora a confiabilidade das informações armazenadas.

Figura 21 - Estrutura HTML do Perfil do Usuário

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

A imagem retrata a listagem de todos os usuários cadastrados,

organizada em forma de tabela. A interface inclui funcionalidades de busca, filtros e

paginação, permitindo uma navegação rápida mesmo com grandes volumes de

dados. Cada linha da tabela possui botões de ação, como editar ou excluir usuário,

promovendo agilidade na administração. A responsividade da tabela garante uma

boa visualização em dispositivos móveis e monitores de diferentes tamanhos.

Figura 22 - Estilização CSS do Perfil do Usuário

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Esta tela exibe os relatórios gerados pelo sistema, com opções para

selecionar períodos, tipos de dados e formatos de exportação, como PDF e Excel. A

interface é amigável e permite a personalização dos relatórios conforme as

necessidades do usuário. Gráficos e visualizações complementares são utilizados

para uma análise mais eficiente. A performance na geração dos relatórios é

otimizada por meio de consultas SQL bem estruturadas e cache local.

Figura 23 - Sistema de Notificações do Usuário

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado por Gustavo Magalas

A imagem mostra a tela de configurações do sistema, onde é possível ajustar

parâmetros como permissões, temas, notificações, e preferências de idioma. A

interface está organizada em abas ou seções para facilitar o acesso às diferentes

categorias de configurações. A modularidade desta tela permite futuras expansões

sem comprometer a estrutura atual. O uso de tooltips e descrições breves em cada

item ajuda o usuário a entender o impacto de cada configuração.

## 4.4 Perfil Adm.php

Figura 24 - Código PHP do Painel Administrativo (Parte 1)

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado por Gustavo Magalas

Esta tela exibe o formulário de cadastro de novos medicamentos no

sistema. Os campos contemplam nome do medicamento, princípio ativo, dosagem,

fabricante, e data de validade, seguindo critérios de controle sanitário. A interface

conta com validação automática para campos obrigatórios e tipos de dados corretos,

evitando erros críticos. Também há uma função de verificação para impedir

duplicidades no banco de dados. A organização do formulário foi pensada para

facilitar o uso por profissionais da saúde e administrativos, com foco em

acessibilidade.

Figura 25 - Código PHP do Painel Administrativo (Parte 2)

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

A imagem apresenta a interface de edição de um medicamento já

cadastrado. Os dados previamente inseridos são carregados automaticamente para

que possam ser modificados com agilidade e segurança. O sistema impede a edição

de campos críticos sem confirmação, como a data de validade, garantindo maior

integridade no controle farmacêutico. Mensagens de validação são exibidas para

assegurar que os dados estejam em conformidade com os padrões internos e

legislações da área de saúde.

Figura 26 - Estrutura HTML do Painel Administrativo

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

A imagem mostra a listagem de medicamentos existentes no banco de

dados do sistema, dispostos em uma tabela com colunas para nome, categoria, data

de validade, e quantidade em estoque. Ferramentas de filtro e ordenação estão

disponíveis para facilitar a busca de medicamentos específicos, além de ícones para

ações rápidas como editar, excluir ou visualizar detalhes. A organização dos dados

visa otimizar o processo de gestão de estoque e facilitar auditorias e inventários.

Figura 27 - Estilização CSS do Painel Administrativo

Texto

O conteúdo gerado por IA pode estar incorreto. Fonte: Autoria própria.

Esta tela fornece uma visão geral do estoque de medicamentos,

permitindo o acompanhamento em tempo real das quantidades disponíveis.

Indicadores visuais, como barras de cores e alertas automáticos, sinalizam

medicamentos com baixo estoque ou próximos da validade. O sistema utiliza

algoritmos de controle de inventário que auxiliam na tomada de decisões para reposição, evitando desperdícios ou falta de medicamentos. A interface é clara e

facilita a leitura rápida dos dados por farmacêuticos e gestores.

Figura 28 - Funcionalidades JavaScript do Painel Administrativo

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelo Gustavo Magalas

A imagem apresenta a funcionalidade de geração de relatórios

específicos sobre medicamentos, com filtros por data, categoria, status de validade, e

movimentações de entrada e saída. O sistema permite a exportação dos relatórios

em formatos como PDF e Excel, com layout otimizado para impressão. Os relatórios

podem ser utilizados para auditorias internas, envio a órgãos reguladores, ou

planejamento logístico. A tela prioriza a clareza e a navegabilidade, permitindo que o

usuário configure os relatórios com poucos cliques.

Figura 29 - Diagrama de Fluxo do Painel Administrativo

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Esta tela exibe todas as informações detalhadas de um medicamento

específico, acessadas a partir da listagem geral. São apresentados dados como

composição, lote, data de fabricação, validade, fabricante, e quantidade atual em

estoque. A interface também mostra o histórico de movimentações do medicamento,

incluindo entradas e saídas. Esse recurso é crucial para rastreabilidade e controle de

qualidade. Ícones visuais ajudam na rápida interpretação dos dados, e botões de

ação permitem a exportação dessas informações para relatórios externos.

Figura 30 - Integração com Banco de Dados

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborado pelo Gustavo Magalas

A imagem mostra a interface que gerencia a exclusão de medicamentos

do sistema. Para evitar perdas acidentais de dados, o sistema exige uma

confirmação dupla antes da remoção definitiva. Um alerta visual e uma descrição do

medicamento são apresentados, além de opções para arquivamento alternativo (soft

delete). A exclusão definitiva só é permitida para usuários com permissões

administrativas. A interface reforça boas práticas de segurança da informação,

prevenindo ações irreversíveis sem a devida autorização.

