



Projeto LUMINI - Tecnocore

(25/02/2025)

Relatório de Requisitos

Requisitos Funcionais

Página Inicial

Introdução à Loja

- **Funcionalidade:** A página inicial deve apresentar uma seção que introduza a loja ao usuário, com uma breve descrição sobre a marca e seus valores, destacando o foco em roupas femininas de academia.
- **Objetivo:** Fornecer uma visão clara sobre o propósito e a identidade da loja, destacando o compromisso com a qualidade, estilo e conforto das roupas voltadas para o público feminino que pratica atividades físicas.

Exibição de Produtos em Destaque

- **Funcionalidade:** A página inicial deve exibir uma seção de produtos em destaque, com até 5 peças selecionadas de roupas femininas de academia.
- **Informações Exibidas:** Para cada produto, a página deve exibir a imagem do produto, o nome, uma breve descrição e o preço.
- **Objetivo:** Destacar produtos selecionados para promover itens populares ou novos da coleção, atraindo o interesse imediato do usuário e incentivando a compra.

Área de Promoções

- **Funcionalidade:** A página deve incluir uma área dedicada às promoções ativas da loja, com até 3 ofertas em destaque, focadas em roupas e acessórios femininos de academia.
- **Exibição das Promoções:** As promoções devem ser destacadas com banners visíveis e links diretos para as páginas dos produtos ou categorias em promoção.

- **Objetivo:** Informar o usuário sobre as ofertas especiais, incentivando ações rápidas, como a compra de produtos com desconto ou em promoção, especialmente itens da categoria fitness.

Navegação Intuitiva

- **Funcionalidade:** A navegação na página inicial deve ser clara e intuitiva, permitindo que o usuário acesse rapidamente outras partes do site, como categorias de produtos, página de promoções e outras coleções de roupas femininas de academia.
- **Experiência do Usuário:** O layout e os elementos da página devem ser organizados de forma lógica, proporcionando uma experiência de navegação fluida e sem esforço, especialmente voltada para o público feminino.
- **Objetivo:** Garantir que o usuário possa explorar a loja facilmente, encontrando rapidamente o que deseja, como roupas de treino, acessórios e coleções específicas.

Desempenho e Acessibilidade

- **Funcionalidade:** A página inicial deve carregar rapidamente e ser acessível em diferentes dispositivos e tamanhos de tela, como desktop, tablets e smartphones.
- **Desempenho:** A página deve ser otimizada para garantir tempos de resposta rápidos, mesmo ao carregar imagens de produtos e banners de promoções.
- **Acessibilidade:** O design da página deve ser acessível, garantindo que todos os usuários, incluindo pessoas com deficiências, possam navegar sem dificuldades, respeitando as diretrizes de acessibilidade.

Atualização de Conteúdo

- **Funcionalidade:** O conteúdo da página inicial, como produtos em destaque e promoções, deve ser atualizado automaticamente de acordo com a disponibilidade de novos produtos ou ofertas.
- **Objetivo:** Garantir que o usuário sempre veja informações relevantes e atuais na página inicial, refletindo as últimas novidades e promoções de roupas femininas de academia.

Catálogo de Produtos

- **Exibição de Produtos:** Imagens de Alta Qualidade: As imagens dos produtos devem ser nítidas e detalhadas, permitindo ao usuário zoom para ver detalhes (exemplo: costura, tecido).

Informações Visíveis:

- **Preço:** Mostrar preço atual, com destaque para descontos, se houver.
 - **Disponibilidade:** Exibir se o produto está em estoque ou fora de estoque.
 - **Especificações Técnicas:** Detalhes como material, tipo de tecido, tamanhos disponíveis e cores.
- **Descrição Detalhada do Produto:** Características: Explicar o material e o que o torna adequado para treinos de academia (ex: "tecido respirável", "elasticidade"). Benefícios: Descrever como o produto pode melhorar a experiência do usuário (ex: "proporciona conforto durante o exercício", "ajuda na mobilidade"). Diferenciais: Informar recursos exclusivos (ex: "tecnologia anti-odor", "secagem rápida").
- **Avaliações e Comentários:** Nota de Avaliação: Mostrar a nota média do produto em estrelas (1 a 5) dada pelos clientes.
Comentários de Clientes: Exibir as opiniões de quem comprou o produto, com a data de cada comentário.
Filtro de Avaliações: Permitir que os usuários filtrem avaliações por mais recentes ou mais úteis.
- **Filtros e Ordenação:** Filtros Disponíveis:
 - **Preço:** Ordenar por preço (baixo para alto ou alto para baixo).
 - **Tamanho:** Mostrar apenas produtos disponíveis no tamanho selecionado.
 - **Cor:** Permitir selecionar a cor desejada do produto.
 - **Avaliação:** Exibir produtos com melhores avaliações.

