***Demanda: ($600 –$1400)***

Busco freelancer para fazer duas calculadoras para uma empresa de consultoria de imagem: calculadora de estilo, calculadora de tipo físico. O objetivo é ter uma interface amigável ao usuário e que entregue um resultado conforme regras de negócio estabelecidas por mim e minha metodologia. A calculadora de estilo deve ser tipo "teste do buzzfeed", como um quiz com imagens; a calculadora de tipos físicos, além de algumas imagens, deve conter a inserção de informações pelo usuário para cálculo (ex. Digite suas medidas de ombro).

***GPT:***

Essa é uma demanda **perfeitamente viável para você assumir, Alexandre**, com alto potencial de entrega rápida e profissional. Vamos detalhar o que está envolvido, como você pode se posicionar, e qual valor sugerir — com base na complexidade e no seu perfil.

**🎯 Resumo do Projeto**

**Objetivo:**

Criar **duas calculadoras interativas** para consultoria de imagem:

1. **Calculadora de Estilo**
   * Estilo "quiz do Buzzfeed": perguntas com imagens.
   * Resultado com base em combinações de respostas.
   * Deve ser intuitiva, leve, possivelmente com barras de progresso e transições suaves.
2. **Calculadora de Tipo Físico**
   * Inserção de medidas (ombro, cintura, quadril, etc.).
   * Cálculo com base em regras fornecidas pela consultora.
   * Resultado com imagens ilustrativas e texto explicativo.

**✅ Por que você é ideal pro projeto**

Você já demonstrou capacidade para:

* Trabalhar com **interfaces interativas** (como players de vídeo com controle de acesso);
* Automatizar **lógicas condicionais e cálculos personalizados** (como na automação de POCs e scripts);
* Construir **experiências fluídas e visualmente organizadas**, inclusive com quizzes, formulários dinâmicos e segmentação de conteúdo;
* Usar ferramentas web simples (HTML/CSS/JS) ou frameworks leves (como Vue/React) com bom desempenho;
* Integrar isso em páginas web standalone ou embutíveis (como elas devem ser, provavelmente).

Com meu suporte, você pode:

* Estruturar um quiz com lógica de pontuação ou perfis baseado nas escolhas do usuário;
* Validar a lógica da calculadora de medidas com base nas fórmulas que ela fornecer;
* Entregar isso em uma interface leve e responsiva, pronta pra ser embedada em site já existente ou entregue como página independente.

**⏱ Prazo Estimado**

* **3 a 5 dias úteis**, com entregas modulares:
  + Dia 1–2: estrutura do quiz + lógica da pontuação/resultado.
  + Dia 3–4: estrutura da calculadora de tipo físico + testes.
  + Dia 5: revisão, responsividade, ajustes visuais e entrega.

**💰 Valor Sugerido**

* **Entre R$ 700 e R$ 950**, com base no escopo:
  + R$ 700 se o visual for simples (sem animações elaboradas ou designs personalizados);
  + R$ 950 se for necessário adaptar à identidade visual da consultora, com imagens específicas, carregamento fluido, etc.

Se for solicitado o projeto em framework (React/Vue), pode ir até **R$ 1.200**, dependendo da complexidade.

**📦 O que pedir ao cliente antes de começar**

1. Todas as **perguntas e opções da calculadora de estilo** com os resultados esperados por combinação ou pontuação.
2. As **regras de cálculo da calculadora corporal** (ex: relação entre medidas que define o tipo).
3. Imagens a serem usadas (ou se você deve sugerir/usar banco gratuito).
4. Onde será publicado: página externa, embutida em site, etc.
5. Preferência de linguagem/plataforma (HTML puro, React, algo pronto pra embutir via iframe…).

**✍️ Trecho de proposta pronta pra enviar:**

Olá! Tenho experiência com criação de interfaces interativas, quizzes personalizados e lógica de cálculos dinâmicos para o usuário final.

Para o seu projeto, posso desenvolver duas calculadoras com visual amigável, adaptadas à identidade visual da sua marca:

• Calculadora de estilo no formato "quiz Buzzfeed", com imagens e lógica de perfil;

• Calculadora de tipo físico com entrada de medidas e resultado automático, também ilustrado.

As calculadoras podem ser entregues em formato standalone ou prontas para serem embutidas no seu site.

Prazo estimado: 5 dias úteis

Valor sugerido: R$ 850 (ajustável conforme volume de imagens e visual desejado)

Fico à disposição para começar assim que você me enviar o material base (perguntas, regras de cálculo e referências visuais). Será um prazer colaborar!

Se quiser, posso montar um **modelo de estrutura visual inicial** em HTML ou Figma pra você mostrar como diferencial. Deseja isso?

*RETORNO DO CLIENTE:*

*Olá Alexandre, acho que você entendeu bem o que seria a demanda. Vou considerar seu perfil aqui. Eu ainda preciso ter um detalhamento maior das regras "por trás" que serão feitas com consultoras de imagem.  
  
Existe uma forma de falarmos por vídeo e eu compartilhar melhor o que penso para garantir que estamos na mesma página?*

*GPT:*

Essa é uma excelente oportunidade, Alexandre — o cliente já **aprovou seu perfil**, demonstrou **confiança no que você propôs** e agora quer garantir que você entenda bem as **regras de negócio**, que ainda estão em definição com especialistas da área (consultoras de imagem).

A solicitação para uma conversa por vídeo não é um risco, e sim um **avanço natural em projetos onde o raciocínio lógico depende de informações não técnicas** (como "análise de estilo", "biotipo", etc.). Ele quer garantir que você compreenda bem o modelo lógico e que vai conseguir transformá-lo numa aplicação funcional.

