

ESCOLA SENAI “A. JACOB LAFER”
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

JULIA ZANIRATO ARAUJO
NATALIA CAROLINE LOPEZ BEZERRA
SARAH VEIGA DANTAS
RAFAEL ARAUJO SANTOS
RAYANNE PICINI DA SILVA

PROJETO DE SISTEMA DE CADASTRO APRESENTADO A
CERVEJARIA DOGMA

SANTO ANDRÉ
2024

RESUMO

“A Cervejaria Dogma é uma marca que atende todos os apreciadores de cerveja, estimulando e incentivando novos paladares e hábitos de consumo. Seus produtos variados oferecem uma gama de estilos que incluem IPA, stout e sabores exóticos como manga, tangerina e chocolate. Fundamentada nos princípios da inovação e qualidade, a Dogma verifica detalhadamente cada lote, assegurando a excelência e originalidade de suas cervejas. Com uma abordagem focada na experimentação e na criação de novos sabores, a empresa contribui para um consumo mais responsável e consciente. A medida que a Dogma cresce, enfrenta desafios crescentes na gestão de produção e distribuição. Atualmente, esses processos são gerenciados de forma terceirizada, com uma produção de 15 mil litros por mês e vendas em mais de mil pontos no país”. No entanto, a administração reconhece a necessidade de modernização para acompanhar a expansão e melhorar a eficiência operacional. Para otimizar a gestão e aumentar a eficiência dos processos internos, a Dogma decidiu investir em soluções digitais. De acordo com a McKinsey & Company, a digitalização pode melhorar a eficiência operacional de uma empresa em até 30%, permitindo uma melhor alocação de recursos e menores custos operacionais. Além disso, a Harvard Business Review destaca que a adoção de sistemas de gestão digital pode melhorar significativamente a precisão dos dados e reduzir erros humanos. A implementação de soluções digitais é fundamental para que a Dogma acompanhe seu crescimento e permaneça competitiva no mercado. Com a modernização, a Dogma não só aprimorará seus processos internos, mas também continuará a entregar produtos de alta qualidade para seus consumidores, mantendo seu compromisso com a inovação e excelência no mundo das cervejas artesanais.

Palavras-chave: cerveja; Dogma; inovação; consumidores;

SUMMARY

“Dogma Brewery is a brand that caters to all beer lovers, stimulating and encouraging new tastes and consumption habits. Its varied products offer a range of styles that include IPA, stout and exotic flavors such as mango, tangerine and chocolate. Based on the principles of innovation and quality, Dogma checks each batch in detail, ensuring the excellence and originality of its beers. With an approach focused on experimentation and the creation of new flavors, the company contributes to more responsible and conscious consumption. As Dogma grows, it faces increasing challenges in production and distribution management. Currently, these processes are managed by outsourced companies, with a production of 15 thousand liters per month and sales in more than a thousand points in the country”. However, management recognizes the need for modernization to keep up with expansion and improve operational efficiency. To optimize management and increase the efficiency of internal processes, Dogma decided to invest in digital solutions. According to McKinsey & Company, digitalization can improve a company's operational efficiency by up to 30%, allowing for better resource allocation and lower operating costs. In addition, Harvard Business Review highlights that the adoption of digital management systems can significantly improve data accuracy and reduce human error. The implementation of digital solutions is essential for Dogma to keep up with its growth and remain competitive in the market. With modernization, Dogma will not only improve its internal processes, but will also continue to deliver high-quality products to its consumers, maintaining its commitment to innovation and excellence in the world of craft beers.

Keywords: beer; Dogma; innovation; consumers;

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	JUSTIFICATIVA	11
1.2	PROBLEMA.....	11
1.3	OBJETIVOS.....	12
2	EMBASAMENTO TEÓRICO	13
2.1	SPRINT 1.....	13
2.1.1	Figma.....	13
2.1.2	HTML	13
2.1.3	CSS	14
2.1.4	Responsividade	14
2.2	SPRINT 2.....	15
2.2.1	Banco de Dados	15
2.2.2	MySQL.....	16
2.2.3	SQL.....	16
2.3	SPRINT 3.....	17
2.3.1	PHP	17
2.4	VIABILIDADE TÉCNICA	18
3	REQUISITOS.....	19
3.1	REQUISITOS FUNCIONAIS.....	19
3.2	REQUISITOS NÃO FUNCIONAIS	20
4	DESENVOLVIMENTO	21
4.1	SPRINT	21
4.2	KANBAN	21
4.3	PRODUÇÃO DA SPRINT 1	22
4.4	SOLUÇÃO INICIAL	22
4.4.1	Protótipo	23
5	ORGANIZAÇÃO	26

6	FLUXOGRAMA DO CÓDIGO	27
6.1	CÓDIGO	28
7	PRODUÇÃO DA SPRINT 2	31
7.1	DICIONÁRIO DE DADOS.....	31
7.2	MODELOS DE BANCO DE DADOS	32
7.2.1	Modelo Conceitual.....	32
7.2.2	Modelo Lógico	33
7.2.3	Modelo Físico	34
8	PRODUÇÃO DA PRINT 3.....	36
8.1	CÓDIGOS.....	36
9	CONSIDERAÇÕES FINAIS	38
	REFERÊNCIAS.....	39
	APÊNDICE A	41
	APÊNDICE B	44
	APÊNDICE C	48
	APÊNDICE D	74
	APÊNDICE E.....	76
	APÊNDICE F	81

1 INTRODUÇÃO

Esse trabalho tem o intuito de solucionar o problema de nosso cliente Cervejaria Dogma, que está tendo problemas em acompanhar o crescimento do mercado nessa era digital. Então, para solucionar o problema, utilizamos diversas ferramentas para a criação de uma web site que ajudará a comunicação entre a Cervejaria Dogma, seus fornecedores e clientes. Portanto, para que chegássemos em um projeto final que atendesse tudo que a Cervejaria queria, separamos em 3 sprints, visando melhoras entre elas e modificações caso o cliente desejasse. Nossa equipe se atentou em pesquisar sobre os programas que iríamos utilizar e, pensando sobre os clientes da Cervejaria Dogma, realizamos pesquisas sobre responsividade para atender os diferentes tipos de dispositivos que acessarão a página da empresa. Separamos as tarefas entre a equipe para que o projeto fosse concluído na data, utilizando a metodologia ágil Scrum e Kanban. Na primeira Sprint entregaremos o protótipo do site, o código e a documentação de tudo que fizemos.

Esse trabalho tem intuito didático, mesmo que a Cervejaria Dogma exista. Esse projeto tem o objetivo de avaliar o conhecimento dos alunos e para que tenhamos experiência em projetos como esse, para nos preparar para o mercado de trabalho.

1.1 JUSTIFICATIVA

Conforme apresentado anteriormente, baseada na necessidade de otimização e de modernização dos processos internos da empresa. Com o aumento crescente na demanda torna-se crucial acrescentar uma solução digital para uma garantia de uma administração eficiente. Essas melhorias são fundamentais para acompanhar o crescimento da empresa e garantir sua competitividade em um mercado cada vez mais dinâmico e digitalizado. Garantindo maior agilidade na tomada de decisões e no acompanhamento dos estoques, preços e fornecedores

1.2 PROBLEMA

Nós fomos contratados como desenvolvedores para criar uma aplicação web que permita à cervejaria Dogma gerenciar seus fornecedores e produtos de forma

mais eficiente. A aplicação deve proporcionar uma interface intuitiva para que os funcionários possam cadastrar, atualizar, visualizar e excluir informações sobre fornecedores e produtos.

1.3 OBJETIVOS

Nosso objetivo é fornecer uma solução que agrade tanto a Cervejaria Dogma, quanto seus clientes e fornecedores. Temos como objetivo entregar o projeto na data marcada e atender a todos os requisitos que o cliente nos pediu, visando o crescimento de nossa equipe de desenvolvedores para solucionar todos os problemas que aparecerem no decorrer do projeto.

2 EMBASAMENTO TEÓRICO

Aqui deixaremos especificado tudo o que iremos utilizar de base para a criação do nosso projeto, sejam sites, ferramentas, linguagens, sistemas etc. Será separado por sprints os conteúdos utilizados, para manter a organização.

2.1 SPRINT 1

Para iniciarmos o projeto, pesquisamos sobre as tecnologias que seriam utilizadas, para aprofundar no assunto e saber mais sobre o que deveríamos atender. Primeiramente, pesquisamos sobre a ferramenta que será usada para a criação do protótipo de alta fidelidade, o Figma. Após o entendimento da ferramenta, passamos para a pesquisa das linguagens utilizadas para a codificação inicial do site, o HTML e o CSS. Finalizamos as pesquisas de embasamento procurando conhecer um pouco mais sobre responsividade e sua importância.

2.1.1 Figma

Segundo Matheus Villain e Maria Silveira (2024), o figma é uma ferramenta gratuita de design versátil, uma plataforma colaborativa para construção de interfaces, protótipos e wireframes. É um dos caminhos mais acessíveis utilizados na criação de telas para produtos como aplicativos, sites ou softwares.

Ele é disponibilizado via web, com acesso através de qualquer navegador e traz todas as funcionalidades de um aplicativo sob medida para o sistema operacional em que será executado.

2.1.2 HTML

De acordo com Andrei L. (2023), a Linguagem de Marcação de Hipertexto (HTML) é uma linguagem de computador utilizada para a criação de diversas páginas da Internet e aplicativos online. Os marcadores, chamados de tags, indicam a função de cada elemento na página, incluindo imagens, parágrafos de texto e vídeos, e suas conexões com outros elementos (links), para como serão interpretados pelo navegador. Em resumo, o HTML é responsável por fazer a estrutura de uma página web.

Mesmo que o HTML seja uma linguagem muito utilizado, não é possível criar um site responsivo e profissional utilizando apenas ele, pois serve para adicionar apenas textos e estrutura da página. Por isso, ele é utilizado junto com outras

linguagens, como por exemplo, o CSS e o JavaScript. Para a criação da página da Cervejaria Dogma, utilizamos o HTML para a estrutura e o CSS para a implementação de funcionalidades na página, como cores e espaçamentos.

2.1.3 CSS

Segundo Ariane de G. (2022) e Itamar de Haim (2024), O CSS (sigla para o termo em inglês Cascading Style Sheets) é uma linguagem para especificar como será a parte de visual apresentada para os usuários, estiliza elementos escritos em uma linguagem de marcação como HTML e desenvolve dinâmica e efeitos da página em sites. Pode ser feito alterações rápidas de layout e é possível separar formatos de conteúdos, bem como definir como serão exibidos em cada página.

2.1.4 Responsividade

De acordo o site Webshare (2022), a responsividade de um sistema se refere à capacidade de um site ou sistema se comportar e funcionar em diferentes dispositivos e tamanhos de tela. A responsividade visa um sistema capaz de oferecer uma experiência de qualidade em todos os dispositivos.

Um site responsivo é aquele que consegue de adaptar as dimensões de qualquer dispositivo, não mudando o layout, mas sim como os elementos de distribuem e o tamanho dos elementos. Essa responsividade acontece para que o usuário final não tenha sua experiência prejudicada por causa de elementos desordenados.

É muito importante que sites respondam à responsividade, como é citado no site Five Performance Digital, pois mecanismos de buscas, como o Google, priorizam sites que possam se adaptar, havendo penalidade dos sites que não tenham a responsividade em resultados de busca.

Para produzir um sistema responsivo, é utilizado técnicas como o CSS e o media queries. Enquanto o CSS fica responsável por definir a modificação de elementos específicos para diferentes tamanhos de tela, o media queries é responsável pela aplicação desses estilos de acordo com as características do dispositivo.

2.2 SPRINT 2

Na Sprint 2, focamos na criação do banco de dados. Até o final da Sprint 2, tudo o que fizemos foi a criação de um banco de dados para a cervejaria que, mais tarde, seria conectado com o site produzido na primeira Sprint. Para concluirmos a entrega com sucesso, pesquisamos muito sobre banco de dados, a linguagem que utilizamos e o sistema de gerenciamento utilizado

2.2.1 Banco de Dados

De acordo com Danielli Oliveira (2024), um banco de dados é um sistema de armazenamento de informações que permite a coleta, o armazenamento, a recuperação e a manipulação de dados de maneira estruturada e eficiente. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados. Nos dias atuais, existem vários tipos de banco de dados, incluindo relacionais, NoSQL e outros, cada um com suas próprias características e usos específicos.

Um banco de dados relacional é um conjunto de informações que organiza dados em relações predefinidas, em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Facilitando a visualização e a compreensão de como diferentes estruturas de dados se relacionam. Os relacionamentos são uma conexão lógica entre diferentes tabelas, que se estabelecem com base na interação entre elas.

O banco de dados não relacional é um banco de dados que possibilita a flexibilidade na hora de armazenar os dados, já que não se limita a tabelas com linhas e colunas, como o banco de dados relacional. Em vez disso, os bancos de dados não relacionais usam um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados que está sendo armazenado, por exemplo: possibilita que os dados sejam armazenados como chave/valor simples; como documentos JSON ou como um gráfico que consiste em bordas e vértices.

2.2.2 MySQL

Conforme o trabalho de Pedro Pisa (2012), o MySQL é um sistema de gerenciamento de bancos de dados relacional de código aberto, usado na maioria das aplicações gratuitas para gerir suas bases de dados. O MySQL utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados. Ele também é parte essencial da amplamente utilizada pilha de aplicativos da Web Linux-Apache-MySQL-PHP/Perl/Python (LAMP), que serve de base para muitos aplicativos, sites e serviços conhecidos.

O SGBD MySQL oferece diversas vantagens para o gerenciamento de bancos de dados, sendo amplamente reconhecido por sua alta performance, confiabilidade e garantia, a flexibilidade e a facilidade de uso.

O MySQL cria um banco de dados para armazenamento e manipulação de dados, definindo a relação de cada tabela. Clientes podem fazer solicitações digitando comandos SQL específicos no MySQL. A aplicação do servidor responde com a informação solicitada fazendo aparecer no cliente.

2.2.3 SQL

Segundo o site AWS da Amazon, o SQL é uma linguagem de programação usada para armazenar e processar informações em bancos de dados relacionais. Os bancos de dados relacionais organizam informações em tabelas com linhas e colunas, onde dentro dessas colunas existem dados que se relacionam. O SQL permite manipular, pesquisar, remover, atualizar e otimizar performance do banco de dados.

