

oi

Oi! 😊

Como posso te ajudar agora — é algo de campanha/creativos, automação, copy, ou uma coisa mais rápida?



vou mandar dois materiais completos abaixo e ele é oq vamos usar pra uma dinamica.

Doc 1:

MODELO CONSOLIDADO

Motor Modular de Estados Cognitivos e Emocionais

Tese central (fechada):

Você não está organizando conteúdo.

Você está orquestrando estados psicológicos, usando um sistema modular.

Tudo abaixo serve a essa tese.

A separação correta — agora definitiva — é:
Tema → o que é verdadeiro

Estação → por que o tema é usado (intenção psicológica)

Dinâmica → como o cérebro é迫使ado a agir

Tarefa → execução concreta que gera estado emocional

- ✖ A variação real mora na tarefa.
- ✖ A estabilidade mora nas três camadas acima.

2 ESTAÇÕES (INTENÇÃO PSICOLÓGICA)

As estações não são pedagógicas.

Elas existem porque o cérebro humano entra em estados diferentes conforme a necessidade emocional.

- ◆ Estação de Prática
Psicologia dominante
redução de incerteza

sensação de controle

previsibilidade

Motivação real de entrada

"Eu ainda não confio em mim."

✗ Não é sobre aprender.

✗ É sobre reduzir ansiedade.

◆ Estação de História

Psicologia dominante

identificação

espelhamento

significado

Motivação real de entrada

"Quero entender por que isso importa."

✗ História não ensina.

✗ História ancora valor emocional.

◆ Estação de Desafio

Psicologia dominante

status

autoimagem

prova interna

Motivação real de entrada

"Quero provar que sei."

- 📌 Desafio não é aprendizado.
- 📌 É validação do ego.

◆ Estação de Revisão

Psicologia dominante
medo de esquecer

aversão à perda

segurança futura

Motivação real de entrada

"Não posso errar isso depois."

◆ Estação de Avaliação

Psicologia dominante
fechamento

clareza

conclusão

Motivação real de entrada

"Onde eu estou agora?"

📌 Nota estrutural importante

Essas 5 estações não são obrigatórias.

Um produto saudável opera com 3 a 5, raramente mais.

3 DINÂMICAS (GATILHOS COGNITIVOS)

Dinâmicas não são didáticas.

Elas são forçadores mentais inevitáveis.

Dinâmica existe para forçar um comportamento mental específico.

Exemplos (psicológicos, não técnicos)

Quiz → força decisão

Escolha → força comparação

Recall → força memória ativa

Simulação → força antecipação

Ranking → força identidade social

Tempo → força foco

- ✗ Dinâmica não ensina.
- ✗ Dinâmica não explica.
- ✗ Dinâmica pressiona o cérebro a agir.

4 TAREFA (ONDE O PODER REAL ESTÁ)

Aqui está o núcleo do sistema.

A tarefa é onde a mesma dinâmica gera estados emocionais totalmente diferentes.

Exemplo consolidado — Dinâmica: Quiz

- ◆ Tarefa A — Neutra
texto

sem tempo

feedback simples

🧠 Estado mental: aprendizado seguro

- ◆ Tarefa B — Tensão

cenário real

tempo limitado

consequência explícita

🧠 Estado mental: risco / foco

◆ Tarefa C — Status
ranking

comparação com outros

pontuação pública

🧠 Estado mental: validação social

- 📌 Mesma dinâmica
- 📌 Mesmo tema
- 📌 Experiência psicológica completamente diferente
- 👉 Isso é o motor do engajamento.