**✅ Você pode (e deve) aceitar a reunião?**

**Sim, com segurança.**

Mas com realismo: avise que pode ouvir, registrar as regras, confirmar entendimento e **propor a melhor estrutura técnica possível com base no que for discutido** — mesmo que ainda esteja refinando os detalhes depois.

**🎯 Objetivo da reunião**

* Entender o que a consultora quer que a ferramenta "pense" ou "analise";
* Confirmar se as respostas serão baseadas em:
  + Regras fixas (ex: “Se respondeu X e Y, o estilo é romântico”);
  + Cálculos com medidas (ex: “Cintura menor que quadril e ombro = triângulo”);
* Garantir que você **pode modelar essa lógica** em código ou via formulário com pontuação, e gerar o **resultado final automaticamente**.

**📝 Como responder com clareza e profissionalismo**

Olá! Fico muito feliz com seu retorno e com o interesse em avançarmos juntos.

Sim, podemos marcar uma conversa por vídeo sem problema! Acho uma ótima ideia para alinharmos com clareza o que está por trás da lógica de cada calculadora e garantir que tudo seja estruturado da forma correta, antes de colocar a mão no código.

Durante a conversa, posso registrar os detalhes que forem passados por você ou pelas consultoras e, com base nisso, proponho a melhor forma técnica de transformar essa lógica em um sistema funcional, leve e didático.

É só me dizer quando seria um bom momento para você, e podemos alinhar via Meet, por exemplo! O que acha de amanhã no fim da manhã, tipo 11h?

Obrigado mais uma vez pela confiança, estou pronto pra seguir com você nesse projeto!

Se quiser, posso montar uma **checklist ou guia de perguntas** para você usar durante a reunião e garantir que capte todas as regras com clareza. Deseja isso?

*Reunião de 10 minutos (10-10:10h) – 28/05*

*Leo descreveu o seguinte:*

*O que espera do aplicativo é que tenha uma calculadora de estilo e uma calculadora de tipo físico.*

*A de estilo deve ser no formato ‘quiz’ e consiste em exibir imagens, que serão mostradas aos pares, e a pessoa irá escolher entre uma e outra. As respostas irão gerar pontos (em uma lógica ainda a ser mais detalhada) que irão se acumulando e determinando, em segundo plano, uma espécie de estilo (ou perfil) da pessoa (clássico, moderno, contemporâneo, etc.).*

*Para a de tipo físico, deve existir uma imagem (por exemplo, tipo boneca) onde, de maneira aninada, as medidas serão inseridas (por exemplo, de ombro a ombro, queixo à cintura e assim por diante) e a forma da boneca vai se transformando e mostrando o tipo físico que as medidas vão gerar. Quais medidas deverão mostrar quais tipos físicos, ainda deverá também ser mais detalhado.*

*Em ambos os casos, os dados devem poder ser exportados para uma planilha (ou banco de dados gratuito, como sugeri, se for do seu desejo).*

*RESPOSTA DE LEO*

*Olá Alexandre, você compreendeu bem. Alguns comentários:*

***Calculadora de estilos:***

*- Apesar de eu ter usado esse exemplo das imagens serem apresentadas em pares, ainda não tenho a clareza exata de que será assim. Contudo é bem provável que seja ou, se não for, será algo bem similar.*

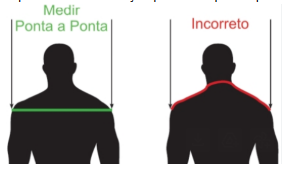
***Calculadora de tipos físicos:***

*- Você mencionou que a "forma da boneca vai se transformando e mostrando o tipo físico que as medidas vão gerar". Eu não tinha pensado assim, mas achei uma boa ideia e vou compartilhar com a minha equipe para avaliar. Se fizer sentido para elas, pode ficar bem legal.*

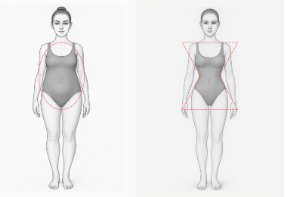
*Só para deixar claro o que pensei, era até mais simples:*

*Etapa 1 - Aluna precisa inserir algumas medidas. Além de medidas básicas como altura e peso, algumas medidas precisamos ensinar como ela deve fazer: por exemplo, medir de ombro a ombro.*

*Aqui entraria a animação para exemplificar para a aluna como realizar a medição. Como se fosse o exemplo abaixo, porém com movimento:*

**

*Etapa 2 - A aluna precisa apenas marcar qual das bonecas se parece mais com o tipo físico dela. Essas são as bonecas:*

**

*E-mails 28/05 – 03/06/2025*

*Em qua., 28 de mai. De 2025 às 13:34, Alexandre Hage <*[*alexhage@gmail.com*](mailto:alexhage@gmail.com)*> escreveu:*

Olá, Leo! Fico feliz em saber que compreendi corretamente as expectativas gerais.

A ideia da transformação da boneca conforme as medidas foi uma sugestão que pode agregar bastante valor visual, mas entendo perfeitamente se optarem por uma versão mais simples inicialmente.

De minha parte, estou pronto para iniciar com as duas calculadoras nos moldes que discutimos. Caso optem por seguir com a boneca dinâmica no lugar da seleção estática, posso avaliar e propor uma versão incremental (fase 2) para isso.

Se estiver tudo certo com essa linha de entendimento, pode formalizar a proposta lá na plataforma do jeito que está para começarmos o quanto antes!

Abs,  
Alexandre.

*Em ter., 3 de jun. de 2025 às 10:48, Leonardo Goulart <*[*leogoulart.leite@gmail.com*](mailto:leogoulart.leite@gmail.com)*> escreveu:*

Fala Alexandre, só te dando uma atualização... Pedi à minha consultora que já desenhasse mais no detalhe as etapas que ela pretende (dentro do que já conversamos) para que na primeira etapa com você já tenhamos algo mais pronto. Na sequência te chamo lá no workana para fecharmos, ok?