Ordenação: Permitir ordenar os produtos por popularidade, preço ou avaliação.
- **Responsividade:** O catálogo deve ser adaptável a dispositivos móveis, com um design que se ajuste automaticamente ao tamanho da tela (desktop, tablet, celular). A navegação deve ser fácil e rápida, com imagens carregando de forma fluída, mesmo em dispositivos móveis.
- **Zoom nas Imagens:** O usuário deve poder ampliar as imagens do produto ao passar o mouse (em desktop) ou ao clicar/tocar na imagem (em dispositivos móveis) para ver detalhes mais de perto.

- **Outras Funcionalidades:** Tamanhos Disponíveis: Mostrar os tamanhos disponíveis para o produto, com a opção de escolher o tamanho antes de adicionar ao carrinho. Notificação de Estoque: Se um produto estiver esgotado, oferecer a opção de aviso quando disponível ou mostrar o tempo estimado de reposição.

Filtros de Produtos

- Filtros Disponíveis : O sistema deve permitir a filtragem de produtos com base nos seguintes critérios:
 - 1) **Marca:** Filtragem por marca específica dos produtos.
 - 2) **Tamanho:** Filtro de acordo com os tamanhos disponíveis dos produtos.
 - 3) **Preço:** Faixa de preço, permitindo filtrar produtos por valores mínimos e máximos.
 - 4) **Categoria:** Filtragem por categoria do produto (por exemplo, roupas, eletrônicos, acessórios, etc.).

Seleção dos Filtros:

- O sistema deve permitir a **seleção individual** ou **múltipla** de critérios de filtragem.
- Deve ser possível combinar diferentes filtros para refinar a busca, como, por exemplo, filtrar por "marca X" e "tamanho Y" simultaneamente.

Atualização Dinâmica

- A aplicação deve atualizar automaticamente os resultados dos produtos **em tempo real**, sem a necessidade de recarregar a página.
- A filtragem deve ser aplicada instantaneamente assim que o usuário selecionar ou modificar qualquer critério de filtro.

Resultados de Busca

- Caso nenhum produto atenda aos critérios de filtro selecionados, o sistema deve exibir uma mensagem informando que não há resultados correspondentes.
- Os produtos exibidos após a aplicação dos filtros devem refletir os critérios selecionados e ser ordenados de acordo com a lógica estabelecida (por exemplo, por preço, popularidade, etc.).

Facilidade de Navegação

- A funcionalidade de filtro deve ser acessível de forma clara e eficiente. A seleção e combinação de filtros devem ser intuitivas para o usuário.
- O sistema deve ser capaz de processar e exibir rapidamente resultados, mesmo

quando uma combinação complexa de filtros for aplicada.

Desempenho e Escalabilidade

- O sistema deve ser capaz de processar um grande volume de dados de produtos sem comprometer a performance. Isso inclui a filtragem eficiente mesmo com um grande número de produtos cadastrados.

Provedor Virtual

• **Cadastro e Inserção de Medidas Pessoais:**

- Permitir que o usuário insira e atualize suas medidas pessoais (
 - altura,
 - peso,
 - cintura,
 - quadril,
 - peitoral.)
- O usuário pode optar por salvar essas medidas em seu perfil para futuras compras.
- O sistema deve permitir que o usuário edite ou exclua as medidas salvas, com a opção de selecionar medidas salvas em futuras compras.

• **Tabelas de Medidas:**

Objetivo: Garantir que as tabelas de medidas sejam atualizadas periodicamente para refletir mudanças nos produtos, se houver.

- O sistema deve ter tabelas de medidas associadas a cada produto, contendo informações como tamanhos (P, M, G, GG), variações de modelo (slim fit, regular fit, etc.) e ajustes específicos de cada produto.
- Os administradores devem ter a capacidade de adicionar, atualizar e excluir as informações das tabelas de medidas dos produtos conforme necessário.
- Administradores do sistema devem ser capazes de atualizar as tabelas de tamanhos e ajustes dos produtos, incluindo informações sobre diferentes variações de modelo (ex: slim fit, regular fit, etc.)