O SQL é uma linguagem de consulta popular usada em diferentes aplicações. Analistas e desenvolvedores usam o SQL porque ele se integra bem com linguagens de programação. Por exemplo, é possível combinar SQL com Java para criar aplicações de processamento de dados de alta performance com sistemas como Oracle e MS SQL. SQL é fácil de aprender, pois usa palavras-chave comuns em inglês.

Alguns dos comandos utilizados no SQL são de criação de tabelas, colunas e linhas (CREATE), o de alteração de dados e colunas (UPDATE e ALTER), os

comandos de exclusão de tabelas e banco (DROP) e o de registro (DELETE), o de inserção de dados (INSERT) e o de consulta e visualização de dados (SELECT).

A Linguagem de consulta estruturada (SQL) é um padrão para criação e manipulação de bancos de dados, enquanto o MySQL é um programa de banco de dados que usa consultas SQL. Comandos SQL seguem padrões internacionais, enquanto o MySQL é atualizado e melhorado regularmente.

Para o desenvolvimento da Sprint 2 do projeto Dogma, será utilizada as linguagens já citadas na Sprint 1, com acréscimo das linguagens PHP e SQL para a criação de banco de dados no sistema MySQL, assim conseguiremos que os dados de fornecedores e produtos estejam armazenados de forma organizada e que possam ser alterados, excluídos e gerenciados, se necessário pelos usuários fornecedores e pela empresa.

2.3 SPRINT 3

Na Sprint 3, nosso objetivo era a conexão do site produzido na Sprint 1 com o banco de dados feito na Sprint 2. Portanto, tivemos que utilizar a linguagem PHP. Para que não ocorresse erros, pesquisamos sobre a linguagem e registramos no arquivo o que achamos de maior interesse.

2.3.1 PHP

Segundo Priscila Polidoro (2024), o PHP (Hypertext Preprocessor) é uma linguagem de programação criada originalmente para a criação de sites e aplicações Web dinâmicas para a internet. Foi uma das primeiras linguagens de inserção em documentos HTML, e a partir do código que é interpretado no servidor, gera uma página web que será visualizada pelo usuário.

O PHP é uma das linguagens de programação mais populares atualmente, originando-se de códigos CGI criados por Rasmus Lerdorf para seu site pessoal. Com o tempo, esse aglomerado de códigos foi transformado em uma linguagem de programação, tornando-se fundamental para a criação de sites e aplicações na internet. Além de blogs e lojas virtuais, o PHP evoluiu para incluir scripts de linha de comando e aplicações para desktops. Por ser fácil de aprender e oferecer bom

desempenho, o PHP se tornou uma ferramenta indispensável para os desenvolvedores.

Principais características do PHP incluem velocidade e robustez, desenvolvimento orientado a objetos, independência de aplicação, a sintaxe é similar a C, C++ e JavaScript, é open source e pode ser utilizada no lado do servidor para respostas HTML.

As vantagens dessa linguagem incluem facilidade de uso, criar e usar scripts sem uso de navegador, linguagem de fácil compreensão, baixo custo e sua integração com diversas bases de dados, como MySQL, Oracle, SQL Server e entre outros.

3.4 VIABILIDADE TÉCNICA

O desenvolvimento desta aplicação web é viável de acordo com as necessidades apresentadas pela Cervejaria Dogma. A escolha das tecnologias (HTML/CSS) para o desenvolvimento baseia-se em eficiência e segurança, permitindo à Dogma potencializar seus processos internos, melhorar a precisão dos dados e acompanhar seu crescimento. O CSS desempenha um papel essencial na estilização do site, proporcionando uma interface mais atraente e funcional, enquanto o HTML oferece uma estrutura sólida e acessível a todos os navegadores.

3 REQUISITOS

Requisitos consiste em uma propriedade ou comportamento que um produto deve ter

3.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais são capacidades concretas e operações em que o software deve atingir a solução dos problemas dos usuários, materializando estes conceitos em forma de funções no novo sistema. O nosso sistema terá como requisitos funcionais:

- Cadastro de Fornecedores:

Nome,

CNPJ,

Endereço,

Telefone,

E-mail,

Observações.

- Cadastro de Produtos:

Nome do Produto,

Código, Descrição,

Quantidade em Estoque,

Preço Unitário,

Fornecedor (associado a um fornecedor).

Funcionalidades Adicionais:

- Pesquisa de fornecedores e produtos.
- Edição e exclusão de registros.
- Sistema de Login para garantir o acesso restrito.

3.2 REQUISITOS NÃO FUNCIONAIS

Enquanto os requisitos funcionais estão focados no que será realizado, os não funcionais descrevem como serão realizados. Eles são a forma como o software tornará realidade o que está sendo estruturado. No caso do nosso projeto são:

- Usabilidade: Interface intuitiva, fácil de usar para todos os níveis de usuários.
- Desempenho: Tempos de resposta adequados para todas as funcionalidades (carga de páginas e consultas em <2s).
- Segurança: Autenticação com login, validação de dados (CNPJ, e-mails).
- Responsividade: Interface adaptada para diferentes dispositivos (desktop, tablet, mobile).
- Velocidade: Velocidade de resposta do site de acordo com a Internet do usuário.

4 DESENVOLVIMENTO

Para o desenvolvimento do projeto de forma ágil, utilizamos a metodologia Scrum, para entregas de etapas do projeto e o Kanban, para divisão de tarefas e poder acompanhar como o projeto está andando, ou se precisamos focar em uma tarefa em específico. Essas metodologias ajudaram a tornar o projeto mais organizado, fazer com que todos os desenvolvedores da equipe participem ativamente e saibam o que seus colegas estão produzindo. Explicaremos um pouco sobre essas metodologias antes de falar o que fizemos do projeto.

4.1 SPRINT

As sprints são ciclos em que a equipe trabalha para desenvolver uma quantidade determinada de tarefas do projeto, elas fazem parte da metodologia scrum, usando esse método ágil acreditamos que teremos um melhor funcionamento. Nesse projeto iremos realizar três sprints com o objetivo de progresso em cada uma delas para entregar um melhor resultado.

4.2 KANBAN

Como estratégia de organização das tarefas, utilizamos o sistema Kanban.

Junto do Scrum, o Kanban é uma metodologia ágil usada para a divisão de tarefas e a otimização do trabalho. De acordo com G. Ferreira (2022), a origem do Kanban surge na montadora japonesa Toyota, com o intuito de controlar a produção de peças e o estoque de materiais, para que as peças não excedessem ou faltassem, eles utilizavam cartões para marcar. O método foi adaptado por David Anderson para detalhar quando, quanto e o que estava sendo desenvolvido.

O funcionamento do Kanban se dá por meio de colunas que mostrem o andamento da tarefa, por exemplo, "a fazer", "em andamento" e "feito", que podem ser modificadas de acordo com a necessidade das tarefas. Pode ser usado post-its, cartões ou fitas para diferenciar as tarefas, com as cores representando a prioridade delas, também especificando o responsável por concluir e o tipo de tarefa. Então, com os cartões feitos, move-os pelas colunas de acordo com o andamento delas.

Há muitos benefícios em utilizar o Kanban, como maior produtividade e a priorização de tarefas mais importantes, trazendo agilidade para a equipe. É um método que ajuda na conclusão de tarefas e mantém a calma da equipe, desde que as tarefas estejam bem organizadas e separadas.

O Kanban feito pela equipe na primeira Sprint será mostrado mais para a frente.

4.3 PRODUÇÃO DA SPRINT 1

Na primeira Sprint, tínhamos como objetivo realizar o protótipo do site, que faríamos na ferramenta Figma, fazer o código de algumas das páginas do site e entregar a documentação. Quando finalizado essa primeira parte proposta, apresentaremos para o nosso cliente para trazer um feedback e fazermos mudanças no protótipo.

Para tudo ocorrer no prazo desejado, fizemos uma reunião inicial para decidir o que cada desenvolvedor faria. Após finalizarmos a discussão, foi produzido um levantamento das tarefas e suas importâncias, colocadas no método Kanban para melhor visualização. Movemos as tarefas de coluna de acordo com o andamento delas.

Depois, cada um dos integrantes iniciou suas tarefas. Rayanne e Julia ficaram responsáveis em pesquisar, levantar os dados e transformá-los em um documento, que o nosso cliente e os outros membros da equipe têm acesso, esse arquivo será modificado entre as Sprints. Sarah e Natalia tiveram como tarefa a produção da tabela de cores e tipografia que serão utilizadas na página web. Assim que finalizaram essa parte, as integrantes Sarah e Natalia começaram a produção do protótipo em desktop no Figma, Rayanne ajudando no protótipo para celulares. Rafael ficou responsável pela parte de codificação do site, usando o HTML e o CSS.

Enquanto todos da equipe produziam, atualizávamos o documento com tudo o que foi feito.

4.4 SOLUÇÃO INICIAL

Nossa estratégia para realizarmos a primeira fase do projeto com êxito foi a separação de tarefas. Ao analisarmos as tarefas mais importantes, conseguimos focar nelas e finalizar sem problemas. Outra estratégia foi o trabalho em conjunto dos desenvolvedores, trabalhando em equipe para que as tarefas fossem terminadas e os obstáculos fossem resolvidos. Também com a otimização do tempo e diversas funções acontecendo ao mesmo tempo, não tivemos muitos problemas com tempo.

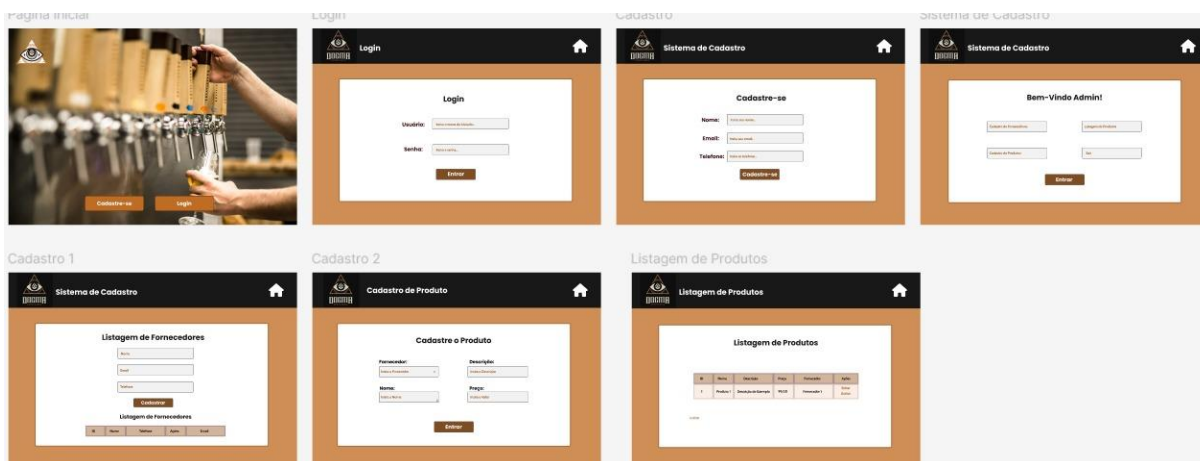
A solução foi feita em etapas, primeiro com o planejamento, depois divisão de tarefas, depois o início das produções, depois uma discussão para ver como o projeto estava andando e a finalização das tarefas da Sprint 1. Foi importante discutir o

andamento das tarefas de horas em horas, para que todos pudessem estar cientes do estavam ocorrendo e, se necessário, todos focarem para a finalização de um só serviço.

4.1.1 Protótipo

Na Figura 1, é possível observar o protótipo de alta fidelidade que foi produzido das páginas webs para desktop, feito na ferramenta digital Figma. As diferentes telas contêm a página inicial, login de usuários, tela de cadastro para fornecedores, tela para linkar com as outras páginas de cadastro, página de cadastro e de lista de fornecedores, tela de cadastro de produtos e tela de listagem de produtos. Para ver com mais detalhes os protótipos, vá para o Apêndice A.

Figura 1 – Protótipo desktop.



Fonte: Elaborado pelos autores (2024).

Na Figura 1, é possível observar o protótipo de alta fidelidade que foi produzido das páginas webs para desktop, feito na ferramenta digital Figma. As diferentes telas contêm a página inicial, login de usuários, tela de cadastro para fornecedores, tela para linkar com as outras páginas de cadastro, página de cadastro e de lista de fornecedores, tela de cadastro de produtos e tela de listagem de produtos. Para ver com mais detalhes os protótipos, vá para o Apêndice A.

Para a produção do protótipo, primeiro foi feito a tipografia e cores da página, como na Figura 2. Foi necessário especificar as cores que seriam usadas, suas variações em diferentes tipos de luminosidade, as fontes e variações, especificação do tamanho das fontes e contraste das cores com o fundo já selecionadas.

Figura 2 – Tipografia e cores.



Fonte: Elaborado pelos autores (2024).

Com as cores escolhidas, iniciamos o processo de produzir o protótipo. Dividimos entre os membros cada página que precisavam ser realizadas. Seguimos um padrão de tamanho para as barras de preenchimento, das fontes, dos títulos, de onde a logo da empresa deveria estar localizada e a coloração da página. Com o projeto do desktop finalizado, partimos para a página de mobile.

Na Figura 3, é apresentado parte do protótipo mobile, para ver todo o protótipo, vá para Apêndice B. Aplicamos a responsividade movendo as caixas de texto, tamanho dos itens e títulos, tudo para que os usuários do mobile também tenham uma ótima experiência com a página. A responsividade é importante para que a página web funcione em qualquer tipo de aparelho, para que agrade qualquer tipo de usuário e fornecedor.