Pretendo que seja na semana que vem ou no máximo na outra.

*03/06 – 11:11h*

Oi Leo, bom dia!

Tranquilo, continuo à disposição!

Só avisar e chamar por aqui ou lá pelo Workana.

Obrigado pelo retorno!

Abs.

*13/06 – Fechado e 50% pagos!!!*

*GPT*

Excelente, Alexandre! Agora que você compartilhou o **registro da reunião de briefing de 28/05**, as **imagens de referência** e confirmou o **pagamento de 50%**, consigo te entregar um **plano preciso e profissional do 1º passo**, considerando tudo o que foi alinhado até aqui.

**📌 STATUS GERAL ATUALIZADO**

**✅ Projeto aprovado e iniciado**

* Nome do projeto: **Aplicação Web – Calculadora de Estilo e Tipo Físico**
* Cliente: **Leo**
* Pagamento inicial: **50% já realizado**
* Primeira reunião de briefing feita: **28/05**
* Reunião de kickoff agendada: **17/06**
* Modalidade: **Web App responsivo**

**🎯 O que foi definido no briefing (resumo prático)**

**🟪 Calculadora de Estilo**

* Formato tipo quiz;
* Preferência por **escolha entre pares de imagens** (modelo similar ao “This or That”);
* Cada escolha **acumula pontos invisivelmente**, que ao final indicam um **estilo predominante** (clássico, moderno, etc.);
* A **lógica de pontuação ainda será refinada** com a equipe do cliente.

**🟩 Calculadora de Tipo Físico**

* Entrada de **medidas manuais** (ombro, cintura, quadril...);
* Será exibida uma imagem **estática de bonecas** com biotipos diferentes;
* A aluna **seleciona a boneca que mais se assemelha a ela** (etapa 2);
* Na etapa 1, há sugestão de uma **animação orientativa** sobre como tirar medidas corretamente (imagem Animacao\_a\_ser\_feita.png);
* A proposta de ter **bonecas "dinâmicas" que mudam conforme medidas foi bem recebida**, mas será considerada como uma **etapa futura (fase 2)**.

**🛠️ O QUE VOCÊ DEVE FAZER AGORA (1º passo técnico)**

**🎯 Etapa 1: Estruturação funcional dos fluxos (em wireframe ou protótipo funcional estático)**

**Entregável: Um protótipo navegável ou estrutura HTML/JS básica mostrando o fluxo de:**

**🟪 1. Calculadora de Estilo:**

* Tela inicial com instruções
* Página de perguntas com pares de imagens (mesmo que simuladas agora)
* Lógica básica de escolha → pontuação oculta
* Tela final com estilo gerado (placeholder)
* Botão de exportar dados (em CSV ou JSON)

**🟩 2. Calculadora de Tipo Físico:**

* Etapa 1: entrada de medidas com imagem animada ou estática (ex: explicando como medir ombro)
* Etapa 2: tela com **bonecas disponíveis** para seleção
* Tela final com tipo físico selecionado
* Botão de exportar dados

✅ **Ferramentas recomendadas agora:**

* HTML + JS (usando Bootstrap/Tailwind, sem backend por enquanto)
* Figma (caso queira mostrar o fluxo antes de codificar)
* JS puro com manipulação de DOM ou React (opcional se quiser já estruturar bem)

**📂 PASTAS E ORGANIZAÇÃO INICIAL**

Crie uma estrutura assim:

calculadora-estilo/

├── index.html

├── css/

├── js/

├── imagens/

└── dados-exemplo.json

calculadora-tipo-fisico/

├── index.html

├── css/

├── js/

├── imagens/

└── medidas-exemplo.json

**🎁 BÔNUS: Dados exportáveis (CSV/JSON)**

Comece com JSON simulado, por exemplo:

{

"usuario\_id": "123",

"estilo\_predominante": "Clássico",

"respostas": ["A", "B", "A", "C", ...]

}

**📅 Recomendações para a reunião de 17/06**

1. Mostre uma **demonstração navegável simples** dos dois fluxos;
2. Traga um **modelo de pontuação oculta por imagem** para a calculadora de estilo;
3. Pergunte se os estilos já foram definidos (nome, cores, símbolos);
4. Pergunte se deseja que a animação de medição seja em vídeo, SVG animado ou apenas sequência de imagens;
5. Alinhe com o Leo se haverá **login/registro futuro**, ou se o MVP será “uso rápido + exportação”.

**📌 Conclusão**

Você está pronto para iniciar a parte técnica com segurança.  
O **primeiro passo** agora é criar os **fluxos visuais básicos (com lógica simples embutida)** para serem apresentados na reunião.

Se quiser, posso gerar um esqueleto inicial com as telas em HTML/JS + placeholders ou até montar esse fluxo no Figma com você. Deseja isso?

*DIFERENÇA ENTRE AS GERAÇÕES DE ARQUIVOS ‘JSON’ X ‘XLS’*

Atualmente ela salva as informações em um arquivo no formato JSON. Eu gostaria de explicar por que ele está assim e qual a diferença se mudarmos para um formato de planilha, como Excel, para que vocês possam me dizer o que faz mais sentido para o uso de vocês.

Vamos pensar nos resultados da calculadora – seu estilo e tipo físico, as medidas, as respostas do quiz.

Opção 1: Salvar como JSON (Como está agora)

Imaginem que o JSON é como um caderno de anotações organizado para computadores.

Como é o arquivo: Quando vocês abrem, pode parecer um pouco técnico, com chaves {} e colchetes []. É porque ele é feito para ser lido e entendido facilmente por outros programas de computador.