• **Integração de informações fornecidas pelos usuários com as tabelas de medidas dos produtos para sugerir o tamanho ideal.**

Objetivo: Integrar as medidas pessoais do usuário com as tabelas de tamanhos dos produtos.

- O sistema deve usar as informações fornecidas pelo usuário (medidas pessoais) e compará-las com as tabelas de medidas dos produtos para sugerir o tamanho ideal.
- A sugestão de tamanho deve levar em consideração as variações do modelo do produto (ex: slim fit, regular fit) e o ajuste específico de cada item.
- O sistema deve exibir a sugestão de tamanho de forma clara e visível para o usuário durante o processo de seleção de produto.

Carrinho de Compras

1. Funcionalidades Principais

Adicionar itens ao carrinho

- O usuário deve poder adicionar peças de roupa ao carrinho a partir da página de listagem de produtos ou da página de detalhes do produto.
- O sistema deve verificar a disponibilidade do item em relação ao estoque e ao tamanho selecionado antes da adição.
- Caso o item já esteja no carrinho com o mesmo tamanho e cor, a quantidade deve ser aumentada, e o subtotal recalculado.
- Deve haver um limite máximo de itens por produto, conforme a disponibilidade em estoque.
- O sistema deve exibir uma mensagem de confirmação da adição bem-sucedida.

Remover itens do carrinho

- O usuário deve poder remover uma peça de roupa específica do carrinho.
- O sistema deve exibir uma mensagem de confirmação antes da remoção.
- Caso o usuário confirme a remoção, o item será excluído e o total do carrinho atualizado automaticamente.
- Caso o usuário remova o último item do carrinho, o sistema deve exibir uma mensagem sugerindo que ele continue comprando.

Atualizar quantidade de itens no carrinho

- O usuário deve poder modificar a quantidade de cada peça no carrinho diretamente na interface.
- O sistema deve impedir a seleção de uma quantidade superior ao estoque disponível para o tamanho e cor escolhidos.
- Se a quantidade for alterada, o subtotal do item e o total do carrinho devem ser recalculados imediatamente.

- Caso o usuário tente reduzir a quantidade para zero, o sistema deve exibir uma mensagem de confirmação antes de remover o item.

Exibição de informações no carrinho

- O carrinho deve exibir para cada peça de roupa:
 - Nome do produto
 - Imagem ilustrativa
 - Tamanho selecionado
 - Cor escolhida (se aplicável)
 - Preço unitário
 - Quantidade selecionada
 - Subtotal (quantidade x preço unitário)
- O valor total da compra deve ser exibido, considerando todos os itens e possíveis descontos.
- O sistema deve destacar peças com baixa disponibilidade em estoque.
- Caso uma peça fique indisponível antes da finalização da compra, o sistema deve notificar o usuário e removê-la automaticamente.

2. Cálculo e Atualização em Tempo Real

Recalcular totais dinamicamente

- Toda ação de adicionar, remover ou modificar a quantidade de uma peça deve acionar um recálculo automático do total da compra.
- O total deve incluir possíveis impostos, taxas de entrega e descontos aplicáveis.

Aplicação de descontos e promoções

- Se houver cupons de desconto, o usuário poderá aplicá-los no carrinho, e o desconto será calculado em tempo real.
- O sistema deve verificar a validade do cupom antes da aplicação.
- O valor atualizado deve ser refletido imediatamente após a aplicação do cupom.

3. Mensagens de Confirmação e Notificações

Mensagens ao usuário

- Toda ação relevante no carrinho deve ser acompanhada por uma mensagem visual e sonora (se aplicável), incluindo:
 - "Item adicionado ao carrinho com sucesso."
 - "Item removido do carrinho."
 - "Quantidade atualizada."
 - "Desculpe, este item está fora de estoque."

- "Você atingiu o limite máximo de compra para este item."
- "Cupom aplicado com sucesso!" / "Cupom inválido ou expirado."

Mensagens de erro

- O sistema deve informar o usuário caso uma ação não possa ser realizada, com mensagens claras e instruções sobre como proceder.