Figura 3 – Protótipo mobile

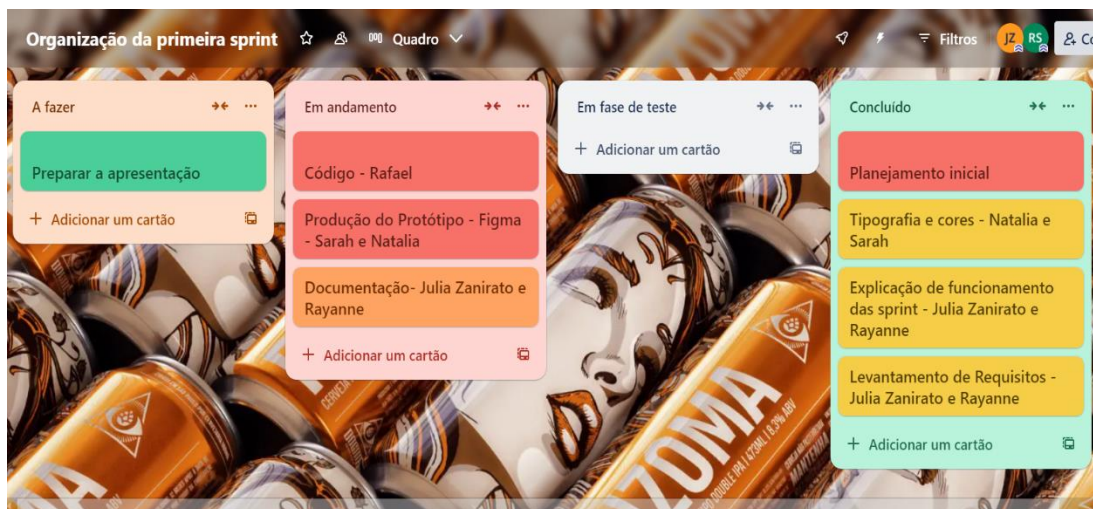
The image displays a mobile application prototype with two main screens. The top screen is the 'Login' screen, which has a dark header with a logo on the left and the word 'Login' on the right. Below the header, there is a white card with a brown border. Inside the card, the title 'Usuário:' is followed by a text input field with the placeholder 'insira o nome do usuário...'. Below this, the title 'Senha:' is followed by another text input field with the placeholder 'insira a senha...'. At the bottom of the card is a brown button labeled 'Entrar'. The bottom screen is the 'Cadastre-se' screen, which has a white background with a brown border. It features the title 'Cadastre-se' at the top. Below the title, there are three text input fields: 'Nome:' with placeholder 'insira o nome do usuário...', 'Email:' with placeholder 'insira o seu email...', and 'Telefone:' with placeholder 'insira o seu telefone...'. At the bottom of the screen is a brown button labeled 'Entrar'.

Fonte: Elaborado pelos autores (2024)

5 ORGANIZAÇÃO

Como é possível observar na Figura 4, foi organizado após a primeira discussão uma tabela, que representa por meio de cores e colunas as tarefas que devem ser realizadas na Sprint 1.

Figura 4 – Kanban



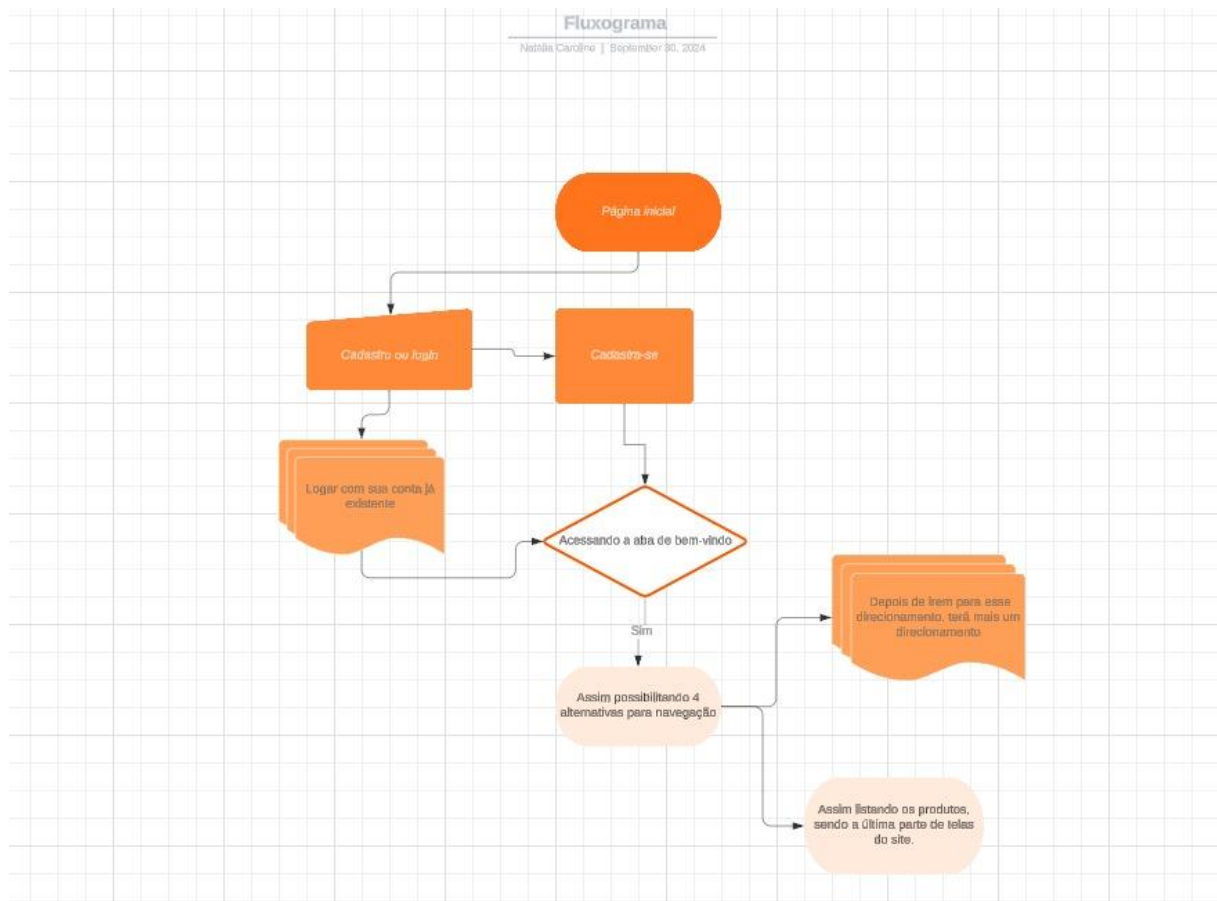
Fonte: Elaborado pela equipe (2024).

As cores das tarefas representam o nível de importância, o verde sendo tarefas que podem ser concluídas depois ou que o prazo não está próximo, o amarelo tarefas que são importantes, mas que não são prioridades, laranja são as tarefas importantes e as vermelhas são as que precisam ser concluídas primeiro, de extrema importância. Há também o nome de quem desenvolverá cada tarefa e em que fase está, representado pelas colunas “a fazer”, “em andamento”, “em fase de teste” e “concluído”.

6 FLUXOGRAMA DO CÓDIGO

Foi feito um fluxograma de qual caminho as páginas deveriam seguir, iniciando na página inicial, como é possível observar na Figura 5. Da página inicial, o fornecedor será levado para a página de cadastro ou login, escolhendo uma das opções, ele será levado para a tela de cadastro ou para a de logar com conta existente. Concluindo o cadastro, será levado para a tela de bem-vindo, onde poderá selecionar 4 opções de páginas diferentes, lista de fornecedores, lista de produtos, cadastro de produtos e cadastro de fornecedores, que depois de selecionadas, finalizam as páginas web.

Figura 5 – Fluxograma.



Fonte – Elaborado pelos autores (2024).

6.1 CÓDIGO

Ao total, para a criação de todas as páginas, foram usados 7 arquivos de HTML e 2 de CSS. Será comentado apenas as partes que a equipe achou importante apresentar. Para visualizar todo o código das páginas, vá para o **Apêndice C**.

Na Figura 6, está apresentado como foi feito o header com as tags no HTML. Esse header aparecerá na maior parte das páginas criadas para a Dogma, sendo modificado apenas os textos presentes na tag *h1*. A tag *img* é utilizada para inserir uma imagem à página, no caso da nossa, a logo da Cervejaria Dogma. A *class* que aparece será utilizada na estilização no CSS. A tag *h1* serve para escrever um título na página.

Figura 6 – Header.

```
<header>
  
  <h1>Bem-Vindo</h1>
</header>
```

Fonte – Elaborado pelos autores (2024).

A Figura 7 está apresentando a parte de estilização do header no CSS, mudando as cores, as fontes, alinhamento dos itens, posição de header, entre outras coisas. Na figura também é mostrado as configurações iniciais das páginas, como margem, cor de fundo, altura, etc.

Figura 7 – Estilização do header.

```
/* Garantindo que o body ocupe toda a tela e centralize o conteúdo */
body,
html {
  height: 100%;
  margin: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: #CF8E55;
  font-family: 'Poppins';
}

/* configuração geral do cabeçalho */
header {
  background-color: black;
  width: 100%;
  color: white;
  display: flex;
  align-items: center;
  padding: 10px;
  box-sizing: border-box;
  position: absolute;
  top: 0;
}
```

Fonte – Elaborado pelos autores (2024).

Com a finalização da estilização do header, esse cabeçalho ficará como a Figura 8.

Figura 8 – Header após estilização.



Fonte – Elaborado pelos autores (2024).

Na Figura 9 é representado a parte central de algumas páginas webs. Esses códigos irão se repetir algumas vezes, pois é a parte responsável pelas barras que levarão para outras páginas, seja de produtos ou de fornecedores. A tag *a href* é utilizada para linkar uma página com outras, como é o exemplo dessa página, que a tela de bem-vindo está conectada com as outras páginas de cadastro e listagem. Ou seja, após uma das barras ser selecionada, o usuário será levado para outra aba, com a página que ele escolheu.

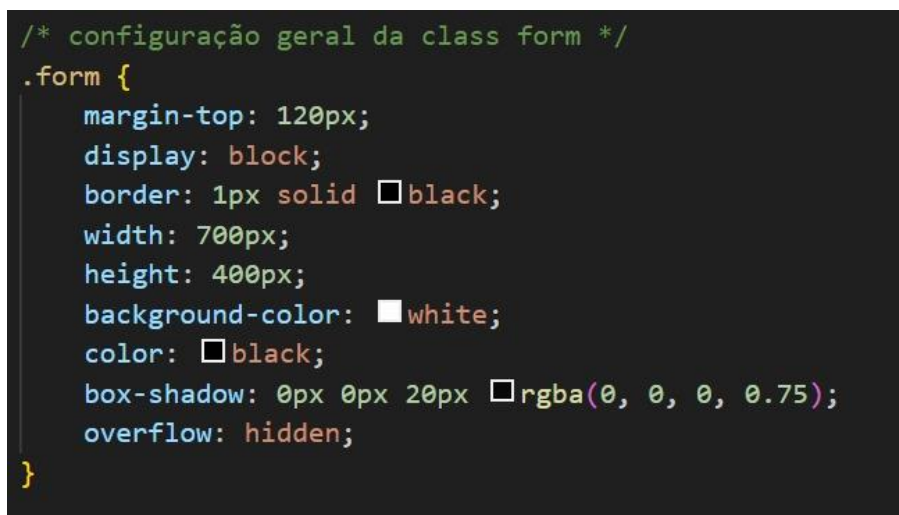
Figura 9 – Conteúdo central.



Fonte – Elaborado pelos autores (2024).

Na figura 10, é possível observar a estilização no CSS do quadro branco que estará atrás das barras de link. Foi modificado o tamanho do retângulo, sua margem, sua borda, sua cor e foi adicionado uma sombra no quadro.

Figura 10 – Estilização do quadro branco central.



Fonte – Elaborado pelos autores (2024).

Ao finalizar a estilização do quadro, partimos para a modificação dos itens que estavam dentro do quadro, ou seja, as barras de link e o botão de entrar. Foi modificado cores, posições, altura da fonte, alinhamento, entre outros itens. A figura 11 contém toda a parte de modificação e estilização desses itens do quadro.

Figura 11 – Estilização dos itens do quadro.

```
/* class dos botões da landing page bem-vindo */
.btn-bem-vindo {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
  align-items: center;
  width: 100%;
  box-sizing: border-box;
  gap: 15px;
  padding: 40px;
  transition: background-color 0.3 ease;
  cursor: pointer;
  border-radius: 5px;
}

/* mexendo no botao da landing page bem-vindo */
.btn-bem-vindo button {
  background-color: white;
  display: flex;
  color: #A66B38;
  font-weight: normal;
  width: 250px;
  padding: 10px;
  justify-content: center flex-start;
  align-items: center;
}

/* hover do btn-bem-vindo */
.btn-bem-vindo button:hover {
  background-color: #CFBE55;
  color: white;
}
```

Fonte – Elaborado pelos autores (2024)

Após finalizar a estilização no CSS, a página ficou como apresentada na Figura 12. Foi estilizada com as cores e fontes presentes na parte de tipografia e cores.

Figura 12 – Conteúdo central.



Fonte – Elaborado pelos autores (2024).

7 PRODUÇÃO DA SPRINT 2

Como objetivo da nossa Sprint 2, tínhamos que entregar à Cervejaria Dogma as páginas prometidas e junto à um banco de dados, que possibilitasse o cadastro de fornecedores e produtos. Os dados que seriam cadastrados teriam que ficar armazenados nesse banco de dados, e poderiam ser alterados, excluídos, atualizados e visualizados. Teríamos que fazer os modelos de banco de dados conceitual, lógico e físico, além da produção do dicionário de dados, todos esses projetos para que assim produzíssemos de fato o banco de dados para ser implementado na página da Cervejaria.

7.1 DICIONÁRIO DE DADOS

Para entender como o banco de dados iria funcionar, foi iniciado a criação de um dicionário de dados. Dicionário de dados são documentos que centralizam informações sobre o conjunto de dados, que são importantes para que seja possível entender como e onde serão adicionadas as informações nos bancos de dados.

No caso da Cervejaria Dogma, foram necessários dois dicionários, uma tabela que organizasse os dados de fornecedores e outra que organizasse as de produtos. Nesses dicionários estão presentes as colunas das tabelas, que nesse caso serão representados em linhas, o tipo de dado que essa coluna irá receber (texto, número ou outro), a quantidade de caracteres que poderão ser adicionados e se será obrigatório o campo da tabela ser preenchido antes de ele ir para o banco de dados. Ao finalizar e adicionar as informações, o Dicionário de Dados está representado pelas Figuras 13 e 14.

Figura 13 – Dicionário de Dados de Produtos.

PRODUTOS			
Nome do Campo	Tipo de Dado	Tamanho	Obrigatório
Código	Numérico	10	SIM
Nome	Texto	100	SIM
Observação	Texto	500	SIM
Quantidade_Estoque	Numérico	3	SIM
Preço Unitário	Decimal	5	SIM

Fonte – Elaborado pelos autores (2024).

Figura 14 – Dicionário de Dados de Fornecedores.