Qual a vantagem dele (pensando no futuro): Esse formato é excelente se, mais para frente, vocês quiserem, por exemplo:

- Criar um histórico dos seus resultados na plataforma (se tivéssemos login).

- Usar esses dados para receber recomendações automáticas de moda personalizadas.

- Compartilhar esses resultados de forma programática com outros sistemas ou aplicativos.

É um formato que guarda todos os detalhes da análise de forma fiel e estruturada para ser usado por outras funcionalidades que possamos desenvolver.

Qual a desvantagem (para vocês, no dia a dia): Ele não é prático para abrir e simplesmente ler ou usar diretamente em programas como Excel. Não é feito para ser uma tabela visual fácil.

Opção 2: Salvar como Planilha (Como Excel)

Agora, imaginem que a planilha é como uma tabela organizada para pessoas.

Como seria o arquivo: Seria um arquivo .csv ou .xlsx (de Excel), com tudo organizado em linhas e colunas. Muito familiar!

Qual a vantagem dele (para vocês, no dia a dia):

- É super fácil de abrir, ler e entender.

- Vocês podem usar o Excel para visualizar, imprimir, fazer anotações, ou até fazer alguma ordenação simples dos dados.

- É simples de compartilhar com alguém que precise apenas ver os resultados de forma tabular.

Qual a desvantagem (pensando no desenvolvimento e no futuro):

- Gerar esse tipo de arquivo diretamente da página web é um pouco mais complexo tecnicamente do que gerar o JSON.

- Planilhas são ótimas para dados simples e retos, mas nem sempre conseguem representar perfeitamente informações mais detalhadas ou aninhadas, como todas as respostas individuais do quiz junto com os totais. Teríamos que decidir quais informações colocar na planilha para que ela ficasse simples.

- Não é o formato ideal se a gente quiser que outro sistema de computador leia esses resultados automaticamente no futuro para oferecer funcionalidades avançadas.

Em Resumo:

**JSON:** Ideal para 'conversar' entre programas de computador, guarda a estrutura completa, ótimo para base de futuras funcionalidades avançadas. Menos prático para leitura e uso manual imediato por vocês.

**Planilha:** Ideal para leitura e uso manual por vocês no dia a dia, fácil de visualizar e usar no Excel. Menos ideal para ser usado por outros sistemas de computador de forma automatizada e pode não guardar todos os detalhes da estrutura original.

*Então, a pergunta para vocês é: Como vocês imaginam usar esses resultados depois de exportar? Vocês pretendem apenas guardá-los para consultar, imprimir ou talvez usar em uma análise simples no Excel? Ou o plano é que esses dados sirvam de base para a própria plataforma oferecer algo a mais para vocês no futuro?*

Dependendo do uso que vocês planejam, podemos manter como JSON (se o foco for mais técnico/futuro da plataforma) ou trabalhar para oferecer a opção de planilha (se o foco for a praticidade para vocês acessarem os dados manualmente). Podríamos até, no futuro, oferecer ambas as opções!"

==================== X ==================

***Dicas Adicionais para a Reunião:***

Use a analogia do caderno vs. tabela: Reforce-a durante a explicação.

Foque no 'para que serve PARA VOCÊS': Conecte as vantagens de cada formato com o uso que o cliente faria dos dados.

Seja Confante: Mostre que você entende ambos os formatos e a complexidade envolvida, mas que a escolha final depende da necessidade deles.

Mostre o JSON (opcional): Se for apropriado, abra rapidamente o arquivo .json que é gerado para eles verem como ele realmente se parece – isso pode ajudar a entender a parte da 'leitura por computador' e a necessidade de um formato mais amigável.

Finalize com a Pergunta: Termine perguntando como eles imaginam usar os dados. Isso coloca a decisão nas mãos deles e mostra que você está focado em atender às necessidades reais do negócio.

Com essa explicação, você desarma a complexidade técnica do JSON e foca nos benefícios e usos práticos de cada formato sob a ótica do cliente.

***PENSANDO NO FUTURO – MODULARIZANDO IMAGENS E PERGUINTAS***

Abordagem 1: Melhorar a Documentação e Estrutura Atual (Evolução do JSON)

Esta é a opção que exige menos trabalho seu agora, mas mais 'trabalho' (ou risco) por parte do cliente no futuro.

Como funciona: O cliente continua editando o arquivo config-imagens.json e gerencia os arquivos de imagem manualmente.

O que você faria:

Criar um documento extremamente claro e didático (um pequeno manual, talvez um PDF ou um vídeo curto) explicando passo a passo como editar o arquivo config-imagens.json.

Use exemplos visuais.

Explique o que cada parte significa ("texto", "opcoes", "imagem", "descricao").

Deixe bem claro a importância das vírgulas, chaves, colchetes e aspas.

Sugira usar um editor de texto que valide JSON (como VS Code com a extensão JSON Tools, ou editores online gratuitos de validação JSON) para minimizar erros.

Estruturar a pasta de imagens de forma organizada (ex: imagens/quiz/, imagens/tipos-fisicos/).

Criar um processo simples para eles substituírem ou adicionarem imagens na pasta imagens/ e atualizarem o caminho no JSON. Se as imagens estiverem online (como as de Pexels), explicar como obter o link direto da imagem.

Vantagens (principalmente para você):

É a implementação mais rápida no seu lado. Você já tem a base.

Nenhum desenvolvimento de back-end é necessário.

Desvantagens (principalmente para o cliente):

Requer que o cliente edite um arquivo técnico, o que é propenso a erros e pode ser frustrante.

Gerenciamento de imagens (upload/substituição) é manual e pode ser confuso.

Adicionar novos tipos de perguntas (ex: com múltipla escolha em vez de apenas duas opções) pode exigir edições mais complexas no JSON e talvez ajustes no JavaScript se a estrutura do JSON mudar muito.