4. Persistência e Recuperação do Carrinho

Salvar estado do carrinho para retomada posterior

- O carrinho deve ser salvo automaticamente no dispositivo do usuário (cookies/local storage) ou na conta do usuário (banco de dados) para que a compra possa ser retomada posteriormente.
- Caso o usuário esteja autenticado, o carrinho deve ser armazenado no servidor para permitir acesso a partir de diferentes dispositivos.
- Se o usuário retornar ao site posteriormente, o sistema deve recuperar o carrinho salvo e exibir um aviso informando que há itens pendentes.
- O sistema deve garantir que as peças ainda estejam disponíveis antes de restaurá-las no carrinho. Caso contrário, exibir uma notificação sobre a indisponibilidade.

Opção de limpar o carrinho

- O usuário deve poder esvaziar o carrinho manualmente por meio de um botão "Esvaziar Carrinho".
- Antes da exclusão, deve haver uma confirmação para evitar remoções acidentais.

5. Experiência do Usuário e Interface

Design responsivo e intuitivo

- O carrinho deve ser de fácil acesso, com um ícone visível na barra de navegação indicando a quantidade de itens.
- A interface deve ser responsiva e otimizada para dispositivos móveis e desktop.
- O usuário deve poder acessar um resumo do carrinho sem sair da página atual.
- A página do carrinho deve ter um botão claro para "Finalizar Compra".

Cadastro e Login de Usuário

Cadastro de Usuário

- **Funcionalidade:** O sistema deve permitir que os usuários realizem seu cadastro utilizando as seguintes informações obrigatórias: nome completo, CPF, número de

telefone, e-mail e senha.

- **Validação:** O CPF e o e-mail fornecidos devem ser únicos no sistema, não permitindo duplicação.
- **Segurança:** As senhas devem ser armazenadas de forma segura, utilizando técnicas adequadas de criptografia.
- **Notificação de Sucesso/Erro:** Após o cadastro, o sistema deve notificar o usuário sobre o sucesso ou falha do processo, indicando o motivo da falha, caso haja.

Login de Usuário

- **Funcionalidade:** O sistema deve permitir que o usuário faça login utilizando seu e-mail ou CPF e senha.
- **Autenticação:** O sistema deve validar as credenciais de login antes de conceder acesso ao usuário.
- **Segurança:** O sistema deve garantir que as senhas sejam verificadas de maneira segura e que o login seja realizado através de sessões seguras (utilizando HTTPS, por exemplo).

Recuperação de Senha

- **Funcionalidade:** O sistema deve permitir que o usuário recupere sua senha caso a tenha esquecido.
- **Métodos de Recuperação:** O usuário poderá optar por recuperação de senha via e-mail ou número de telefone.
- **Validação de CPF:** Antes do envio do código de recuperação, o sistema deve solicitar e validar o CPF do usuário para garantir a autenticidade da solicitação.
- **Notificação:** O usuário deve ser notificado por e-mail ou SMS sobre o envio do código de recuperação.

Página de Conta do Usuário

- **Funcionalidade:** Após o login, o usuário deve ser redirecionado para uma página de conta, onde poderá visualizar e gerenciar suas informações pessoais, como nome, e-mail, telefone, e senha.
- **Edição de Dados:** O sistema deve permitir que o usuário edite suas informações pessoais de forma segura.
- **Segurança:** O sistema deve garantir que apenas o próprio usuário tenha acesso à edição de suas informações, através de autenticação e autorização apropriadas.

Área do Usuário

1. Acesso à Área do Usuário

- 0 O sistema deve oferecer uma área exclusiva para o usuário, acessível apenas mediante **autenticação com e-mail e senha**.
- 0 O acesso deve ser protegido por **sessão segura** com expiração automática após **30 minutos de inatividade**.
- 0 O sistema deve permitir a recuperação de senha via e-mail caso o usuário esqueça a credencial.

2. Histórico de Pedidos

- 0 O usuário poderá visualizar um histórico completo de suas compras.
- 0 Cada pedido deve exibir as seguintes informações:
 - ▮ **Número do pedido**
 - ▮ **Data da compra**
 - ▮ **Status do pedido** (Exemplo: "Aguardando Pagamento", "Pago", "Enviado", "Entregue", "Cancelado")
 - ▮ **Lista de produtos adquiridos**, incluindo nome, quantidade e valor unitário
 - ▮ **Valor total pago**, incluindo frete e possíveis descontos
- 0 O usuário poderá filtrar pedidos por **data** e **status** para facilitar a busca.
- 0 O sistema deve permitir que o usuário **baixe o comprovante de pagamento** ou **nota fiscal** (se disponível).