FORNECEDORES			
Nome do Campo	Tipo de Dado	Tamanho	Obrigatório
Nome	Texto	100	SIM
CNPJ	Númerico	18	SIM
Endereço	Texto	70	SIM
Telefone	Númerico	11	SIM
E-mail	Texto	50	NÃO
Observações	Texto	500	SIM

Fonte – Elaborado pelos autores (2024).

7.2 MODELOS DE BANCO DE DADOS

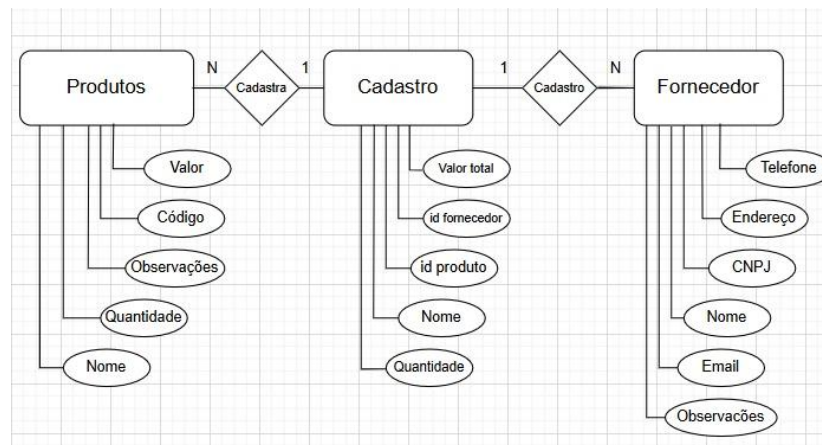
Foram modelados 3 tipos de Banco de Dados: Conceitual, Lógico e Físico. Esses modelos são dados organizados em um estilo semelhante de uma árvore, e esses dados se conectam entre si por meio de links.

Foi feito esses 3 modelos para o nosso projeto. Iniciaremos falando do modelo conceitual.

7.2.1 Modelo Conceitual

O primeiro modelo feito foi o conceito. O modelo conceitual tem o objetivo de fazer a modelagem inicial para organizar conceitos e regras, sendo o modelo mais simples dos 3, pois é um modelo utilizado para a organização de dados. Apresentado na Figura 15, temos o modelo criado, onde os retângulos representam as tabelas do banco de dados, as figuras ovais representam as colunas que estão vinculadas às suas respectivas tabelas e os losangos representam o relacionamento entre as páginas.

Figura 15 – Modelo de Banco de Dados Conceitual.

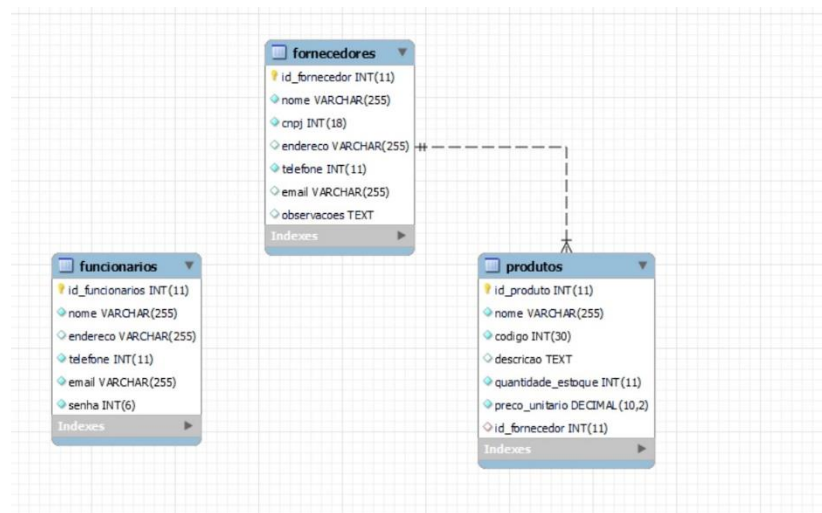


Fonte – Elaborado pelos autores (2024).

7.2.2 Modelo Lógico

A figura 16 contém o modelo lógico do projeto da Cervejaria Dogma. Nesse modelo, as tabelas já estão mais visíveis, e as colunas da tabela também. Os valores de cada coluna estão presentes, e o tipo de valor está escrito na frente do nome das colunas.

Figura 16 – Modelo de Banco de Dados Lógico.



Fonte – Elaborado pelos autores (2024).

O modelo lógico de dados é uma versão mais refinada do modelo conceitual. Representando diagramaticamente restrições de dados, nomes de entidades e relacionamentos para implementação de forma independente da plataforma.

Ele serve como uma representação abstrata dos requisitos de dados da organização, concentrando-se em entidades, atributos e relacionamentos sem se

preocupar com detalhes de implementação. Ele é projetado para capturar as necessidades e regras de negócios de forma compreensível e independente de tecnologia.

- Entidades (tabelas que existirão no banco de dados);
- Relacionamentos (como essas tabelas estão conectadas);
- Atributos (colunas dessas tabelas).

7.2.3 Modelo Físico

A figura 17 apresenta o modelo físico do projeto. Em resumo, o modelo físico é a programação para a criação do banco de dados. Explicaremos como funcionou a programação do banco de dados com a linguagem SQL.

Figura 17 – Modelo de Banco de Dados Físico.

```
DDL for sprint.produtos

1 CREATE TABLE `produtos` (
2   `id_produto` int(11) NOT NULL AUTO_INCREMENT,
3   `nome` varchar(255) NOT NULL,
4   `codigo` int(30) NOT NULL,
5   `descricao` text DEFAULT NULL,
6   `quantidade_estoque` int(11) NOT NULL,
7   `preco_unitario` decimal(10,2) NOT NULL,
8   `id_fornecedor` int(11) DEFAULT NULL,
9   PRIMARY KEY (`id_produto`),
10  KEY `id_fornecedor` (`id_fornecedor`),
11  CONSTRAINT `produtos_ibfk_1` FOREIGN KEY (`id_fornecedor`) REFERENCES `fornecedores` (`id_fornecedor`)
12 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci
```

Fonte – Elaborado pelos autores (2024).

Primeiramente será feito a criação de um banco de dados utilizando o "CREATE DATABASE" e em seguida o comando "USE" para selecioná-lo como um banco ativo, onde os próximos comandos serão realizados.

Após a criação e configuração do banco de dados, para criar as tabelas usamos o comando "CREATE TABLE".

Comandos usados:

INT PRIMARY KEY: Recurso aplicado a um campo (id_fornecedor) para que ele possa gerar e fornecer automaticamente, um valor exclusivo para cada registro que for inserido.

VARCHAR: É um tipo de dado de caractere de comprimento variado.

NOT NULL: força um campo a sempre conter o mesmo valor.

TEXT: Armazena dados de texto.

DECIMAL: Projetado para valores exatos com decimal.

REFERENCES: Define qual tabela e coluna são usadas em um relacionamento de chave estrangeira.

SELECT: Busca ou seleciona informações específicas.

FROM: Lista as tabelas e quaisquer junções necessárias para a consulta no MySQL.

JOIN: Usado para combinar linhas de duas ou mais tabelas.

No final do código, foi feito um retorno na lista de produtos com seus respectivos preços e o nome do fornecedor, utilizando o JOIN para combinar as informações da tabela produtos com a tabela fornecedores. Para visualizar todo o código feito, vá para o **Apêndice D**.

8 PRODUÇÃO DA SPRINT 3

Na Sprint, os desenvolvedores tiveram como tarefa a integração do banco de dados nas páginas, utilizando o PHP. Aproveitamos para fazer alterações no site, de acordo com o feedback do cliente, algumas cores e planos de fundo foram modificadas, para deixar a página mais interessante e a cara da Cervejaria Dogma. Para visualizar as modificações feitas, vá para o **Apêndice E**.

8.1 CÓDIGOS

Nessa Sprint, tivemos que conectar as páginas com o banco de dados, utilizando a linguagem PHP para que ocorra essa integração. Será explicado algumas partes do código. Para visualizar todos os códigos produzidos e as modificações feitas no código, vá para o Apêndice F.

O primeiro código feito para essa conexão foi o conexão.php, que pode ser observado na Figura 18. Nesse código é onde informa o nome do banco de dados, o nome do server onde esse banco estará, senha e entre outras coisas. Caso a conexão dê algum erro, aparecerá uma mensagem na página, e caso ocorra com sucesso, o funcionamento da página ocorrerá sem nenhum problema.

Figura 18 – Conexão PHP.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sprint";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error){
    die("Connection failed: " . $conn->connect_error);
}

?>
```

Fonte – Elaborado pelos autores (2024).

A outra parte que achamos importante de comentar, foi uma parte do código da página de cadastro de produtos. Na Figura 19, retrata da inserção de dados, da localização dos dados, do redirecionamento, retrata o nome das variáveis e se elas vão ser inseridas automaticamente ou pelo usuário.

Figura 19 – Cadastro de produtos PHP.

```

7 // Verifica se o formulário foi enviado
8 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
9     $id = $_POST['id_produto'] ?? '';
10    $fornecedor = $_POST['id_fornecedor'];
11    $nome = $_POST['nome'];
12    $descricao = $_POST['descricao'];
13    $preco = str_replace(',', '.', $_POST['preco_unitario']); // Converte vírgula para ponto
14
15    // Prepara a query SQL para inserção ou atualização
16    if ($id) {
17        // Atualização de produto existente
18        $sql = "UPDATE produtos SET id_fornecedor=?, nome=?, descricao=?, preco_unitario=? WHERE id_produto=?";
19        $stmt = $conn->prepare($sql);
20        // Ajuste do número e tipos de parâmetros
21        $stmt->bind_param("issdi", $fornecedor, $nome, $descricao, $preco, $id);
22        // "i" para id_fornecedor e id_produto (inteiros), "s" para strings, "d" para preco_unitario (decimal)
23        $mensagem = "Produto atualizado com sucesso!";
24    } else {
25        // Inserção de novo produto
26        $sql = "INSERT INTO produtos (id_fornecedor, nome, descricao, preco_unitario) VALUES (?, ?, ?, ?)";
27        $stmt = $conn->prepare($sql);
28        // Ajuste do número e tipos de parâmetros
29        $stmt->bind_param("issd", $fornecedor, $nome, $descricao, $preco);
30        // "i" para id_fornecedor (inteiro), "s" para strings, "d" para preco_unitario (decimal)
31        $mensagem = "Produto cadastrado com sucesso!";
32    }

```

Fonte – Elaborado pelos autores (2024).

9 CONSIDERAÇÕES FINAIS

Por fim, finalizamos parte do projeto com êxito. Tendo em vista tudo o que foi realizado, mesmo que com dificuldades, ao fim da entrega tivemos um resultado satisfatório e próximo do planejado. Todas as tarefas foram concluídas e com a ajuda dos integrantes, finalizamos no prazo desejado.

Como desenvolvedores, aprendemos muito fazendo o site. Aprendemos nas pesquisas que fizemos para a documentação, e também ampliamos nossos conhecimentos com a utilização das ferramentas. Desenvolvemos nossas habilidades produzindo os códigos e melhoramos nosso trabalho como equipe.

REFERÊNCIAS

- FERREIRA, G. N. Kanban: O que é, como funciona e quais suas vantagens? **K21**. 16 nov. 2022. Disponível em: <https://br.k21.global/gestao-de-times-ageis/kanban-o-que-e-como-funciona-vantagens>. Acesso em: 24 set. 2024.
- VILLAIN, Matheus; SILVEIRA, Maria. Figma: o que é a ferramenta, design e como usar. **Allura**. 22 ago. 2024. Disponível em: <https://www.alura.com.br/artigos/figma>. Acesso em: 24 set. 2024.
- L., Andrei. O Que é HTML: O Guia Definitivo para Iniciantes: **Hostigner Tutoriais**. 7 jul. 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 24 set. 2024.
- G, Ariane de. O que é CSS? Guia Básico para Iniciantes: **Hostigner Tutoriais**. 12 dez. 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 24 set. 2024.
- HAIM, Itamar de. What Is CSS? and How to Use it in Web Design. **Elementor**. 21 mar. 2024. Disponível em: <https://elementor.com/blog/what-is-css>. Acesso em: 24 set. 2024.
- ANÔNIMO. O que é: Responsividade do sistema. **Five Performance Digital**. Disponível em: <https://fiveperformancedigital.com.br/glossario/o-que-e-responsividade-do-sistema/>. Acesso em: 24 set. 2024.
- WEBSHARE. O que é responsividade? **Webshare**. 11 fev. 2022. Disponível em: <https://www.webshare.com.br/blog/glossario-o-que-e-responsividade/>. Acesso em: 24 set. 2024.
- ANÔNIMO. O que é o MySQL? **OCI**. Disponível em: <https://www.oracle.com/br/mysql/what-is-mysql/>. Acesso em 12 nov. 2024.
- ANÔNIMO. "O que é um banco de dados?"; **OCI**. Disponível em: [https://www.oracle.com/br/database/what-is-database/\(DBMS\)](https://www.oracle.com/br/database/what-is-database/(DBMS)). Acesso em 12 nov. 2024.
- ANÔNIMO. O que é SQL (linguagem de consulta estruturada)? **AWS**. Disponível em: <https://aws.amazon.com/pt/what-is/sql/>. Acesso em: 12 nov. 2024.

POLIDORO, Priscila. O que é PHP e para que serve? **Locaweb**. 21 mar. 2024. Disponível em: <https://www.locaweb.com.br/blog/temas/codigo-aberto/php/>. Acesso em: 12 nov. 2024.

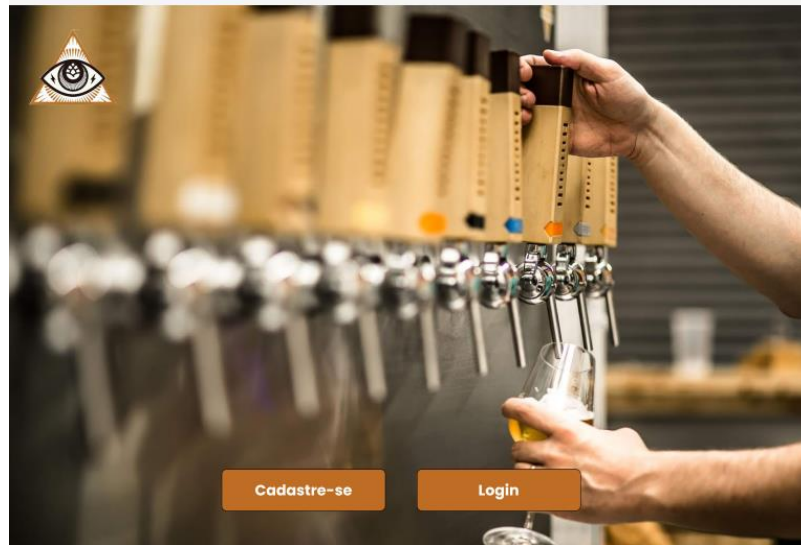
OLIVEIRA, Danielle. Banco de Dados: o que é, principais tipos e um guia para iniciar. **Alura**. Disponível em: <https://www.alura.com.br/artigos/banco-de-dados>. Acesso em 12 nov. 2024.

MAGALHÃES, Beatriz. O que é e como é aplicado MySQL no mercado. **Alura**. Disponível em: <https://www.alura.com.br/artigos/mysql-no-mercado>. Acesso em 12 nov. 2024.

PISA, Pedro. O que é e como usar o MySQL? **Techtudo**. Disponível em: <https://www.techtudo.com.br/noticias/2012/04/o-que-e-e-como-usar-o-mysql.ghml>. Acesso em 12 nov. 2024.

APÊNDICE A

Página inicial



Login

A mockup of a login form. The header is dark with the 'Dogma Cervejaria' logo on the left, the word 'Login' in the center, and a home icon on the right. The main content area has an orange background. In the center, there is a white box with the title 'Login'. Below the title, there are two input fields: 'Usuário:' with a placeholder 'Insira o nome do Usuário...' and 'Senha:' with a placeholder 'Insira a senha...'. Below these fields is a dark brown button labeled 'Entrar'.

Cadastro

A mockup of a registration form. The header is dark with the 'Dogma Cervejaria' logo on the left, the text 'Sistema de Cadastro' in the center, and a home icon on the right. The main content area has an orange background. In the center, there is a white box with the title 'Cadastre-se'. Below the title, there are three input fields: 'Nome:' with a placeholder 'Insira seu nome...', 'Email:' with a placeholder 'Insira seu email...', and 'Telefone:' with a placeholder 'Insira se telefone...'. Below these fields is a dark brown button labeled 'Cadastre-se'.

Sistema de Cadastro


Sistema de Cadastro


Bem-Vindo Admin!

Cadastrar de Fornecedores
Cadastrar de Produtos

Listagem de Produtos
Sair

Entrar

Cadastro 1


Sistema de Cadastro


Listagem de Fornecedores

Nome

Email

Telefone

Cadastrar

Listagem de Fornecedores

ID	Nome	Telefone	Ações	Email
----	------	----------	-------	-------

Cadastro 1


Sistema de Cadastro


Listagem de Fornecedores

Nome

Email

Telefone

Cadastrar

Listagem de Fornecedores

ID	Nome	Telefone	Ações	Email
----	------	----------	-------	-------

Listagem de Produtos


**Listagem de Produtos**

Listagem de Produtos


ID	Nome	Descrição	Preço	Fornecedor	Ações
1	Produto 1	Descrição de Exemplo	99,00	Fornecedor 1	Editar Excluir

[Voltar](#)


Cadastro 2

**Cadastro de Produto**

Cadastre o Produto

Fornecedor:
 

Descrição:

Nome:
 

Preço:

Entrar

APÊNDICE B



Bem-Vindo Admin!

[Cadastro de Fornecedores](#)[Listagem de produtos](#)[Cadastro de Produtos](#)[Sair](#)[Entrar](#)

Listagem de Fornecedores

[Cadastrar](#)

Listagem de Fornecedores

ID	Nome	Telefone	Ações	Email
----	------	----------	-------	-------



Login

Usuário:

Senha:

Cadastre-se

Nome:

Email:

Telefone:

Cadastre o Produto

Fornecedor:

Descrição:

Nome:

Preço:

Entrar

Listagem de Produtos

ID	Nome	Descrição	Preço	Fornecedor	Ações
1	Produto 1	Descrição de Exemplo	99,00	Fornecedor 1	Editar Excluir

Voltar

APÊNDICE C

HTML – Página inicial

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" href="img-e-video/logo-dogma.jpg" type="image/jpg">
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=sw
ap" rel="stylesheet">
  <link rel="stylesheet" href="css/style.css">
  <title>Cervejaria Dogma</title>
</head>

<body>
  <main>
    <div class="container1">
      <div>
        <video autoplay muted loop class="video">
          <source src="img-e-video/video-dogma.mp4" type="video/mp4">
          Seu navegador não suporta a tag de vídeo.
        </video>
      </div>
      <div class="logomarca">
        
      </div>
    </div>
  </main>
  <footer>
    <div class="botoes">
```

```

        <a href="sistema-cadastro.html"><input type="button" value="Cadastre-se"
id="cadastro"></a>
        <a href="login.html">
            <input type="button" value="Login" id="login">
        </a>
    </div>
</footer>
</body>
</html>