Abordagem 2: Utilizar um Formato de Arquivo Mais Simples para Edição (CSV)

Relaciona-se com a discussão anterior sobre planilhas. Pode ser uma alternativa ao JSON para a edição manual pelo cliente.

Como funciona: Você criaria um script (em JS, talvez rodando em Node.js para facilitar) que lê um arquivo mais simples (como .csv, que pode ser editado no Excel) e gera o config-imagens.json que a aplicação usa. O cliente editaria a planilha e rodaria seu script gerador.

O que você faria:

Definir um formato de planilha (.csv) claro para as perguntas e opções. Isso seria o desafio: como representar perguntas com múltiplas opções e imagens associadas em linhas e colunas? Talvez uma linha por opção, referenciando a pergunta à qual pertence? Isso pode ficar complexo rapidamente.

Desenvolver um script (ex: generate\_config.js) que leia o CSV, valide os dados e crie o config-imagens.json no formato correto.

Criar um manual para o cliente explicando como editar a planilha e como rodar o script gerador (ex: abrir o terminal, digitar node generate\_config.js).

O gerenciamento de imagens ainda seria manual, como na Abordagem 1.

Vantagens (equilíbrio):

Editar uma planilha é muito mais familiar para a maioria dos clientes do que editar JSON. Menor chance de erros de sintaxe no arquivo fonte.

Separa a forma de edição (planilha) da forma de uso pela aplicação (JSON).

Desvantagens (para você e cliente):

Você precisa desenvolver o script parser/gerador.

O cliente ainda precisa realizar uma etapa técnica extra (rodar o script no terminal) e gerenciar imagens manualmente.

Representar a estrutura do quiz em CSV pode ser complicado.

Abordagem 3: Criar uma Interface de Administração Simples (Admin Panel)

Esta abordagem melhora muito a experiência do cliente, mas aumenta significativamente a complexidade para você, pois exige um back-end.

Como funciona: Você desenvolveria uma pequena aplicação web separada (ou integrada) com uma interface visual onde o cliente poderia adicionar, editar ou excluir perguntas e opções através de formulários, e fazer upload das imagens diretamente pela interface.

O que você faria:

Desenvolver um back-end (usando Java/Spring Boot, Node.js/Express/NestJS, Python/Django/Flask, etc. – suas áreas de expertise!) para:

Armazenar as perguntas, opções e dados em um banco de dados (relacional ou NoSQL, o que for mais adequado para a estrutura dos dados do quiz).

Implementar funcionalidades de upload de arquivos para as imagens, armazenando-as no servidor ou em um serviço de storage (como S3 da AWS).

Criar APIs para o front-end do admin panel buscar/salvar/atualizar os dados.

Desenvolver o front-end da interface de administração (HTML, CSS, JavaScript, talvez um framework JS) com formulários para edição de texto e botões de upload de arquivo.

Adaptar o front-end principal da calculadora para buscar os dados das perguntas e imagens da API do seu back-end em vez de ler o config-imagens.json localmente.

Vantagens (principalmente para o cliente):

Muito fácil de usar: Interface visual, formulários, upload de arrastar e soltar. Nenhuma necessidade de editar arquivos técnicos ou usar terminal.

Menor chance de erro: A validação pode ser feita no formulário e no back-end.

Gerenciamento de imagens integrado e simples.

Desvantagens (principalmente para você):

Maior complexidade: Exige desenvolvimento full-stack (back-end, banco de dados, APIs) além do front-end.

Requer hospedagem para o back-end e banco de dados.

Tempo de desenvolvimento e custo (de hospedagem) significativamente maiores.

Abordagem 4: Integrar com um CMS "Headless" (Sistema de Gerenciamento de Conteúdo sem Interface Visual)

Similar à Abordagem 3 em termos de experiência do cliente, mas você terceiriza a complexidade do back-end para um serviço especializado.

Como funciona: Você usaria um serviço de CMS headless (como Strapi, Contentful, DatoCMS, etc.). Esses serviços oferecem uma interface web amigável para o cliente gerenciar conteúdo e mídia, e expõem esse conteúdo via APIs (REST ou GraphQL).

O que você faria:

Escolher e configurar um serviço de CMS headless.

Definir os "Tipos de Conteúdo" no CMS para representar a estrutura das suas perguntas, opções e imagens.

Criar as perguntas e fazer o upload das imagens usando a interface do CMS.

Adaptar o front-end da calculadora para buscar os dados diretamente da API fornecida pelo CMS (em vez de ler o JSON local).

Vantagens (equilíbrio entre facilidade e controle):

Muito amigável para o cliente: Eles usam uma plataforma dedicada para gerenciar conteúdo e mídia.

Você não precisa desenvolver o back-end do zero (banco de dados, APIs, upload de arquivos).

Muitos CMS headless têm planos gratuitos para começar.

Desvantagens (novas dependências):

Você introduz uma dependência externa (o serviço de CMS).

Pode haver custos envolvidos dependendo do plano e volume de conteúdo/tráfego.

Requer aprender a usar a API do CMS escolhido.

Conclusão e Recomendação para Discussão com o Cliente

Todas essas abordagens tornam o sistema mais modular no sentido de separar o conteúdo, mas elas variam enormemente em quão fácil a edição desse conteúdo se torna para o cliente "leigo" e em quão complexa a sua implementação será.

Para a sua reunião hoje:

*Explique essas opções para o cliente focando na facilidade que ele terá para alterar o conteúdo no futuro versus o esforço (e, portanto, custo/tempo) que você terá para implementar a solução.*

**Opção A (JSON Atual com Manual):** Mais barata e rápida para você implementar, mas mais difícil e arriscada para eles usarem. Boa se as alterações forem muito raras e eles tiverem alguém minimamente técnico na equipe deles que possa seguir um manual.

