Checklist de Desenvolvimento: Frete (Melhor Envio)

Parte 1: Backend - Banco de Dados e Configuração

- 1.1: Revisar Tabela `variants`
  - Confirmar que a tabela variants possui uma coluna weight_g (peso em gramas) para cada variação de produto. Se não, criar uma nova migração para adicioná-la.
- 1.2: Criar Tabela para Cache de Cotações
  - Criar um novo arquivo de migração SQL (..._create_shipping_quotes_table.sql).
  - Definir uma tabela shipping_quotes para armazenar em cache os resultados da API de frete.
    - Colunas Sugeridas: cep (text), total_weight_g (integer), quotes (jsonb), created_at (timestamptz).
    - Chave Primária: Composta por (cep, total_weight_g).
- 1.3: Configurar Variáveis de Ambiente
  - Adicionar as seguintes chaves ao arquivo .env.local:

1 # Token de acesso para a API do Melhor Envio 2 MELHOR_ENVIO_API_TOKEN=SEU_TOKEN_AQUI 3 4 # Subtotal mínimo para ativar a regra de frete grátis 5 FRETE_GRATIS_SUBTOTAL=249.90 6 7 # Lista de CEPs (separados por vírgula) elegíveis para retirada em loja 8 LOJA_CEPS_RETIRADA="01001000,01002000"

Parte 2: Backend - Rota de API (`/api/frete/cotacao`)

- 2.1: Criar Estrutura da Rota
  - Criar o arquivo da rota: src/app/api/frete/cotacao/route.ts.
  - Definir um schema Zod para validar o corpo da requisição, que deve conter cep (string) e items (array de objetos com variant_id e quantity).
- 2.2: Implementar Lógica de Cálculo e Regras
  - Na função POST, buscar os dados das variantes (price, weight_g) no banco a partir dos items recebidos.
  - Calcular o subtotal e o total_weight_g do carrinho.
  - Implementar a regra de Frete Grátis: Se o subtotal for maior que FRETE_GRATIS_SUBTOTAL, adicionar uma opção de frete grátis à resposta.

    - Implementar a regra de Retirada em Loja: Se o cep do cliente estiver na lista LOJA_CEPS_RETIRADA, adicionar esta opção à resposta.
- 2.3: Implementar Lógica de Cache
    - Antes de chamar a API externa, consultar a tabela shipping_quotes com o cep e total_weight_g.
    - Se um resultado válido (criado há menos de 30 minutos) for encontrado, retorná-lo diretamente.
- 2.4: Integrar com a API do Melhor Envio
    - Se não houver cache válido, montar a requisição para a API do Melhor Envio, enviando os dados necessários (CEP de origem/destino, peso total, etc.).
    - Executar a chamada fetch para a API do Melhor Envio.
- 2.5: Processar e Retornar a Resposta
    - Da resposta do Melhor Envio, filtrar e extrair as 2 melhores cotações (ex: mais barata e mais rápida).
    - Salvar o resultado processado na tabela shipping_quotes para futuras requisições.
    - Combinar as cotações da API com as opções de "Frete Grátis" e "Retirada em Loja" (se aplicáveis).
    - Retornar a lista final de opções de frete para o frontend.

Parte 3: Frontend - Integração no Checkout

- 3.1: Modificar a Página de Checkout (`/checkout/page.tsx`)
    - Criar um novo estado para armazenar as opções de frete (ex: shippingOptions).
    - Após o usuário preencher o endereço (no AddressStep), fazer uma chamada fetch para a nova API /api/frete/cotacao.
    - Salvar as opções de frete retornadas no estado shippingOptions.
- 3.2: Atualizar o Passo de Frete (`ShippingStep`)
    - Modificar o componente ShippingStep para receber e exibir as shippingOptions como uma lista de seleção (ex: radio buttons).
    - Cada opção deve mostrar o nome do serviço (ex: "SEDEX"), o prazo de entrega e o preço.
    - Criar um estado para guardar a opção de frete selecionada pelo usuário.
- 3.3: Ajustar o Passo de Pagamento
    - Ao avançar para o passo de pagamento, garantir que o custo do frete selecionado seja incluído no valor total do pedido.
    - Modificar a chamada à API /api/checkout/criar-pedido para enviar o shipping_cost da opção de frete escolhida.