```

HTML – Login

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="icon" href="img-e-video/logo-dogma.jpg" type="image/jpg">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=sw
ap" rel="stylesheet">
    <link rel="stylesheet" href="css/style2.css">
    <title>Cervejaria Dogma</title>
</head>

<body>
    <header>
        
        <h1>Login</h1>
    </header>
    <main>
        <div class="form login">

```

```

<h2>Login</h2>
<div class="formulario">
  <form>
    <label>Usuário:</label>
    <input type="text" placeholder="Insira o nome de Usuário...">
    <label>Senha:</label>
    <input type="password" placeholder="Insira a senha...">
    <a href="bem-vindo.html"><button type="button">Entrar</button></a>
  </form>
</div>
</div>
</main>
</body>

</html>

```

HTML – Cadastro

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" href="img-e-video/logo-dogma.jpg" type="image/jpg">
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=sw
ap" rel="stylesheet">
  <link rel="stylesheet" href="css/style2.css">
  <title>Cervejaria Dogma</title>
</head>

<body>
  <header>

```

```

    
    <h1>Sistema de Cadastro</h1>
</header>
<main></main>
    <div class="form cadastro">
        <h2>Cadastre-se</h2>
        <div class="formulario">
            <form1>
                <label>Nome:</label>
                <input type="text" placeholder="Insira o seu nome ...">
                <label>Email:</label>
                <input type="text" placeholder="Insira o seu email...">
                <label>Telefone</label>
                <input type="text" placeholder="Insira o seu telefone...">
                <a href="login.html"><button>Cadastre-se</button></a>
            </form1>
        </div>
    </div>
</main>
</body>
</html>

```

HTML – Seja bem-vindo

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="icon" href="img-e-video/logo-dogma.jpg" type="image/jpg">

```

```

<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=sw
ap" rel="stylesheet">
<link rel="stylesheet" href="css/style2.css">
<title>Cervejaria Dogma</title>
</head>

<body>
<header>

<h1>Sistema de cadastro</h1>
</header>
<main>
<div class="form bem-vindo">
<h2>Bem-vindo Admin</h2>
<div class="btn-bem-vindo">
<a href="sistema-cadastro-fornecedor.html"><button>Cadastro de
Fornecedores</button></a>
<a href="listagem-produtos.html"><button>Listagem de
Produtos</button></a>
<a href="cadastro-de-produto.html"><button>Cadastro de
Produtos</button></a>
<a href="login.html"><button>Sair</button></a>
</div>
<a href="listagem-produtos.html"><button
class="entrar">Entrar</button></a>
</div>
</main>
</body>
</html>

```


HTML – Cadastro de fornecedor

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <link rel="icon" href="img-e-video/logo-dogma.jpg" type="image/jpg">
```

```
  <link
```

```
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=sw
ap" rel="stylesheet">
```

```
  <link rel="stylesheet" href="css/style2.css">
```

```
  <title>Cervejaria Dogma</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    
```

```
    <h1>Sistema de Cadastro</h1>
```

```
  </header>
```

```
  <main>
```

```
    <div class="form fornecedores">
```

```
      <h2>Listagem de Fornecedores</h2>
```

```
      <div class="fornecedor">
```

```
        <input type="text" placeholder="Insira seu telefone...">
```

```
        <input type="text" placeholder="Insira seu nome...">
```

```
        <input type="text" placeholder="Insira seu e-mail...">
```

```
        <a href="listagem-produtos.html"><button>Cadastrar</button></a>
```

```
      </div>
```

```
    <h4>Listagem de Fornecedores</h4>
```

```
    <table>
```

```
      <tr>
```

```

        <th>ID</th>
        <th>Nome</th>
        <th>Descrição</th>
        <th>Preço</th>
        <th>Fornecedor</th>
        <th>Ações</th>
    </tr>
</table>
    <a href="bem-vindo.html" class="voltar">Voltar</a>
</div>
</main>
</body>
</html>

```

HTML – Cadastro de produtos

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="icon" href="img-e-video/logo-dogma.jpg" type="image/jpg">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=sw
ap" rel="stylesheet">
    <link rel="stylesheet" href="css/style2.css">
    <title>Cervejaria Dogma</title>
</head>

<body>
    <header>
        

```

```

    <h1>Cadastro de Produto</h1>
</header>
<main>
    <div class="form produtos">
        <h2>Cadastre o Produto</h2>
        <div class="cadastro produto">
            <form2>
                <label>Fornecedor:</label>
                <select></select>
                <label>Descrição:</label>
                <textarea></textarea>
                <label>Nome:</label>
                <input type="text" placeholder="Insira o nome">
                <label>Preço:</label>
                <input type="text" placeholder="Insira o preço">
            </form>
        </div>
        <a href="listagem-produtos.html"><button>Cadastrar</button></a>
    </div>
</main>
</body>
</html>

```

HTML – Listagem de produto

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="icon" href="img-e-video/logo-dogma.jpg" type="image/jpg">

```

```

<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=sw
ap" rel="stylesheet">
<link rel="stylesheet" href="css/style2.css">
<title>Cervejaria Dogma</title>
</head>

<body>
<header>

<h1>Listagem de Produtos</h1>
</header>
<main>
<div class="form lista">
<h2>Listagem de Produtos</h2>
<table class="table">
<tr>
<th>ID</th>
<th>Nome</th>
<th>Descrição</th>
<th>Preço</th>
<th>Fornecedor</th>
<th>Ações</th>
</tr>
<tr>
<td>1</td>
<td>Produto 1</td>
<td>Descrição de Exemplo</td>
<td>99,00</td>
<td>Fornecedor 1</td>
<td><a href="#" class="a">Editar</a><br><a href="#"
class="a">Excluir</a></td>
</tr>

```

```
        </table>
        <a href="cadastro-de-produto.html" class="volta">Voltar</a>
    </div>
</main>
</body>
</html>
```

CSS – Cabeçalho

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    display: flex;
    height: 100%;
    width: 100%;
    overflow: hidden;
    flex: 1 1 100%;
    font-family: 'Poppins', sans-serif;
    /* Usando a fonte Poppins */
    background-color: #CF8E55;
    justify-content: center;
    align-items: center;
}

.container1 {
    position: relative;
    height: 100vh;
    width: 100vw;
}
```

```
.video {
  position: absolute;
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  object-fit: cover;
  /* Faz o vídeo cobrir a tela inteira sem distorcer */
  z-index: -1;
  /* Mantém o vídeo no fundo */
}

footer {
  display: flex;
  justify-content: center;
  /* Centraliza os botões horizontalmente */
  align-items: center;
  /* Centraliza verticalmente se o footer tiver altura definida */
  /* Cor de fundo do rodapé (opcional) */
  padding: 20px;
  position: absolute;
  bottom: 0;
  width: 300px;
}

.botoes {
  display: flex;
  gap: 20px;
  /* Espaçamento entre os botões */
}

input[type="button"] {
  padding: 10px 20px;
  background-color: #bf6c2467;
```

```

border: 1px solid #A66B38;
border-radius: 5px;
color: white;
font-weight: bold;
font-size: 16px;
cursor: pointer;
transition: background-color 0.3s ease;
/* Suaviza a transição da cor */
}

/* Estilo dos botões ao passar o mouse */
input[type="button"]:hover {
    background-color: #A66B38;
    /* Muda a cor de fundo ao passar o mouse */
    color: black;
    /* Muda a cor do texto */
}

#logo-dogma {
    padding: 2%;
    height: 200px;
    position: fixed;
}

```

CSS 2

```

/* zerando as configurações */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

/* Garantindo que o body ocupe toda a tela e centralize o conteúdo */

```

```
body,
html {
    height: 100%;
    margin: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #CF8E55;
    font-family: 'Poppins';
}

/* configuração geral do cabeçalho */
header {
    background-color: black;
    width: 100%;
    color: white;
    display: flex;
    align-items: center;
    padding: 10px;
    box-sizing: border-box;
    position: absolute;
    top: 0;
}

/* configuração do h1 */
h1 {
    margin-left: 2%;
}

/* configuração geral do h2 */
h2 {
    text-align: center;
    margin-top: 20px;
    width: 100%;
```



```
}
```

```
/* configuração do h4 */
```

```
h4 {  
    text-align: center;  
    padding: 0px 0px 5px 0px;  
}
```

```
/* configuração geral dos forms */
```

```
form {  
    display: flex;  
    box-sizing: border-box;  
    padding: 30px;  
    width: 50%;  
    height: 100%;  
    flex-direction: column;  
    gap: 15px;  
}
```

```
form1 {  
    display: flex;  
    box-sizing: border-box;  
    padding: 30px;  
    width: 50%;  
    height: 100%;  
    flex-direction: column;  
    gap: 5px;  
}
```

```
/* flexibilidade dos formularios */
```

```
.formulario {  
    display: flex;  
    justify-content: center;  
    align-items: center;
```

```
}
```

```
/* Fazendo formulario de cadastro de produtos */
```

```
form2 {  
    display: flex;  
    flex-wrap: wrap;  
    padding: 50px;  
    justify-content: center;  
    flex-direction: column;  
    width: 700px;  
    height: 240px;  
}
```

```
/* configuração geral da class form */
```

```
.form {  
    margin-top: 120px;  
    display: block;  
    border: 1px solid black;  
    width: 700px;  
    height: 400px;  
    background-color: white;  
    color: black;  
    box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.75);  
    overflow: hidden;  
}
```

```
/* Iniciando sistema cadastro de fornecedor */
```

```
.fornecedor {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    width: 100%;  
    box-sizing: border-box;  
    gap: 10px;
```