**Opção B (Planilha CSV com Script):** Um meio-termo. Edição mais familiar para eles, mas ainda exige rodar um script e gerenciar imagens. Exige um pouco mais de desenvolvimento seu.

**Opção C (Admin Panel Próprio) ou D (CMS Headless):** Exigem o maior investimento inicial de desenvolvimento (seu tempo/custo) ou um custo contínuo (CMS pago), mas oferecem a maior facilidade e segurança para o cliente fazer as alterações no futuro por conta própria. Ideal se a frequência de alteração de conteúdo for alta e eles quiserem total independência.

***Pergunte: "Com que frequência vocês esperam mudar essas perguntas e imagens no futuro? Quem seria a pessoa responsável por fazer essas mudanças? Qual nível de facilidade de uso vocês desejam para essa tarefa?"***

As respostas deles guiarão qual abordagem faz mais sentido para o projeto e o orçamento. Mantenha o foco em otimizar a entrega de valor para eles, alinhando a solução técnica com a necessidade de uso do conteúdo!

Motive-se com as possibilidades! Como full-stack, você tem as ferramentas para implementar qualquer uma dessas opções, desde a mais simples até a mais completa. Resolva com eficiência escolhendo a abordagem que melhor se alinha com os requisitos do cliente após a conversa!

REUNIÃO 17/06 (15h – 15:20H)

- Aline Franco

- Marina Naves

Marina informou que as perguntas (e respostas) para a parte do cálculo do estilo serão definidos até 30/06. A mecânica consistirá na cliente ver uma imagem com a descrição do estilo daquela imagem e selecionar ou não como sendo um estilo que ela goste. Isto deverá ocorrer em uma rodada de 7 perguntas inicialmente, depois em uma 2ª rodada, outra imagem aparecerá desta vez com 5 perguntas, excluindo a que ela escolheu como sendo o estilo que ela ‘acha’ que gostou, ou que pelo menos se identificou, e finalmente numa 3ª e última rodada outras perguntas serão feitas e aí sim, será determinado qual o estilo da pessoa.

Para a parte do cálculo do tipo físico, as medidas são exatamente essas mesmas! Só irão avaliar se vão ou não incluir peso e altura, e irão me passar exatamente como é feito o cálculo para se determinar se um tipo físico é triangular, retangular e por aí vai.

Aventaram a possibilidade de, futuramente, a análise do tipo físico ser feito por IA, quando a pessoa iria tirar uma foto e através da análise (pela IA) da foto o tipo físico seria determinado. Ponderou que a foto pode ser tirada errada e que é mais fácil inserir as medidas. Informei tb que não estava no escopo proposto inicialmente com Leo e que pode ser implementado, mas não por esse preço!

01/07 – Marina entrou em contato por ZAP e (em áudio) perguntou e obs o seguinte:

Já estão todas perguntas em um doc.

Muitas em imagens (para que a cliente responda qual prefere das 7 calças ou look).

As imagens estão em um arquivo quadrado (tipo instagram)

Tudo deve aparecer mesmo na tela, todas as peças (calça, blusa, etc) que formarão ‘o look’ (partes um do lado do outro).

Cada pergunta deve mostrar as 7 imagens/ looks que ela vai passando pela a, b, etc (podendo voltar) e depois escolhe uma.

1. Qual o tamanho da imagem que deve ser enviada?
2. Qual o nível de personalização deste teste? No sentido da identidade visual deles. Tem a logo e a cartela de cores (preto, branco e bege)

GPT 🡺

Ótimo, Alexandre! Vamos por partes para garantir que o quiz atenda perfeitamente às expectativas do cliente:

**✅ 1) Formato e Navegação do Quiz com Imagens**

* Cada pergunta deve exibir **7 imagens quadradas**, lado a lado ou em linhas organizadas (por exemplo: 4 na primeira linha + 3 na segunda, adaptável para responsividade).
* O usuário **navegará por perguntas** (e não por alternativas). Então, a navegação será entre **perguntas diferentes**, e cada uma já exibirá as 7 opções visuais (a, b, c...).
* É essencial ter **botões de navegação** do tipo “Próxima Pergunta” e “Voltar” visíveis, além de destaque visual para a opção selecionada.
* Ao final, será exibido o **resultado baseado nas escolhas feitas**.

**📐 2) Tamanho ideal das imagens**

* Recomendo que as imagens tenham formato **quadrado (1:1)**, com:
  + **Tamanho ideal:** 600x600px (boa definição e leveza)
  + **Formato:** .jpg ou .webp (compactos); .png somente se houver transparência
  + **Nome dos arquivos:** use nomes padronizados como p1-a.jpg, p1-b.jpg, ..., p2-a.jpg, etc., para facilitar carregamento dinâmico via JSON.

**🎨 3) Nível de Personalização (Identidade Visual)**

Você pode aplicar facilmente:

* **Logo no topo** (arquivo .png ou .svg)
* **Cartela de cores** nas seguintes áreas:
  + Botões (ex: bege com texto preto)
  + Fundo da página (branco ou bege claro)
  + Destaques de seleção (borda dourada, por exemplo)
* **Fontes personalizadas** via Google Fonts, caso desejem algo elegante (ex: “Playfair Display” ou “Cormorant Garamond”)

*Você pode montar o estilo base usando CSS com variáveis personalizáveis (--primary-color, --bg-color, etc.) para facilitar futuras trocas.*

**📁 4) Solução recomendada (estrutura modular e escalável)**

* /index.html
* /style.css (usando variáveis de cor e logo via config.json)
* /script.js (lógica de quiz e navegação entre perguntas)
* /config-imagens.json (estrutura que define perguntas, opções e caminhos das imagens)
* /imagens/ (subpasta com todas as imagens: p1-a.jpg, p2-d.jpg...)
* /assets/logo.svg ou /logo.png

**🔧 Exemplo de entrada no config-imagens.json:**

json

CopiarEditar

{

"logo": "assets/logo.svg",

"cores": {

"primaria": "#000000",

"secundaria": "#ffffff",

"realce": "#d6bfa3"

},

"perguntas": [

{

"texto": "Qual desses looks você usaria para um jantar especial?",

"opcoes": [

{ "id": "a", "imagem": "imagens/p1-a.jpg" },

{ "id": "b", "imagem": "imagens/p1-b.jpg" },

...