3. Gerenciamento de Dados Pessoais

- 0 O usuário poderá atualizar as seguintes informações em sua conta:
 - ▮ **Nome completo**
 - ▮ **Número de telefone**
 - ▮ **E-mail**
 - ▮ **Senha**
- 0 Para qualquer alteração de e-mail ou senha, o sistema deve exigir a **inserção da senha atual** como medida de segurança.
- 0 Caso o e-mail seja alterado, o sistema deve enviar um **código de verificação** para o novo e-mail antes de confirmar a mudança.
- 0 Caso o usuário esqueça sua senha, o sistema deve oferecer a opção de redefinição via **e-mail cadastrado**.

0

4. Segurança e Proteção de Dados

- Todas as informações sensíveis, como senha e dados pessoais, devem ser armazenadas utilizando **criptografia segura (exemplo: bcrypt para senhas)**.
- A comunicação entre o usuário e o sistema deve ser protegida por **HTTPS/TLS** para evitar interceptação de dados.
- O sistema deve implementar **proteção contra ataques de força bruta**, bloqueando tentativas de login após **5 falhas consecutivas**, com desbloqueio após **15 minutos** ou por verificação via e-mail.
- A sessão do usuário deve ser encerrada automaticamente após **30 minutos de inatividade** para aumentar a segurança.

Opções de Pagamento

1. Meios de Pagamento Disponíveis

O sistema deve oferecer as seguintes formas de pagamento para os usuários:

- Pix
- Boleto bancário
- Cartão de crédito (com parcelamento de até 12x sem juros para compras acima de R\$ 100,00 e até 3x sem juros para compras abaixo desse valor)

2. Pagamento via Pix

- O sistema deve gerar um **QR Code dinâmico** que o usuário pode escanear para realizar o pagamento.
- O QR Code deve ter um tempo de validade de **1 hora** após a geração.
- O sistema também deve disponibilizar uma **chave Pix copia e cola** para facilitar a transação.
- Após a confirmação do pagamento pelo banco, o status do pedido deve ser atualizado automaticamente para **"Pagamento Aprovado"** dentro de **30 segundos**.
- Se o pagamento não for realizado dentro do prazo de 1 hora, o QR Code deve ser invalidado, e o usuário precisará gerar um novo.

3. Pagamento via Boleto Bancário

- O sistema deve gerar um boleto bancário com vencimento em **3 dias úteis** a partir da data de emissão.

- 0 O usuário poderá reemitir o boleto somente se ele já estiver vencido e o pedido ainda não tiver sido cancelado.
- 0 A compensação do pagamento pode levar até **2 dias úteis**. O sistema deve atualizar o status do pedido para "**Pagamento Aprovado**" assim que a compensação for confirmada.
- 0 Se o boleto não for pago até a data de vencimento, o status do pedido deve ser atualizado automaticamente para "**Cancelado**".

4. Pagamento via Cartão de Crédito

- 0 O sistema deve aceitar pagamentos com cartões das bandeiras **Visa, Mastercard, Elo e American Express**.
- 0 O parcelamento seguirá as seguintes regras:
 - Para compras **acima de R\$ 100,00**, será permitido parcelamento em **até 12 vezes sem juros**.
 - Para compras **entre R\$ 50,00 e R\$ 99,99**, será permitido parcelamento em **até 3 vezes sem juros**.
 - Para compras **abaixo de R\$ 50,00**, o parcelamento **não será permitido**.
- 0 O sistema deve exibir para o usuário, antes da finalização da compra:
 - O **valor total da compra**
 - O **valor das parcelas**
 - O **número máximo de parcelas disponíveis**, conforme as regras acima
- 0 O sistema deve validar os seguintes dados do cartão antes de processar a transação:
 - Número do cartão (BIN válido e estrutura correta)
 - Data de validade (não pode estar vencida)
 - Código de segurança (CVV de 3 ou 4 dígitos, conforme a bandeira)
 - Nome do titular (deve corresponder ao cadastrado no banco emissor)
- 0 A resposta da transação deve ser imediata e pode ter os seguintes status:
 - **Aprovado**: O pagamento foi autorizado pelo banco. O pedido será processado.
 - **Recusado**: O pagamento foi negado pelo banco. O usuário deve ser informado do motivo.
 - **Erro na transação**: Houve falha na comunicação com a operadora do cartão. O usuário pode tentar novamente.