```
padding: 20px 30px 7px;
}

/* configuração geral dos inputs */
input {
    height: 40px;
    border-radius: 5px;
    display: flex;
    border-color: black;
}

.formulario input {
    background-color: white;
    display: flex;
    color: #A66B38;
    font-weight: normal;
    width: 100%;
    padding: 10px;
    justify-content: center flex-start;
    align-items: center;
}

.formulario a {
    display: flex;
    justify-content: center;
    align-items: center;
}

form1 input {
    display: flex;
    color: #A66B38;
    font-weight: normal;
    width: 250px;
    padding: 10px;
```

```
    justify-content: center;
    align-items: center;
}
```

```
/* botoes input formulario 2 */
```

```
form2 input {
    display: flex;
    color: #A66B38;
    font-weight: normal;
    width: 250px;
    padding: 10px;
    justify-content: center;
    align-items: center;
}
```

```
.fornecedor input {
    display: flex;
    color: #A66B38;
    font-weight: normal;
    width: 250px;
    padding: 10px;
    justify-content: center;
    align-items: center;
}
```

```
/* configuração geral dos placeholders */
```

```
::placeholder {
    color: #CF8E55;
}
```

```
/* configuração geral dos botoes */
```

```
button {
    background-color: #A66B38;
    width: 50%;
```

```
color: white;
align-self: center;
height: 35px;
font-family: 'Poppins';
font-weight: bold;
margin-top: 20px;
transition: background-color 0.3 ease;
cursor: pointer;
border-radius: 5px;
}
```

```
button.entrar {
  width: 120px;
  position: relative;
  margin-left: 41%;
}
```

```
/* hover de todos os botoes */
button:hover {
  background-color: #CF8E55;
  color: black;
}
```

```
/* class dos botões da landing page bem-vindo */
.btn-bem-vindo {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
  align-items: center;
  width: 100%;
  box-sizing: border-box;
  gap: 15px;
  padding: 40px;
  transition: background-color 0.3 ease;
```

```
    cursor: pointer;
    border-radius: 5px;
}

/* mexendo no botao da landing page bem-vindo */
.btn-bem-vindo button {
    background-color: white;
    display: flex;
    color: #A66B38;
    font-weight: normal;
    width: 250px;
    padding: 10px;
    justify-content: center flex-start;
    align-items: center;
}

.btn-bem-vindo button:hover {
    background-color: #CF8E55;
    color: white;
}

/* configuração botao cadastro fornecedores */
.fornecedor button {
    width: 100px;
    margin-top: 10px;
}

/* botao sistema de cadastro fornecedores */
.produtos button {
    width: 120px;
    position: relative;
    margin-left: 38%;
}
```

```
/* tamanho da logo dogma preta */
.logo-preta {
    height: 100px;
    margin-left: 20px;
}

/* configuração link voltar fornecedores */
a {
    text-decoration: none;
}

.fornecedores a:hover {
    color: black;
}

/* configurando os fornecedores da tabela */
.fornecedores table {
    width: 100%;
    margin-top: 0;
    padding: 0px;
}

/* tabela de cadastro fornecedores */
.form table {
    display: flex;
}

/* configurando a tabela */
table {
    width: 100%;
    border-collapse: collapse;
    justify-content: center;
    padding: 40px;
```

```
    align-items: center;
    margin-top: 40px;
}

/* mexendo no th e td */
th,
td {
    border: 3px solid #D8C1AB;
    /* Adiciona borda ao redor de cada célula */
    padding: 10px;
    text-align: left;
    font-size: 12px;
}

/* mexendo no th */
th {
    background-color: #D8C1AB;
    color: #4E372B;
    border: 3px solid #A66B38;
    width: 100%;
}

/* mudando o fundo do td */
td {
    background-color: #F7EFE5;
}

/* configuração de link especial */
.a {
    color: #C45C2C;
    text-decoration: none;
}

/* fazendo a mudança de cor com o hover */
```



```
.a:hover {  
    text-decoration: underline;  
}
```

```
/* configuração do select */
```

```
select {  
    display: flex;  
    color: black;  
    font-weight: normal;  
    width: 250px;  
    padding: 10px;  
    justify-content: center;  
    align-items: center;  
    border-radius: 5px;  
    border-width: 2px;  
    border-color: black;  
}
```

```
/* configuração do textarea */
```

```
textarea {  
    width: 250px;  
    height: 40px;  
    border-color: black;  
    border-width: 2px;  
    border-radius: 5px;  
}
```

```
.voltar{  
    color: #A66B38;  
    margin-left: 30px;  
}
```

```
.volta{  
    color: #A66B38;
```

```
margin-left: 30px;
transition: ease 0.3s;
}
```

```
.volta:hover{
  color: black;
}
```

```
@media (max-width: 480px) {
```

```
  .form {
    width: 300px;
    margin-top: 200px;
    margin-bottom: 20px;
    border-radius: 5px;
  }
```

```
form {
  padding: 0;
  margin-top: 20px;
  width: 90%;
  padding: 10px;
  height: auto;
}
```

```
form1 {
  padding: 0;
  margin-top: 20px;
  width: 90%;
}
```

```
form2 {  
  padding: 0%;  
  gap: 2;  
  height: 400px;  
  margin-top: -60px;  
  margin-bottom: -70px;  
  width: 320px;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  margin-left: -10px;  
  
}
```

```
button {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

```
table {  
  display: flex;  
  margin-left: 15px;  
}
```

```
tbody {  
  width: 300px;  
}
```

```
th {
```

```
    font-size: 6.5px;  
    height: 3px;  
    width: 10px;  
    padding: 9px;  
}
```

```
a {  
    display: flex;  
    font-size: 12px;  
}
```

```
.table {  
    display: flex;  
    padding: 0;  
}
```

```
tbody {  
    width: 300px;  
}
```

```
th {  
    font-size: 6.5px;  
    height: 3px;  
    width: 10px;  
    padding: 9px;  
}
```

```
td{  
    font-size: 6.5px;  
}
```

```
.a{  
    font-size: 6.5px;
```

```
}
```

```
.voltar{  
  color: #A66B38  
}
```

```
.btn-bem-vindo{  
  margin-top: -20px;  
  gap: 5px;  
  margin-bottom: -30px;  
  padding: 5px 0px 10px;  
  margin-bottom: 10px;  
  margin-top: 20px;  
}  
}
```

APÊNDICE D

```
CREATE DATABASE sprint;
```

```
USE sprint;
```

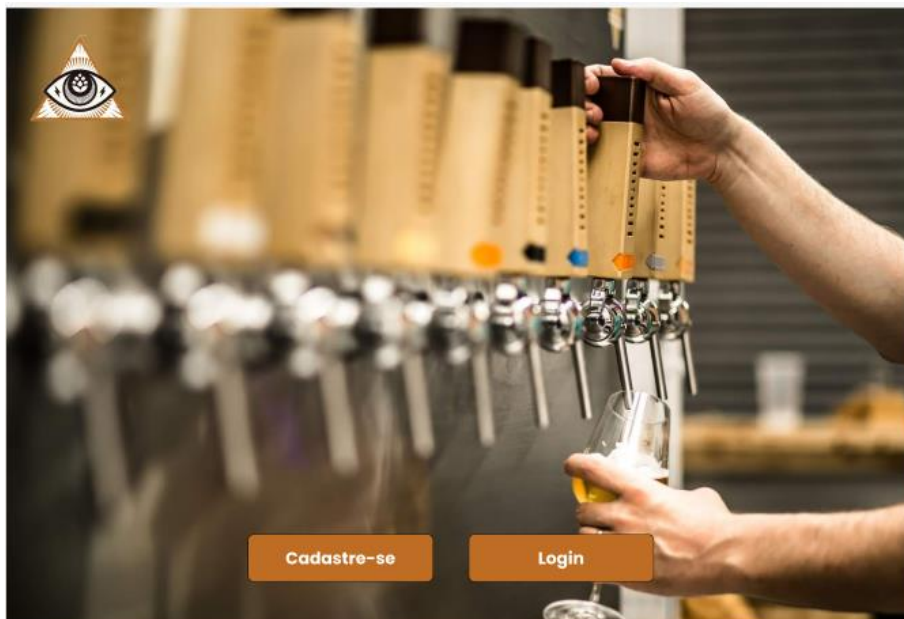
```
CREATE TABLE fornecedores (  
    id_fornecedor INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255) NOT NULL,  
    cnpj INT(18) NOT NULL,  
    endereco VARCHAR(255),  
    telefone INT(11) NOT NULL,  
    email VARCHAR(255),  
    observacoes TEXT  
);
```

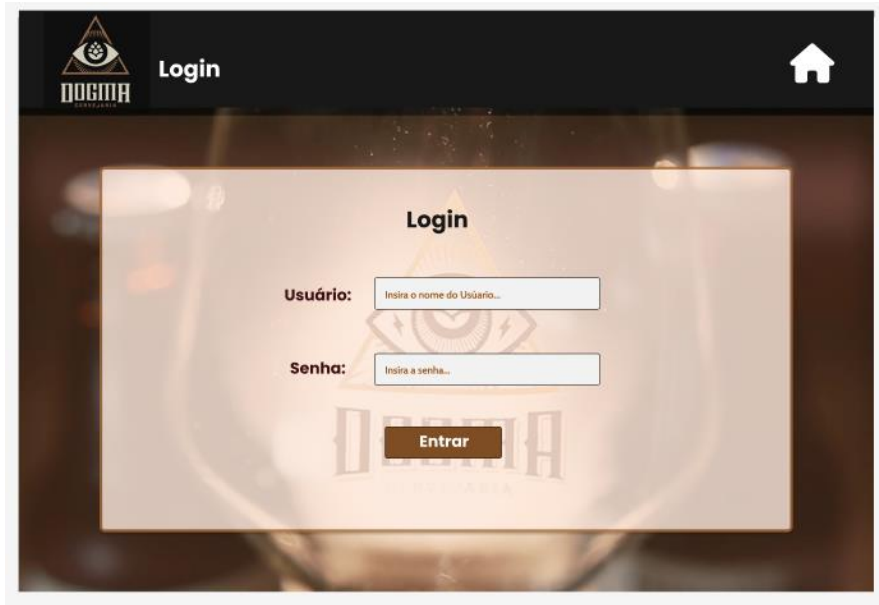
```
CREATE TABLE produtos (  
    id_produto INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255) NOT NULL,  
    codigo INT(30) NOT NULL,  
    descricao TEXT,  
    quantidade_estoque INT NOT NULL,  
    preco_unitario DECIMAL(10, 2) NOT NULL,  
    id_fornecedor INT,  
    FOREIGN KEY (id_fornecedor) REFERENCES fornecedores(id_fornecedor)  
);
```

```
CREATE TABLE funcionarios(  
    id_funcionarios INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(255) NOT NULL,  
    endereco VARCHAR(255),  
    telefone INT(11) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    senha INT(6) NOT NULL  
);
```

```
SELECT produtos.nome AS nome_produto,  
       produtos.preco_unitario AS preco,  
       fornecedores.nome AS nome_fornecedor  
FROM produtos  
JOIN fornecedores ON produtos.id_fornecedor = fornecedores.id_fornecedor;
```

APÊNDICE E - Desktop





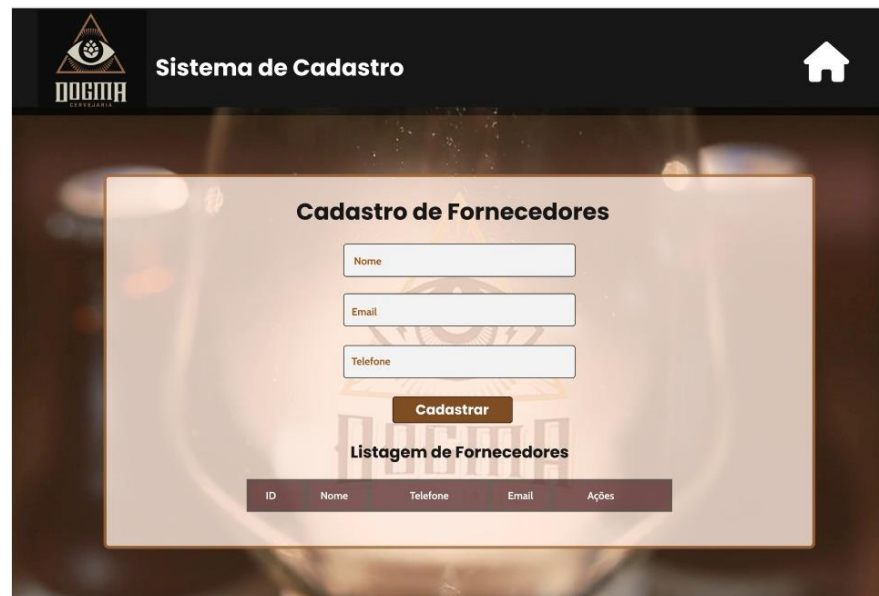
The image shows a web application interface for a login page. At the top, there is a dark header bar containing the 'DOGMA' logo on the left and a home icon on the right. The main content area has a dark background with a large, faint, golden 'DOGMA' watermark. In the center, there is a white rectangular box titled 'Login'. Inside this box, there are two input fields: 'Usuário:' with a placeholder 'Insira o nome do Usuário...' and 'Senha:' with a placeholder 'Insira a senha...'. Below these fields is a brown button labeled 'Entrar'.

Login

Usuário:

Senha:

Entrar



The image shows a web application interface for a 'Cadastro de Fornecedores' (Supplier Registration) page. The header is dark with the 'DOGMA' logo and a home icon. The main content area has a dark background with a large, faint, golden 'DOGMA' watermark. In the center, there is a white rectangular box titled 'Cadastro de Fornecedores'. Inside this box, there are three input fields: 'Nome', 'Email', and 'Telefone'. Below these fields is a brown button labeled 'Cadastrar'. Below the 'Cadastrar' button, there is a section titled 'Listagem de Fornecedores' which contains a table with five columns: 'ID', 'Nome', 'Telefone', 'Email', and 'Ações'.