5 ATRAÇÃO NÃO É PREÇO

(ajuste conceitual importante)
Você acertou em cheio aqui.
Preço é um atalho fraco de valor.
O que realmente gera atração é:
◆ Tensão psicológica bem calibrada
As pessoas se envolvem quando:
algo está em jogo

existe uma lacuna

existe risco simbólico

existe curiosidade ativa

- 📌 Um sistema barato + alta tensão
- 📌 supera
- 📌 um sistema caro + baixa tensão

6 COMO GERAR ATRAÇÃO SEM MEXER NO PREÇO

O sistema faz isso em 3 níveis claros:

- 1 Entrada livre
usuário escolhe a estação

sensação de autonomia

2 Variação de tarefa
nunca sabe exatamente como será

curiosidade contínua

3 Feedback significativo
não só "acertou/errou"

mas "o que isso revela sobre você"

📌 Isso gera engajamento psicológico, não dependência financeira.

7 REGRA DE OURO (fechada)
Guarda isso como princípio do sistema:
Preço escala receita.
Psicologia escala engajamento.
Seu sistema não é educacional.
Ele é psicologia aplicada via estrutura modular.

8 O QUE VOCÊ REALMENTE CONSTRUIU (síntese final)
Você não construiu:
um sistema educacional

uma plataforma de cursos

Você construiu:

Um motor modular de estados cognitivos e emocionais,
onde a variação psicológica acontece na tarefa,
e a estabilidade estrutural garante escala.

Isso é:

raro

sólido

difícil de copiar

e agora... bem desenhado

9 O QUE FALTA (lacunas reais, não teóricas)

Agora, honestamente, só faltam 3 coisas para isso virar produto-operável:

Catálogo mínimo de dinâmicas (8–12 no máximo)

Regras de permissão

quais dinâmicas entram em quais estações

Template oficial de tarefa

o "molde" que toda tarefa deve respeitar

Nada além disso.

Próximo passo (sugestão objetiva)

Se eu fosse você, seguiria nesta ordem:

- 1 Definir catálogo de dinâmicas
- 2 Criar matriz Estação x Dinâmica
- 3 Criar template de tarefa psicológica

Se quiser, faço qualquer um dos três na próxima resposta.

DOCUMENTO MESTRE: O MOTOR MODULAR DE ESTADOS COGNITIVOS

Tese Central (O DNA do Sistema)

Você não está construindo uma plataforma de cursos ou gestão de conteúdo. Você está construindo um Motor Modular de Estados Cognitivos e Emocionais.

O objetivo do sistema não é apenas "ensinar", mas gerenciar a tensão psicológica e o fluxo de aprendizado através de uma arquitetura componetizada.

1 A ARQUITETURA (A Hierarquia de Estabilidade)

A ordem correta — definitiva e imutável — é:

TEMA (A Verdade) → O conhecimento puro e neutro.

ESTAÇÃO (A Intenção) → O estado psicológico que queremos ativar.

DINÂMICA (O Gatilho) → O mecanismo que força o cérebro a agir.

TAREFA (A Experiência) → A execução concreta que gera a emoção.

📌 Regra de Ouro: A estabilidade mora nas camadas de cima. A variação infinita mora na Tarefa.

2 AS ESTAÇÕES (O Mapa Psicológico)

As estações não são divididas por "tipo de mídia", mas pela necessidade emocional do usuário.

Estação de Prática

Psicologia: Redução de incerteza, sensação de controle, previsibilidade.

Gatilho de Entrada: "Eu ainda não confio em mim."

Função: Não é sobre aprender, é sobre diminuir a ansiedade.

Estação de História

Psicologia: Identificação, espelhamento, busca de significado.

Gatilho de Entrada: "Quero entender por que isso importa."

Função: História não ensina técnica. História ancora valor emocional.

Estação de Desafio

Psicologia: Status, autoimagem, prova interna.

Gatilho de Entrada: "Quero provar que sei/sou capaz."

Função: Desafio não é aprendizado. É validação do ego.

Estação de Revisão

Psicologia: Aversão à perda, medo de esquecer, segurança futura.

Gatilho de Entrada: "Não posso perder o que conquistei."

Função: Manutenção e segurança.

Estação de Avaliação

Psicologia: Fechamento, clareza, localização espacial.

Gatilho de Entrada: "Onde eu estou agora?"

Função: Conclusão de ciclo.