5. Confirmação e Status do Pagamento

- O sistema deve validar todas as informações de pagamento antes de processar a transação.
- Após a finalização do pagamento, o usuário deve receber uma **notificação em tempo real** com um dos seguintes status:
 - **Pagamento Aprovado:** O pedido será confirmado e processado.
 - **Pagamento Pendente:** O pagamento está em análise (casos de boleto ou revisão manual do cartão de crédito).
 - **Pagamento Recusado:** O usuário deve tentar outro meio de pagamento.
- Em caso de falha no pagamento, o sistema deve exibir mensagens de erro específicas, como:
 - “Saldo insuficiente”
 - “Cartão recusado pelo emissor”
 - “Erro na comunicação com a instituição financeira”
- O usuário poderá acessar o histórico de pagamentos e acompanhar o status pelo painel de pedidos.

6. Regras e Restrições

- O parcelamento sem juros será aplicado conforme as regras mencionadas na seção do cartão de crédito.
- O sistema deve aceitar apenas cartões de crédito emitidos no Brasil.
- Caso um pagamento via Pix ou boleto não seja compensado dentro do prazo estipulado, o pedido será automaticamente **cancelado**.

Endereço de Entrega

Objetivo:

Permitir que o sistema colete e valide o endereço de entrega do cliente, incluindo o CEP, para garantir que a entrega dos produtos de vestuário seja feita corretamente. O sistema deve validar a entrada, integrar com APIs externas (se necessário), e permitir a atualização de informações ao longo do tempo.

Descrição:

1. **Campos obrigatórios:** O cliente deve preencher todos os campos obrigatórios de endereço de entrega:
 - Nome completo
 - Rua
 - Número
 - Complemento
 - Bairro
 - Cidade
 - Estado
 - CEP (obrigatório)
2. **Validação:** O sistema deve validar a entrada desses campos, garantindo que eles sejam preenchidos corretamente antes de permitir o envio.
3. **Formato de CEP:** O CEP deve ser inserido no formato '**XXXXX-XXX**' (ex: 12345-678). Caso o usuário insira o CEP em outro formato, o sistema deve alertá-lo para que corrija o formato.
4. **Validação do CEP:** O sistema deve validar o CEP inserido, realizando uma verificação no formato correto e também consultando uma API externa (como os Correios ou serviços de CEP) para garantir que o CEP informado seja válido e que a cidade e o bairro correspondam ao CEP.
5. **Exibição de Mensagem de Erro:** Caso o usuário insira um CEP mal formatado ou inválido (não encontrado na API de CEP), o sistema deve exibir uma mensagem clara de erro pedindo para o usuário corrigir o CEP. A mensagem pode ser algo como:
 - "CEP inválido. Por favor, insira um CEP no formato '**XXXXX-XXX**' e válido."
6. **Integração com API de CEP:** Após a validação do CEP, o sistema deve integrar-se com uma API externa de consulta de CEP para preencher automaticamente os campos de **cidade** e **bairro**, baseando-se na resposta da API.
7. **Armazenamento de Endereço:** Após a validação completa dos campos, o endereço de entrega completo (incluindo o CEP) deve ser armazenado no banco de

dados, vinculado ao perfil do cliente. Caso o cliente já tenha um endereço salvo, este deve ser atualizado com as novas informações fornecidas.

8. **Atualização de Endereço:** O cliente deve poder atualizar seu endereço de entrega a qualquer momento. O sistema deve garantir que:

- O novo endereço será validado e armazenado corretamente.
- Caso haja uma alteração de endereço, o sistema deve garantir que o histórico de pedidos esteja atualizado, utilizando o novo endereço para as futuras entregas.

Painel Administrativo

Objetivo:

O sistema deve fornecer um painel administrativo robusto para que os administradores possam gerenciar o **estoque**, os **pedidos** e os **clientes**. O painel deve permitir o **controle de produtos**, a **visualização e atualização dos status dos pedidos**, e a **gestão de clientes**, possibilitando a geração de **relatórios e filtros** para facilitar a administração.