Cadastro de Fornecedores

Nome

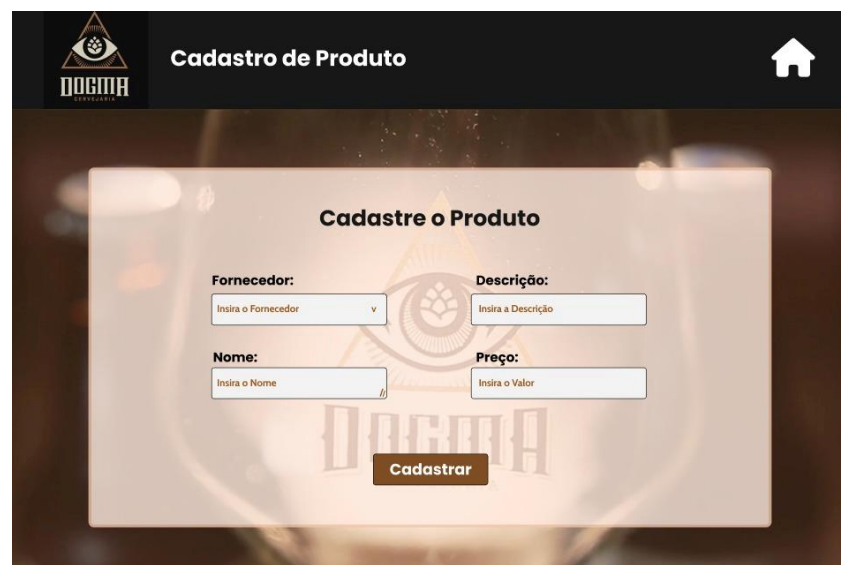
Email

Telefone

Cadastrar

Listagem de Fornecedores

ID	Nome	Telefone	Email	Ações
----	------	----------	-------	-------



The image shows a web application interface for a 'Cadastro de Produto' (Product Registration) page. The header is dark with the 'DOGMA' logo and a home icon. The main content area has a dark background with a large, faint, golden 'DOGMA' watermark. In the center, there is a white rectangular box titled 'Cadastre o Produto'. Inside this box, there are four input fields: 'Fornecedor:' with a placeholder 'Insira o Fornecedor', 'Descrição:' with a placeholder 'Insira a Descrição', 'Nome:' with a placeholder 'Insira o Nome', and 'Preço:' with a placeholder 'Insira o Valor'. Below these fields is a brown button labeled 'Cadastrar'.

Cadastre o Produto

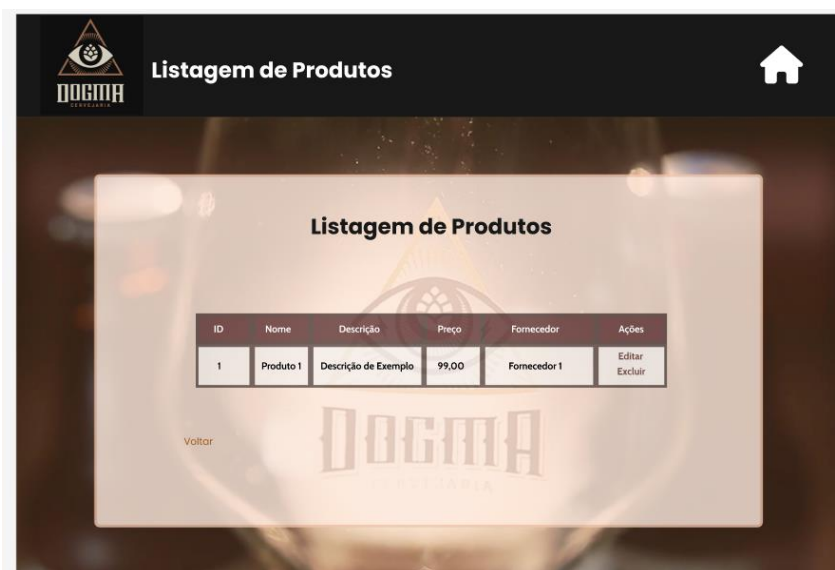
Fornecedor:

Descrição:

Nome:

Preço:

Cadastrar



Mobile





Cadastre o Produto

Fornecedor:

Descrição:

Nome:

Preço:

Cadastrar

Listagem de Produtos

ID	Nome	Descrição	Preço	Fornecedor	Ações
1	Produto 1	Descrição de Exemplo	99,00	Fornecedor 1	Editar Excluir

[Voltar](#)

APÊNDICE F

PHP – Cadastro de fornecedores

```
<?php
// Inclui o arquivo que valida a sessão do usuário
include('valida_sessao.php');

// Inclui o arquivo de conexão com o banco de dados
include('conexao.php');

// Verifica se o formulário foi enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $telefone = $_POST['telefone'];

    // Processa o upload da imagem
    // Prepara a query SQL para inserção ou atualização
    if ($id) {
        // Se o ID existe, é uma atualização
        $sql = "UPDATE fornecedores SET nome='$nome', email='$email',
telefone='$telefone'";
        $sql .= " WHERE id_fornecedor='$id'";
        $mensagem = "Fornecedor atualizado com sucesso!";
    } else {
        // Se não há ID, é uma nova inserção
        $sql = "INSERT INTO fornecedores (nome, email, telefone) VALUES
('$nome', '$email', '$telefone')";
        $mensagem = "Fornecedor cadastrado com sucesso!";
    }

    // Executa a query e verifica se houve erro
    if ($conn->query($sql) === TRUE) {
        $mensagem = $mensagem;
    } else {
        $mensagem = "Erro: " . $conn->error;
    }
}

// Verifica se foi solicitada a exclusão de um fornecedor
if (isset($_GET['delete_id'])) {
    $delete_id = $_GET['delete_id'];

    // Verifica se o fornecedor tem produtos cadastrados
    $check_produtos = $conn->query("SELECT COUNT(*) as count FROM produtos
WHERE id_fornecedor = '$delete_id'")->fetch_assoc();
```

```

    if ($check_produtos['count'] > 0) {
        $mensagem = "Não é possível excluir este fornecedor pois existem
produtos cadastrados para ele.";
    } else {
        $sql = "DELETE FROM fornecedores WHERE id_fornecedor='$delete_id'";
        if ($conn->query($sql) === TRUE) {
            $mensagem = "Fornecedor excluído com sucesso!";
        } else {
            $mensagem = "Erro ao excluir fornecedor: " . $conn->error;
        }
    }
}

// Busca todos os fornecedores para listar na tabela
$fornecedores = $conn->query("SELECT * FROM fornecedores");

// Se foi solicitada a edição de um fornecedor, busca os dados dele
$fornecedor = null;
if (isset($_GET['edit_id'])) {
    $edit_id = $_GET['edit_id'];
    $fornecedor = $conn->query("SELECT * FROM fornecedores WHERE
id_fornecedor='$edit_id'")->fetch_assoc();
}
?>

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Cadastro de Fornecedor</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <header>
        
        <h1>Sistema de Cadastro</h1>
    </header>
    <div class="container" style="width: 900px;">
        <!-- Formulário para cadastro/edição de fornecedor -->
        <form method="post" action="" enctype="multipart/form-data"
class="form">
            <h2>Cadastro de Fornecedor</h2>
            <input type="hidden" name="id" value="<?php echo
$fornecedor['id_fornecedor'] ?? ''; ?>">

```

```
<input type="text" name="nome" id="nome" value="<?php echo
$fornecedor['nome'] ?? ''; ?>" required placeholder="Nome">

<input type="email" name="email" id="email" value="<?php echo
$fornecedor['email'] ?? ''; ?>" placeholder="Email">

<input type="text" name="telefone" id="telefone" value="<?php echo
$fornecedor['telefone'] ?? ''; ?>" placeholder="Telefone">

<button type="submit"><?php echo $fornecedor ? 'Atualizar' :
'Cadastrar'; ?></button>

<!-- Exibe mensagens de sucesso ou erro -->
<?php
if (isset($mensagem)) echo "<p class='message " . (strpos($mensagem,
'Erro') !== false ? "error" : "success") . "'>$mensagem</p>";
if (isset($mensagem_erro)) echo "<p class='message
error'>$mensagem_erro</p>";
?>

<h2>Listagem de Fornecedores</h2>
<!-- Tabela para listar os fornecedores cadastrados -->
<table>
    <tr>
        <th>ID</th>
        <th>Nome</th>
        <th>Telefone</th>
        <th>Email</th>
        <th>Ações</th>
    </tr>
    <?php while ($row = $fornecedores->fetch_assoc()): ?>
        <tr>
            <td><?php echo $row['id_fornecedor']; ?></td>
            <td><?php echo $row['nome']; ?></td>
            <td>
                <?php
                echo preg_replace('/(\d{2})(\d{5})(\d{4})/', '($1)
$2-$3', $row['telefone']);
                ?>
            </td>
            <td><?php echo $row['email']; ?></td>
            <td>
                <a href="?edit_id=<?php echo
$row['id_fornecedor']; ?>">Editar</a>
                <a href="?delete_id=<?php echo
$row['id_fornecedor']; ?>" onclick="return confirm('Tem certeza que deseja
excluir?')">Excluir</a>
            </td>
        </tr>
    </?php>
</table>
```

```

        <?php endwhile; ?>
    </table>
    <div class="actions">
        <a href="pagina.php" class="back-button">Voltar</a>
    </div>
</form>

</div>
</body>

</html>

```

PHP – Cadastro de produtos

```

<?php
// Inclui o arquivo que valida a sessão do usuário
include('valida_sessao.php');
// Inclui o arquivo de conexão com o banco de dados
include('conexao.php');

// Verifica se o formulário foi enviado
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $id = $_POST['id_produto'] ?? '';
    $fornecedor = $_POST['id_fornecedor'];
    $nome = $_POST['nome'];
    $descricao = $_POST['descricao'];
    $preco = str_replace(',', '.', $_POST['preco_unitario']); // Converte
vírgula para ponto

    // Prepara a query SQL para inserção ou atualização
    if ($id) {
        // Atualização de produto existente
        $sql = "UPDATE produtos SET id_fornecedor=?, nome=?, descricao=?,
preco_unitario=? WHERE id_produto=?";
        $stmt = $conn->prepare($sql);
        // Ajuste do número e tipos de parâmetros
        $stmt->bind_param("issdi", $fornecedor, $nome, $descricao, $preco,
$id);
        // "i" para id_fornecedor e id_produto (inteiros), "s" para strings,
"d" para preco_unitario (decimal)
        $mensagem = "Produto atualizado com sucesso!";
    } else {
        // Inserção de novo produto
        $sql = "INSERT INTO produtos (id_fornecedor, nome, descricao,
preco_unitario) VALUES (?, ?, ?, ?)";
        $stmt = $conn->prepare($sql);
        // Ajuste do número e tipos de parâmetros
        $stmt->bind_param("issd", $fornecedor, $nome, $descricao, $preco);
    }
}

```



```

        // "i" para id_fornecedor (inteiro), "s" para strings, "d" para
        preco_unitario (decimal)
        $mensagem = "Produto cadastrado com sucesso!";
    }

    // Executa a query e verifica se houve erro
    if ($stmt->execute()) {
        $class = "success";
    } else {
        $mensagem = "Erro: " . $stmt->error;
        $class = "error";
    }
}

// Verifica se foi solicitada a exclusão de um produto
if (isset($_GET['delete_id'])) {
    $delete_id = $_GET['delete_id'];
    $sql = "DELETE FROM produtos WHERE id_produto=?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("i", $delete_id);
    if ($stmt->execute()) {
        $mensagem = "Produto excluído com sucesso!";
        $class = "success";
    } else {
        $mensagem = "Erro ao excluir produto: " . $stmt->error;
        $class = "error";
    }
}

// Busca todos os produtos para listar na tabela
$productos = $conn->query("SELECT p.id_produto, p.nome, p.descricao,
p.preco_unitario, f.nome AS fornecedor_nome FROM produtos p JOIN fornecedores
f ON p.id_fornecedor = f.id_fornecedor");

// Se foi solicitada a edição de um produto, busca os dados dele
$producto = null;
if (isset($_GET['edit_id'])) {
    $edit_id = $_GET['edit_id'];
    $stmt = $conn->prepare("SELECT * FROM produtos WHERE id_produto=?");
    $stmt->bind_param("i", $edit_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $produto = $result->fetch_assoc();
    $stmt->close();
}

// Busca todos os fornecedores para o select do formulário

```

```

$provedores = $conn->query("SELECT id_provedor, nome FROM provedores");
?>

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cadastro de Produto</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <header>
        
        <h1>Cadastro de Produtos</h1>
    </header>
    <div class="container">

        <!-- Formulário para cadastro/edição de produto -->
        <form method="post" action="" enctype="multipart/form-data"
class="form2">
            <h2>Cadastro de Produto</h2>
            <input type="hidden" name="id" value="<?php echo
$produto['id_produto'] ?? '' ?>">

            <label>Provedor:</label>
            <select name="id_provedor" required>
                <?php while ($row = $provedores->fetch_assoc()): ?>
                    <option value="<?php echo $row['id_provedor']; ?>" <?php
if ($produto && $produto['id_provedor'] == $row['id_provedor']) echo
'selected'; ?>><?php echo $row['nome']; ?></option>
                <?php endwhile; ?>
            </select>

            <label>Nome:</label>
            <input type="text" name="nome" value="<?php echo $produto['nome']
?? '' ?>" required placeholder="Insira o Nome">

            <label>Descrição:</label>
            <textarea name="descricao" placeholder="Insira a Descrição"><?php
echo $produto['descricao'] ?? '' ?></textarea>

            <label>Preço: </label>
            <input type="text" name="preco_unitario" value="<?php echo
$produto['preco_unitario'] ?? '' ?>" required id="preco" placeholder="Insira
o Valor">

```

```

        <button type="submit"><?php echo $produto ? 'Atualizar' :
'Cadastrar'; ?></button>
        <?php if (isset($mensagem)): ?>
            <p class="message <?php echo $class; ?>"><?php echo $mensagem;
?></p>

            <?php endif; ?>
            <a href="pagina.php" class="back-button">Voltar</a>
        </form>

    </div>
</body>

</html>

```

PHP – Conexão PHP

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sprint";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error){
    die("Connection failed: " . $conn->connect_error);
}

?>

```

PHP – Index

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sprint";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error){
    die("Connection failed: " . $conn->connect_error);
}

?>