3 DINÂMICAS (Os Gatilhos Cognitivos)

Dinâmicas são forçadores de comportamento. Elas são agnósticas de conteúdo e emoção.

O que elas fazem: Pressionam o cérebro a sair da passividade.

Exemplos de Mecanismos:

Quiz → Força Decisão

Escolha → Força Comparação

Recall → Força Memória Ativa

Simulação → Força Antecipação

Ranking → Força Identidade Social

Tempo → Força Foco

4 TAREFA (O Coração do Engajamento)

Aqui reside o diferencial do seu sistema. É na Tarefa que a mesma dinâmica e o mesmo tema geram experiências emocionais opostas.

Exemplo Prático Consolidado (Dinâmica: Quiz)

Tarefa A (Neutra / Segurança)

Config: Texto simples, sem tempo, feedback explicativo.

🧠 Estado Mental: Aprendizado seguro. (Ideal para Estação de Prática).

Tarefa B (Tensão / Risco)

Config: Cenário realista, cronômetro regressivo, consequência de erro explícita.

🧠 Estado Mental: Foco absoluto e adrenalina. (Ideal para Estação de

Desafio).

Tarefa C (Status / Social)

Config: Comparação com média, ranking, pontuação pública.

🧠 Estado Mental: Validação social e competitividade. (Ideal para Estação de Desafio/Avaliação).

📌 Conclusão: Você escala o produto criando Tarefas, não criando novas Estações ou Dinâmicas.

5 A LÓGICA DE ATRAÇÃO (Psicologia > Preço)

O valor percebido do produto não vem do preço, vem da Tensão Psicológica Bem Calibrada.

A Fórmula: Um sistema barato com alta tensão psicológica vence um sistema caro com baixa tensão (tédio).

Os 3 Pilares da Atração:

Entrada Livre (Autonomia): O usuário escolhe a estação. Isso gera compromisso.

Variedade de Tarefa (Curiosidade): O usuário nunca sabe exatamente qual variação virá (recompensa variável).

Feedback Significativo (Identidade): A resposta não é "Errou", é "O que isso diz sobre sua habilidade".

6 REGRAS DE GOVERNANÇA (A Constituição)

Para garantir que o sistema não quebre ao escalar:

Regra da Escala: Preço escala receita; Psicologia escala engajamento.

Regra da Modularidade: Nunca crie uma nova Estação para resolver um problema que pode ser resolvido com uma nova Tarefa.

Regra da Imutabilidade: O Tema nunca muda por causa do formato.

Regra da Orquestração: Quem define o tom do produto é a combinação de Estações, não o conteúdo.

O QUE FALTA PARA ISSO RODAR (Próximos Passos Reais)

O conceito está fechado, sólido e pronto. As lacunas para transformar isso em "chão de fábrica" (dev/produto) são apenas três:

O Catálogo Mínimo de Dinâmicas: Definir as 8 a 12 dinâmicas mestras que cobrem 90% das interações humanas.

A Matriz de Permissão: Definir quais Dinâmicas são permitidas em quais Estações (ex: Ranking não entra em Prática).

O Template de Tarefa: O modelo de dados (JSON/Estrutura) que define uma tarefa para o banco de dados.

Qual desses 3 você quer fechar agora para oficializar o sistema?

Tese Central (O DNA do Sistema)

Você não está construindo uma plataforma de cursos ou gestão de conteúdo. Você está construindo um Motor Modular de Estados Cognitivos e Emocionais.

O objetivo do sistema não é apenas "ensinar", mas gerenciar a tensão psicológica e o fluxo de aprendizado através de uma arquitetura componetizada.

1 A ARQUITETURA (A Hierarquia de Estabilidade)

A ordem correta — definitiva e imutável — é:

TEMA (A Verdade) → O conhecimento puro e neutro.

ESTAÇÃO (A Intenção) → O estado psicológico que queremos ativar.

DINÂMICA (O Gatilho) → O mecanismo que força o cérebro a agir.

TAREFA (A Experiência) → A execução concreta que gera a emoção.