Descrição Detalhada das Funcionalidades:

1. Gerenciamento de Produtos:

O painel deve permitir que os administradores adicionem novos produtos(exemplo: camisas,calças,bermudas,vestidos,etc..), editem os existentes, removam produtos quando necessário e atualizem a quantidade de estoque. O sistema deve ser capaz de organizar os produtos por **categorias** (exemplo: camisetas, calças, calçados, vestidos), e as funcionalidades devem ser aplicáveis a todos os produtos.

Funções:

- ▮ **Adicionar Produto:** O administrador deve poder adicionar um novo produto, fornecendo detalhes como nome, categoria, cores, tamanhos, preço e quantidade em estoque.
- ▮ **Editar Produto:** O administrador pode atualizar informações como preço, quantidade, descrição ou cores/tamanhos disponíveis.

- ▮ **Remover Produto:** O administrador pode remover um produto do catálogo.
- ▮ **Notificação de Estoque Baixo:** Quando a quantidade de um produto ficar abaixo de um valor mínimo configurado, o sistema deve gerar uma notificação de estoque baixo.

2. Gerenciamento de Pedidos:

O painel deve permitir aos administradores visualizar, atualizar e gerenciar os pedidos de clientes. O administrador pode alterar o status dos pedidos conforme o progresso da entrega.

Status de Pedido:

- ▮ **Pendente:** O pedido foi realizado, mas ainda não foi processado.
- ▮ **Em Processo:** O pedido está sendo preparado para envio.
- ▮ **Enviado:** O pedido foi enviado para o cliente.
- ▮ **Entregue:** O pedido foi entregue ao cliente.
- ▮ **Cancelado:** O pedido foi cancelado por solicitação do cliente ou por falha na entrega.

Funções:

- ▮ **Visualizar Detalhes do Pedido:** O administrador pode visualizar o histórico completo do pedido, incluindo produtos, quantidades, status de envio, e dados do cliente.
- ▮ **Alterar Status do Pedido:** O administrador pode alterar o status do pedido conforme o andamento da entrega.
- ▮ **Filtrar Pedidos:** O painel deve permitir a filtragem de pedidos por status, data, valor ou cliente.

3. Gerenciamento de Clientes:

O painel deve exibir informações detalhadas sobre os clientes, permitindo o acesso rápido ao histórico de pedidos e a edição de dados de contato e endereços.

Funções:

- ▮ **Visualizar Informações do Cliente:** O administrador pode visualizar o nome, endereço, e histórico de pedidos de um cliente.
- ▮ **Editar Informações do Cliente:** O administrador pode editar o nome, endereço de entrega e outros dados do cliente, caso necessário.
- ▮ **Pesquisar Clientes:** O painel deve permitir a pesquisa de clientes por nome ou outros critérios.

4. Geração de Relatórios:

O painel administrativo deve permitir a geração de relatórios detalhados com base em diversos filtros, como pedidos por status, pedidos por data, e pedidos por valor ou cliente.

Funções:

- **Filtrar Relatórios por Status:** O administrador pode gerar um relatório filtrando os pedidos por status (pendente, em processo, enviado, etc.).
- **Filtrar Relatórios por Data:** O administrador pode gerar relatórios filtrando os pedidos por intervalo de datas.
- **Filtrar Relatórios por Valor:** O administrador pode gerar relatórios filtrando os pedidos por faixa de valor.
- **Filtrar Relatórios por Cliente:** O administrador pode gerar relatórios filtrando os pedidos de um cliente específico.
- **Exportação de Relatórios:** O painel deve permitir a exportação dos relatórios gerados para formatos como PDF ou CSV.

Requisitos Não Funcionais

Escalabilidade

- O sistema deve ser estruturado para suportar um grande número de usuários simultaneamente, garantindo a manutenção da performance mesmo em momentos de alta demanda.
- A arquitetura do sistema deve permitir a expansão de recursos conforme a necessidade, garantindo um desempenho estável.
- Deve ser possível realizar balanceamento de carga para distribuir o tráfego entre diferentes servidores, evitando sobrecarga em momentos de pico.
- O banco de dados deve ser otimizado para operações simultâneas, garantindo tempos de resposta adequados mesmo com alto volume de acessos.
- Tecnologias e práticas de caching devem ser adotadas para reduzir a necessidade de requisições repetitivas ao servidor, melhorando a eficiência do sistema.
- O sistema deve contar com ferramentas de monitoramento e autoescala, ajustando automaticamente os recursos conforme a demanda para evitar quedas de desempenho.
- Deve possuir tolerância a falhas, permitindo a continuidade da operação mesmo em caso de falha de um servidor ou componente crítico.