## 4.5 Script.js

O JSON (JavaScript Object Notation) é uma forma de organizar dados de maneira simples e estruturada, muito usada para comunicação entre sistemas. No JavaScript, você pode trabalhar com JSON para enviar ou receber dados, e existem duas operações principais para isso: converter um objeto JavaScript em JSON e converter JSON de volta para um objeto JavaScript. De acordo com Flanagan (2020), "JSON é um formato leve e fácil de usar para a troca de dados, permitindo a conversão entre objetos JavaScript e seu formato textual, facilitando a comunicação entre sistemas diferentes".

Figura 31 - Script JavaScript para Interações e Controle de Slideshow na Página Inicial

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

O código JavaScript implementa interações básicas na página inicial, incluindo a exibição de um alerta ao clicar no botão do carrinho, o redirecionamento para a página de login ao clicar no botão de cadastro e a configuração de controles (manual e automático) para um slideshow de imagens.

## 4.6 Js.jhson

JSON (JavaScript Object Notation) é uma forma de organizar dados de maneira que eles fiquem fáceis de ler e de entender, tanto para pessoas quanto para computadores. Ele é muito usado na comunicação entre sistemas e aplicações, especialmente na web. Imagine que você quer passar informações — como nome, idade e cidade de uma pessoa — de um lugar para outro, por exemplo, entre um site e um banco de dados. O JSON (que significa "Notação de Objetos JavaScript") organiza essas informações de forma estruturada. Em vez de simplesmente enviar uma string de texto aleatória, o JSON apresenta os dados de forma clara e padronizada. Segundo Flanagan (2020), JSON é um formato leve de troca de dados, de fácil leitura e escrita, amplamente utilizado para a comunicação entre cliente e servidor em aplicações web modernas.

Figura 32 - Exemplo de Estrutura JSON para Dados de Login

Texto

O conteúdo gerado por IA pode estar incorreto.

Fonte: Autoria própria.

Figura 33 - Explicação Simples de Código JSON de Decoração de Aniversário

Fonte: Autoria própria.

Esse código é uma lista com duas decorações de aniversário, escrita de um jeito que os computadores entendem (chamado JSON). A primeira decoração tem o tema do Lewis Hamilton, é de tamanho pequeno, tem uma imagem associada e está disponível para uso. A segunda tem o tema do Snoop Dogg, é de tamanho médio, também tem uma imagem, mas está marcada como indisponível. Cada conjunto de informações descreve o tipo de festa, o tema, o tamanho, o caminho da imagem e se a decoração está ou não disponível.

# **5 Conclusão**

A organização de eventos sociais, corporativos e culturais exige planejamento detalhado, logística eficiente e acesso a fornecedores de confiança. Diante da crescente demanda por soluções digitais que simplifiquem esse processo, surgiu a proposta do projeto *Decor Puffy*: uma plataforma web voltada à locação de itens e serviços de decoração para eventos.

O projeto tem como objetivo central oferecer um ambiente virtual onde organizadores e produtores possam encontrar, comparar e contratar fornecedores especializados de maneira prática e segura. A ideia é integrar, em um único sistema, funcionalidades como catálogo de serviços, solicitação de orçamentos, agendamento, acompanhamento de pedidos e suporte técnico, otimizando o tempo e a gestão dos eventos.

A plataforma foi desenvolvida com base em uma metodologia prática, envolvendo etapas de pesquisa, análise de requisitos, design de interface, codificação e testes. A equipe utilizou tecnologias como HTML, CSS, JavaScript, PHP e SQL, além de ferramentas de prototipagem como o Figma, para criar um sistema funcional, responsivo e acessível.

Ao propor uma solução digital inovadora, o *Decor Puffy* busca contribuir significativamente para o mercado de eventos, promovendo uma experiência mais eficiente e integrada para organizadores e fornecedores.

**REFERÊNCIAS**

Atom Cenografia: <https://www.atomcenografia.com.br/>. Acesso em: 06 Maio, 2025.

Mc93: <https://www.mc93.com.br/> . Acesso em : 06 Maio, 2025.

GetNinjas: <https://www.getninjas.com.br/>. Acesso em 06 Maio, 2025.

Carambola locações: <https://www.carambolalocacoes.com.br/>. Acesso em 06 Maio, 2025.

KRUG, Steve. ***Não me faça pensar: uma abordagem de bom senso à usabilidade na web***. Rio de Janeiro: Alta Books, 2014.

SOMMERVILLE, Ian**. *Engenharia de software***. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

FLANAGAN, David**. *JavaScript: o guia definitivo***. 7. ed. Rio de Janeiro: Alta Books, 2020.

SILVA, João; SANTOS, Maria. ***Gestão de eventos e a digitalização de processos***. São Paulo: Editora Eventos, 2018.

FLANAGAN, David. ***JavaScript: o guia definitivo***. 7. ed. Rio de Janeiro: Alta Books, 2020.

MCLOUD, Scott. ***Design de interfaces: como criar experiências digitais eficientes***. São Paulo: Novatec, 2019.

PRESSMAN, Roger S. ***Engenharia de Software***. 8. ed. São Paulo: McGraw-Hill, 2016.

BORGES, Vinícius; SOUZA, André**. *Desenvolvimento de APIs RESTful com Node.js*.** 2. ed. São Paulo: Novatec, 2020.

ANDERSON, David J. ***Kanban: Sucesso evolutivo com métodos ágeis***. São Paulo: Editora Bookman, 2010.

KERZNER, Harold. ***Gestão de Projetos: As melhores práticas***. 3. ed. Porto Alegre: Bookman, 2017.