```

PHP – Listagem de Produtos

```

<?php include('valida_sessao.php'); ?>
<?php include('conexao.php'); ?>

<?php
if (isset($_GET['delete_id'])) {
    $delete_id = $_GET['delete_id'];
    $sql = "DELETE FROM produtos WHERE id_produto='$delete_id'";
    if ($conn->query($sql) === TRUE) {
        $mensagem = "Produto excluído com sucesso!";
    } else {
        $mensagem = "Erro ao excluir produto: " . $conn->error;
    }
}
}
$produtos = $conn->query("SELECT p.id_produto, p.nome, p.descricao,
p.preco_unitario, f.nome AS fornecedor FROM produtos p JOIN fornecedores f ON
p.id_fornecedor = f.id_fornecedor ");
?>

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Listagem de Produtos</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
<header>
    
    <h1>Listagem de Produtos</h1>
</header>
<div class="container">
    <form class="form3">
        <h2>Listagem de Produtos</h2>
        <?php if (isset($mensagem)) echo "<p class='message " . ($conn-
>error ? "error" : "success") . "'>$mensagem</p>"; ?>
        <table class="tabela">
            <tr>
                <th>ID</th>
                <th>Nome</th>
                <th>Descrição</th>
                <th>Preço</th>
                <th>Fornecedor</th>
                <th>Ações</th>
            </tr>
            <?php while ($row = $produtos->fetch_assoc()): ?>

```

```

        <tr>
            <td><?php echo $row['id_produto']; ?></td>
            <td><?php echo $row['nome']; ?></td>
            <td><?php echo $row['descricao']; ?></td>
            <td><?php echo 'R$ ' .
number_format($row['preco_unitario'], 2, ',', '.'); ?></td>
            <td><?php echo $row['fornecedor']; ?></td>
            <td>
                <a href="cadastro_produto.php?edit_id=<?php
echo $row['id_produto']; ?>">Editar</a>
                <a href="?delete_id=<?php echo
$row['id_produto']; ?>" onclick="return confirm('Tem certeza que deseja
excluir?')">Excluir</a>
            </td>
        </tr>
    </table>
    <?php endwhile; ?>
    <a href="pagina.php" class="back">Voltar</a>
</form>
</div>

</body>

</html>

```

PHP – Login

```

<?php
session_start();
include('conexao.php');

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nome = $_POST['nome'];
    $senha =($_POST['senha']);

    $sql = "SELECT * FROM funcionarios WHERE nome='$nome' AND senha='$senha'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        $_SESSION['nome'] = $nome;
        header('Location: pagina.php');
    } else {
        $error = "Usuario ou senha inválidos.";
    }
}

?>
<!DOCTYPE html>
<html lang="pt-br">

```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Login</title>
  <link rel="shortcut icon" href="img-e-video/logo-dogma.jpg"
type="image/jpg">
</head>

<body>
  <header>
    
    <h1>Login</h1>
  </header>
  <div class="container">

    <form action="" method="post"> <h2>Login</h2>
    <label for="usuario">Usuário:</label>
    <input type="text" name="nome" required placeholder="Insira o nome
de usuário:">
    <label for="senha">Senha:</label>
    <input type="password" name="senha" required placeholder="Insira
sua senha:">
    <button type="submit" style="margin-bottom: 30px;">Entrar</button>
    <?php if (isset($error)) echo "<p class='message
error'>$error</p>"; ?>
    </form>
  </div>
</body>

</html>

```

PHP – Logout

```

<?php
// Inicia a sessão para garantir que ela está ativa.
session_start();

// Destroi todas as variáveis da sessão, encerrando a sessão do usuário.
session_destroy();

// Redireciona o usuário para a página de login.
header('Location: login.php');

// Garante que nenhum código adicional seja executado após o redirecionamento.
exit();
?>

```

PHP – Página Inicial

```
<?php include('valida_sessao.php'); ?>
<!-- Inclui o arquivo 'valida_sessao.php' para garantir que o usuário esteja
autenticado -->

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <title>Painel Principal</title>
    <!-- Link para o arquivo CSS para estilização da página -->
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <header>
        
        <h1>Sistema de Cadastro</h1>
    </header>
    <div class="container">
        <!-- Exibe uma mensagem de boas-vindas com o nome do usuário logado --
>

        <div class="menu">
            <h2>Bem-Vindo, <?php echo $_SESSION['nome']; ?>!</h2>
            <a href="cadastro_fornecedor.php"><input type="button"
value="Cadastro de Fornecedores"></a>
            <a href="cadastro_produto.php"><input type="button"
value="Cadastro de produtos"></a>
            <a href="listagem_produtos.php"><input type="button"
value="Listagem de produtos"></a>
            <a href="login.php"><input type="button" value="Sair"
id="sair"></a>
        </div>
    </div>
</body>

</html>
```

PHP – Validação

```
<?php
// Inicia a sessão para verificar se o usuário está autenticado.
session_start();

// Verifica se a variável de sessão 'usuario' não está definida.
if (!isset($_SESSION['nome'])) {
```

```
// Se o usuário não estiver logado, redireciona para a página de login.
header('Location: login.php');

// Garante que o restante do código não seja executado após o
redirecionamento.
exit();
}
?>
```

CSS – Style do Index.php

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,
300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,
700;1,800;1,900&display=swap');

*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', serif;
}
:root{
  --primary-color: #BF6D24;
  --segundary-color: #A66B38;
  --background-color: #f19a4e;
}
body {
  display: flex;
  height: 100vh;
  justify-content: center; /* Centraliza horizontalmente */
  align-items: flex-end; /* Centraliza verticalmente */
}

#background-video {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  object-fit: cover;
  z-index: -1;
}

.btn {
  display: flex; /* Mantém flex caso tenha itens internos */
  height: 40px;
  width: 250px;
```



```

        justify-content: center;
        margin-bottom: 5vh;
    }
    input[type="button"] {
        height: 50px;
        width: 150px;
        background-color: var(--primary-color);
        border: 0;
        border-radius: 7px;
        color: white;
        font-weight: 500;
        font-size: 18px;
        cursor: pointer;
        transition: 0.3s ease;
    }
    input[type="button"]:hover{
        background-color: var(--background-color);
        color: rgb(108, 81, 49);
    }
    img{
        opacity: 80%;
        max-height: 150px;
        justify-self: center;
        align-self: center;
    }
    header{
        display: flex;
        align-self: flex-start;
        width: 200px;
        position: absolute;
        left: 0;
        padding: 20px;
    }
}

```

CSS – Style de todas as outras páginas

```

@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

/* Resetando estilos padrão do navegador */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

/* Definindo variáveis para cores e fontes */

```

```

:root {
  --borderTable: #3E0F12;
  --primary-color: #BF6D24;
  --primary-color-1: #9D5719;
  --secondary-color: #261816;
  --segundary-color-1: #181818;
  --bloquinho: #FFEBDF;
  --botao: #7E4D23;
  --corLabeleTable: #3E0F12;
  --backInput: #F2F2F2;
  --error-color: #e74c3c;
  --success-color: #2ecc71;
  --font-family: 'Poppins', sans-serif;
  --headertext: #FFFFFF;
  --borda: #A66B38;
  --hoverInput: #bf6c2492;
  --hoverBorda: #cf9b6d;
  --backTable: #754E48;
}

/* Estilos globais */
body {
  height: 100vh;
  font-family: var(--font-family);
  background-color: var(--background-color);
  color: var(--text-color);
  line-height: 1.6;
  background: url(img-e-video/backgroundPC.png);
  background-repeat: repeat;
  background-size: cover;
  /* Ajusta o tamanho da imagem para cobrir toda a área, mantendo a
proporção */
  background-position: center;
  /* Centraliza a imagem */
}
header img{
  height: 90%;
  margin-left: 30px;
}
header{
  display: flex;
  background-color: var(--segundary-color-1);
  width: 100%;
  height: 80px;
  align-items: center;
  opacity: 90%;
}
h1{
  color: var(--headertext);

```

```

    margin-left: 50px;
    font-size: 1.5em;
    width: 450px;
}
.container {
    width: 900px;
    justify-self: center;
    margin-top: 3%;
}

/* Estilos para cabeçalhos */
h2 {
    color: var(--secondary-color);
    margin-bottom: 20px;
    text-align: center;
    margin-top: 15px;
    width: 100%;
    height: 20px;
}

/* Estilos para formulários */
form {
    background-color: var(--bloquinho);
    height: auto;
    padding: 20px;
    display: flex;
    flex-direction: column;
    border-radius: 7px;
    border: solid var(--borda);
    opacity: 90%;
}
label{
    color: var(--corLabeleTable);
    font-size: 1.5rem;
    margin-top: 40px;
    margin-left: 100px;
    font-weight: 600;
}
input[type="text"],
input[type="password"]{
    width: 400px;
    height: 50px;
    border-radius: 7px;
    margin-left: 250px;
    padding: 7px;
    margin-top: -40px;
    background-color: var(--backInput);
    border: solid var(--borda);
    font-size: 16px;

```

```

}
input::placeholder{
  color: #181818;
  font-weight: 400;
}

button[type="submit"]{
  background-color: var(--botao);
  height: 50px;
  width: 150px;
  margin-left: 350px;
  margin-top: 50px;
  border-radius: 7px;
  border: solid var(--primary-color-1);
  color: var(--headertext);
  font-size: 21px;
  cursor: pointer;
  font-weight: bold;
  opacity: 100%;
  transition: 0.3s ease;
}
button[type="submit"]:hover{
  background-color: var(--primary-color);
  border: solid var(--hoverBorda);
}
.menu h2{
  margin-bottom: -40px;
}
.menu{
  background-color: var(--bloquinho);
  height: 450px;
  padding: 20px;
  display: flex;
  flex-direction: row;
  border-radius: 7px;
  border: solid var(--borda);
  opacity: 90%;
  flex-wrap: wrap;
}
.menu input[type="button"]{
  width: 300px;
  height: 90px;
  border-radius: 5px;
  background-color: var(--backInput);
  border: solid var(--borda);
  font-size: 1.3rem;
  cursor: pointer;
  color: var(--primary-color-1);
  font-weight: bolder;
}

```

```

    text-align: start;
    padding: 10px;
    transition: 0.3s ease;
}
.menu input[type="button"]:hover{
    background-color: var(--hoverInput);
    border: solid var(--hoverBorda);
}
#sair{
    background-color: var(--botao);
    color: var(--headertext);
    transition: 0.3s ease;
}
#sair:hover{
    background-color: var(--primary-color);
    border: solid var(--hoverInput);
}
.menu a{
    height: 50px;
    width: 300px;
    z-index: -1;
    margin-left: 90px;
    margin-top: 20px;
    margin-bottom: -20px;
}
#nome, #email, #telefone{
    margin-top: 30px;
    margin-left: 0;
}
.form{
    height: auto;
    align-items: center;
}
.form input{
    margin-top: 400px;
    width: 400px;
    height: 50px;
    border-radius: 5px;
    padding: 7px;
    margin-top: -40px;
    background-color: var(--backInput);
    border: solid var(--borda);
    font-size: 16px;
}
.form button{
    margin-left: 0;
}
.message{
    background-color: var(--success-color);

```

```

padding: 7px;
color: var(--secondary-color-1);
border-radius: 7px;
margin: 20px;
}
.error{
background-color: var(--error-color);
padding: 7px;
color: var(--secondary-color-1);
border-radius: 7px;
margin-top: 20px;
}
table {
border-collapse: collapse; /* Faz com que as bordas se juntem */
border-spacing: 0; /* Remove qualquer espaçamento entre as células */
font-size: 18px;
}
table a{
text-decoration: none;
color: var(--botao);
transition: 0.3s ease;
font-weight: 600;
}
table a:hover{
color: var(--borderTable);
}
th, td{
border: solid 5px var(--borderTable);
padding: 5px;
}
th{
color: var(--headertext);
background-color: var(--backTable);
padding: 7px;
}
td{
background-color: var(--backInput);
opacity: 80%;
}
.actions {
margin-top: 20px;
width: 100%;
display: flex;
justify-content: flex-start; /* Centraliza horizontalmente */
align-items: center; /* Centraliza verticalmente */
padding: 10px;
}
.actions a {

```

```

width: 100px;
height: 40px;
background-color: var(--botao);
color: var(--backInput);
display: flex;           /* Garante que o conteúdo interno do link também
seja centralizado */
justify-content: center;
align-items: center;
text-decoration: none;   /* Remove o sublinhado padrão do link */
border-radius: 5px;      /* Opcional: Adiciona bordas arredondadas */
font-weight: bold;       /* Opcional: Torna o texto mais visível */
transition: 0.3s ease;
}
.actions a:hover{
background-color: var(--hoverInput);
color: var(--borderTable);
border: solid var(--borda);
}
.form2{
background-color: var(--bloquinho);
height: auto;
padding: 20px;
display: flex;
flex-direction: row;
border-radius: 7px;
border: solid var(--borda);
opacity: 90%;
flex-wrap: wrap;
justify-items: center;
}
.form2 select, textarea{
width: 300px;
height: 50px;
border-radius: 5px;
margin-top: 30px;
margin-right: 40px;
margin-left: -98px;
padding: 7px;
background-color: var(--backInput);
border: solid var(--borda);
font-size: 16px;
z-index: -100;
}
.form2 select{
margin-left: -110px;
}
.form2 input{
width: 300px;
margin-left: -60px;

```

```

    margin-top: 30px;
    z-index: -100;
}
.form2 label{
    position: relative;
    margin: 0;
    padding: 0;
    font-size: 18px;
    z-index: -1;
    height: 20px;
    margin-left: 65px;
    font-weight: bold;
}
.form2 textarea{
    margin-right: 45px;
}

.form2 button{
    width: 100%;
    margin-right: 40%;
}

.tabela {
    display: flex; /* Define o contêiner como flexível */
    justify-content: center; /* Centraliza horizontalmente */
    height: 100vh; /* Garante que ocupa a altura total da tela */
    max-height: 450px;
}

.form3{
    background-color: var(--bloquinho);
    height: 450px;
    padding: 20px;
    display: flex;
    flex-direction: row;
    border-radius: 7px;
    border: solid var(--borda);
    opacity: 90%;
    flex-wrap: wrap;
    justify-items: center;
    overflow: auto; /* Permite rolar caso o conteúdo seja maior */
}

.form3 a{
    text-decoration: none;
    color: var(--borda);
    margin-left: 40px;
}

.back-button{
    z-index: 100;
    text-decoration: none;

```



```
    color: var(--borda);  
    position: fixed;  
    align-self: flex-end;  
  }  
  .form2 .message{  
    z-index: -10;  
    margin-top: -45px;  
    margin-bottom: 60px;  
  }
```