Segurança

- O sistema deve implementar medidas de segurança robustas para garantir a proteção das informações pessoais e financeiras dos clientes, prevenindo acessos não autorizados e vazamentos de dados.
- Os dados sensíveis, como senhas e informações financeiras, devem ser protegidos por criptografia forte tanto em repouso quanto em trânsito.
- A autenticação de usuários deve incluir políticas de senhas seguras, exigindo um mínimo de complexidade e troca periódica quando necessário.
- Deve ser implementada proteção contra ataques comuns, como SQL Injection, Cross-Site Scripting (XSS) e Cross-Site Request Forgery (CSRF).
- O sistema deve contar com autenticação em duas etapas (2FA) como opção adicional para reforçar a segurança do login.
- Os acessos administrativos devem ser controlados por meio de gestão de permissões e logs de auditoria, garantindo rastreabilidade e prevenção de ações mal-intencionadas.
- As transações financeiras devem ser protegidas por certificados SSL/TLS para garantir a integridade e a confidencialidade das informações.
- Deve ser implementado um sistema de monitoramento contínuo, detectando atividades suspeitas e notificando administradores em caso de tentativa de invasão ou comportamento anômalo.

Design Responsivo

- A plataforma será desenvolvida de forma a garantir total compatibilidade e usabilidade em diferentes dispositivos, como smartphones, tablets e desktops.
- A interface do sistema deve se ajustar automaticamente ao tamanho da tela, proporcionando uma experiência consistente e fluida em dispositivos móveis, tablets e desktops.
- O design deverá priorizar a usabilidade, garantindo que os elementos da interface sejam acessíveis e legíveis em qualquer dispositivo.
- Deve ser utilizado design adaptativo para otimizar o layout e os recursos, assegurando a melhor experiência de navegação em diferentes plataformas.
- A plataforma deve ser otimizada para suportar diferentes sistemas operacionais e navegadores, assegurando que o usuário não enfrente dificuldades ao acessar o sistema de qualquer dispositivo.

Desempenho

- O sistema deve ser projetado para garantir um desempenho rápido e eficiente, mesmo com uma alta carga de usuários e dados.

- O tempo de resposta das páginas deve ser inferior a 2 segundos, garantindo uma navegação fluida para o usuário.
- O sistema deve implementar otimização de consulta ao banco de dados para minimizar a latência nas transações.
- Deve ser utilizada compressão de imagens e recursos estáticos para reduzir o tempo de carregamento das páginas.
- O sistema deve ser capaz de processar um alto volume de transações simultâneas sem degradação significativa do desempenho.
- Deve ser realizado testes de carga e stress periodicamente para garantir que o sistema aguente volumes de tráfego inesperados.

Backup e Recuperação de Dados

- O sistema deve contar com um processo robusto de backup e recuperação de dados para garantir a integridade das informações.
- Os dados críticos, como informações de clientes e transações, devem ser automaticamente backupados em intervalos regulares.
- O sistema deve permitir a recuperação de dados em caso de falha, garantindo que as informações possam ser restauradas rapidamente.
- O processo de backup deve ser seguro, com criptografia dos dados para evitar acesso não autorizado.
- O sistema deve contar com uma estratégia de backup incremental, onde apenas as mudanças desde o último backup são armazenadas, otimizando o tempo e o armazenamento.
- A recuperação de dados deve ser testada periodicamente para garantir sua eficácia em situações de emergência.

Acessibilidade

- A plataforma deve ser acessível para todos os usuários, incluindo aqueles com deficiências, garantindo que o sistema possa ser utilizado por qualquer pessoa.
- O design da interface deve seguir as diretrizes de acessibilidade, como a utilização de contraste adequado entre texto e fundo, tamanhos de fonte ajustáveis e navegação por teclado.
- O sistema deve ser compatível com leitores de tela, garantindo que pessoas com deficiência visual possam navegar e interagir com o site.
- Deve ser implementada uma função de tradução de conteúdo para garantir que a plataforma seja acessível em diferentes idiomas.
- O sistema deve ser testado periodicamente com ferramentas de acessibilidade

para garantir conformidade com as normas de acessibilidade da WCAG (Web Content Accessibility Guidelines).