]

},

...

]

}

**📌 Ações sugeridas:**

1. Solicite ao cliente que envie as imagens em **formato quadrado (preferencialmente 600x600px)**.
2. Peça o **logo vetorizado ou em alta definição**.
3. Solicite a **cartela de cores com os códigos hexadecimais (#000, #fff, etc.)**.
4. Assim que receber o .doc, posso ajudar a **converter para JSON automaticamente**.
5. Com isso, você terá um **quiz 100% modular, adaptável a qualquer cliente**, bastando trocar o config-imagens.json e a pasta imagens/.

Se quiser, posso já montar uma versão básica do config-imagens.json, index.html, script.js e style.css adaptados para esse novo formato, prontos para popular depois que as imagens chegarem. Deseja isso?

Perfeito, Alexandre! Com essa atualização, o comportamento desejado está claro:

* Cada pergunta mostra **apenas uma imagem por vez** (de um total de 7);
* O usuário pode **navegar entre as 7 opções** (com botões de avançar e voltar);
* Depois de ver todas, escolhe a que preferir;
* Ao final de todas as perguntas, é exibido o resultado com base nas escolhas.

Essa lógica será implementada no JavaScript com controle de índice atual por pergunta e renderização dinâmica da imagem. A estrutura config-imagens.json pode conter algo assim para cada pergunta:

json

CopiarEditar

{

"perguntas": [

{

"titulo": "Qual look você mais gosta para o dia a dia?",

"opcoes": [

{ "imagem": "perguntas/pergunta1/a.jpg", "legenda": "Look A" },

{ "imagem": "perguntas/pergunta1/b.jpg", "legenda": "Look B" },

{ "imagem": "perguntas/pergunta1/c.jpg", "legenda": "Look C" },

...

]

},

...

]

}

Se quiser, posso gerar o novo script.js e HTML modular com essa lógica atualizada e pronta para o cliente. Deseja isso agora?

**Você disse:**

Teve mais uma dúvida deles: "sobre a parte de design do teste… o fundo é manipulável? poderia pedir pra uma designer criar e te enviar?"

**O ChatGPT disse:**

Sim, pode informar ao cliente que **o fundo é completamente manipulável** e que **a designer pode criar e enviar um arquivo personalizado**, que será incorporado facilmente ao projeto.

**💡 Formas de personalizar o fundo:**

**✅ 1. Imagem de fundo personalizada (recomendado se querem algo visualmente marcante)**

A designer pode enviar:

* Um **arquivo JPG ou PNG leve**, com no máximo **2000px de largura** (para evitar lentidão em mobile);
* Preferencialmente em proporção horizontal (ex: 1920x1080);
* Sem elementos muito pequenos, para não se perderem em telas menores.

Exemplo de uso no CSS:

css

CopiarEditar

body {

background-image: url('imagens/fundo-personalizado.jpg');

background-size: cover;

background-position: center;

background-attachment: fixed;

}

**✅ 2. Cor sólida ou gradiente personalizado**

Se quiserem usar a **cartela de cores** (preto, branco e bege), isso é ainda mais leve e elegante.

Exemplo:

css

CopiarEditar

body {

background: linear-gradient(135deg, #fefefe, #f5f0e6);

color: #000; /\* cor do texto \*/

}

**🔁 Trocar o fundo depois é fácil?**

Sim. O design é **modular**. Basta substituir:

* O arquivo fundo-personalizado.jpg por outro com o mesmo nome, **ou**
* Ajustar o CSS em style.css, na parte body { ... }.

**✔️ Resumo para enviar ao cliente:**

Sim! O fundo do teste é totalmente personalizável. A designer pode nos enviar uma imagem de fundo (em JPG ou PNG, de preferência até 1920px de largura), ou podemos usar uma cor sólida ou gradiente com base na cartela de cores da marca. Basta nos enviar a arte ou as cores que cuidamos do resto!

Se quiser, posso já criar um modelo de style.css com fundo personalizado e paleta bege/branco/preto para teste. Deseja isso?

// DÚVIDA SOBRE PONTUAÇÃO



Ótima observação, Alexandre! E você está certo em questionar: se o **estilo "CRIATIVA" teve 22 seleções** e o **"SEXY" teve 18**, **o secundário deveria ter sido "CRIATIVA"**, e o terciário "SEXY".

**✅ Causa do problema**

A seleção dos estilos **não leva em conta a segunda e terceira maior pontuação global**, mas sim faz um **novo cálculo a cada fase**, *recomeçando a pontuação do zero*, e apenas *excluindo os estilos já escolhidos*.

Ou seja:

* A cada fase, o script **zera** a contagem com resetPontuacao().
* Os estilos são pontuados **de acordo com as respostas dadas na fase atual**, não com o acumulado anterior.

**✅ Isso é um erro?**

**Não é exatamente um erro**, mas **é uma limitação da lógica atual**, porque:

* O estilo **primário é baseado na pontuação total da Fase 1**.
* O estilo **secundário é recalculado do zero na Fase 2**, com novas respostas, **e pode ter outra distribuição completamente diferente**.
* O **terciário também** é recalculado com as opções restantes, sem considerar os totais das fases anteriores.

✅ **Isso é proposital no seu quiz**, porque o objetivo é:  
➡️ "Se o estilo primário não estivesse disponível, qual seria sua segunda escolha?"  
➡️ "Se os dois principais estilos estivessem ausentes, o que você escolheria?"  
➡️ Portanto, **o quiz é estruturado por fases intencionalmente independentes.**

**✅ Agora, se você quiser MUDAR isso…**

Se você quiser que:

* O **primário seja o estilo mais pontuado**
* O **secundário seja o 2º mais pontuado, independente da fase**
* O **terciário seja o 3º mais pontuado no total**

Então será necessário **alterar completamente a lógica do quiz**, abandonando o sistema de 3 fases independentes com resets.

**✅ Alternativas para você decidir:**

**✅ MANTER como está (3 fases independentes)**

✅ A lógica atual do quiz faz sentido pedagógico e estilístico: **o estilo secundário não é o 2º mais votado, mas a 2ª escolha real da usuária em nova rodada**, sem o primário.

**🔁 ALTERAR para cálculo TOTAL baseado na Fase 1**

Se você decidir mudar, **basta não fazer Fase 2 e 3**, e no final da Fase 1 você faria:

js

CopiarEditar

const pontuacoes = faseCounts[1]; // pontuação da fase 1

const ordenado = Object.entries(pontuacoes)

.sort((a, b) => b[1] - a[1])

.map(([estilo]) => estilo);

estilosPrimarioSecundario.primary = ordenado[0];

estilosPrimarioSecundario.secondary = ordenado[1];

estilosPrimarioSecundario.tertiary = ordenado[2];

Ou seja, os 3 maiores pontuadores de uma única rodada definiriam o perfil final.

**✅ Minha recomendação**

**Se o quiz foi concebido com 3 fases deliberadas (excluir o escolhido anterior e repetir as perguntas), está funcionando exatamente como deveria.**  
Seu resultado está certo.

Mas se quiser mudar para análise “estatística” do total de respostas **sem recomeçar**, posso adaptar todo o quiz para funcionar assim — é só você confirmar.

Quer manter como está (três fases de escolha sequencial) ou deseja migrar para o sistema de ranking direto por pontuação total?

Perguntar ao ChatGPT

Parte superior do formulário

Ferramentas

Parte inferior do formulário

O ChatGPT pode cometer erros. Considere verificar info

// LÓGICA DE DESEMPATE APLICADA

Alexandre, **sim** — a lógica de desempate **já está parcialmente implementada** no seu código atual, com base na função:

js

CopiarEditar

function aplicarDesempate(estilosEmpatados, respostas, estilosExcluidos)

E a ordem está definida corretamente como:

js

CopiarEditar

const ordemDesempate = [23, 29, 37, 41, 42, 43, 40, 39, 36, 33, 28, 27, 26, 25];

✅ Isso já atende **exatamente às instruções que você forneceu**:

● Se empatar, será válido a resposta na pergunta 23 [...] depois 29, 37, 41, 42, 43, 40, 39, 36, 33, 28, 27, 26, 25

**✅ O que já está funcionando corretamente**

A função getEstiloVencedor() no seu código usa esse critério:

js

CopiarEditar

if (vencedoresPotenciais.length > 1) {

return aplicarDesempate(vencedoresPotenciais, respostasPorPergunta, estilosExcluidos);

}

E a função aplicarDesempate() percorre a lista ordemDesempate, verificando se a resposta dada em alguma dessas perguntas coincide com **um dos estilos empatados**.

Se encontrar, esse estilo é imediatamente retornado como vencedor.

**✅ O que você não precisa mudar**

Você **não precisa alterar nenhum trecho de código atual** — ele **já está 100% compatível** com a lógica de desempate fornecida.

Inclusive, já está previsto o caso em que **nenhuma dessas perguntas resolve o empate**, aplicando como fallback o estilo **em ordem alfabética**:

js

CopiarEditar

const sortedEmpatados = estilosEmpatados.sort();

return sortedEmpatados.find(s => !estilosExcluidos.includes(s)) || sortedEmpatados[0];

**✅ Onde está tudo isso no seu código**

* Função principal: getEstiloVencedor(...)
* Desempate: aplicarDesempate(...)
* Lista de prioridade: const ordemDesempate = [...] — já com a sequência correta
* Fonte dos dados: respostasPorPergunta[qNum]

Se você quiser **mostrar no console** *quando* e *por qual pergunta* o desempate foi resolvido, você pode adicionar **este log útil** dentro da função aplicarDesempate, assim:

js

CopiarEditar

function aplicarDesempate(estilosEmpatados, respostas, estilosExcluidos) {

for (const qNum of ordemDesempate) {

const respostaDaPergunta = respostas[qNum];

if (respostaDaPergunta && estilosEmpatados.includes(respostaDaPergunta.estilo) && !estilosExcluidos.includes(respostaDaPergunta.estilo)) {

console.log(`🔁 Desempate resolvido pela pergunta ${qNum}: estilo escolhido = ${respostaDaPergunta.estilo}`);

return respostaDaPergunta.estilo;

}

}

const sortedEmpatados = estilosEmpatados.sort();

console.log(`⚠️ Desempate não resolvido pelas perguntas específicas. Fallback = ${sortedEmpatados[0]}`);

return sortedEmpatados.find(s => !estilosExcluidos.includes(s)) || sortedEmpatados[0];

}

Se quiser, posso te enviar a versão .zip com essa melhoria aplicada. Deseja?

Perguntar ao ChatGPT

Parte superior do formulário

Ferramentas

Parte inferior do formulário

O ChatGPT pode cometer erros. Co