📌 Regra de Ouro: A estabilidade mora nas camadas de cima. A variação infinita mora na Tarefa.

2 AS ESTAÇÕES (O Mapa Psicológico)

As estações não são divididas por "tipo de mídia", mas pela necessidade emocional do usuário.

Estação de Prática

Psicologia: Redução de incerteza, sensação de controle, previsibilidade.

Gatilho de Entrada: "Eu ainda não confio em mim."

Função: Não é sobre aprender, é sobre diminuir a ansiedade.

Estação de História

Psicologia: Identificação, espelhamento, busca de significado.

Gatilho de Entrada: "Quero entender por que isso importa."

Função: História não ensina técnica. História ancora valor emocional.

Estação de Desafio

Psicologia: Status, autoimagem, prova interna.

Gatilho de Entrada: "Quero provar que sei/sou capaz."

Função: Desafio não é aprendizado. É validação do ego.

Estação de Revisão

Psicologia: Aversão à perda, medo de esquecer, segurança futura.

Gatilho de Entrada: "Não posso perder o que conquistei."

Função: Manutenção e segurança.

Estação de Avaliação

Psicologia: Fechamento, clareza, localização espacial.

Gatilho de Entrada: "Onde eu estou agora?"

Função: Conclusão de ciclo.

3 DINÂMICAS (Os Gatilhos Cognitivos)

Dinâmicas são forçadores de comportamento. Elas são agnósticas de conteúdo e emoção.

O que elas fazem: Pressionam o cérebro a sair da passividade.

Exemplos de Mecanismos:

Quiz → Força Decisão

Escolha → Força Comparação

Recall → Força Memória Ativa

Simulação → Força Antecipação

Ranking → Força Identidade Social

Tempo → Força Foco

4 TAREFA (O Coração do Engajamento)

Aqui reside o diferencial do seu sistema. É na Tarefa que a mesma dinâmica e o mesmo tema geram experiências emocionais opostas.

Exemplo Prático Consolidadado (Dinâmica: Quiz)

Tarefa A (Neutra / Segurança)

Config: Texto simples, sem tempo, feedback explicativo.

🧠 Estado Mental: Aprendizado seguro. (Ideal para Estação de Prática).

Tarefa B (Tensão / Risco)

Config: Cenário realista, cronômetro regressivo, consequência de erro explícita.

🧠 Estado Mental: Foco absoluto e adrenalina. (Ideal para Estação de Desafio).

Tarefa C (Status / Social)

Config: Comparação com média, ranking, pontuação pública.

🧠 Estado Mental: Validação social e competitividade. (Ideal para Estação de Desafio/Avaliação).

📌 Conclusão: Você escala o produto criando Tarefas, não criando novas Estações ou Dinâmicas.

5 A LÓGICA DE ATRAÇÃO (Psicologia > Preço)

O valor percebido do produto não vem do preço, vem da Tensão Psicológica Bem Calibrada.

A Fórmula: Um sistema barato com alta tensão psicológica vence um sistema caro com baixa tensão (tédio).

Os 3 Pilares da Atração:

Entrada Livre (Autonomia): O usuário escolhe a estação. Isso gera compromisso.

Variedade de Tarefa (Curiosidade): O usuário nunca sabe exatamente qual variação virá (recompensa variável).

Feedback Significativo (Identidade): A resposta não é "Errou", é "O que isso diz sobre sua habilidade".

6 REGRAS DE GOVERNANÇA (A Constituição)

Para garantir que o sistema não quebre ao escalar:

Regra da Escala: Preço escala receita; Psicologia escala engajamento.

Regra da Modularidade: Nunca crie uma nova Estação para resolver um problema que pode ser resolvido com uma nova Tarefa.

Regra da Imutabilidade: O Tema nunca muda por causa do formato.

Regra da Orquestração: Quem define o tom do produto é a combinação de Estações, não o conteúdo.

O QUE FALTA PARA ISSO RODAR (Próximos Passos Reais)

O conceito está fechado, sólido e pronto. As lacunas para transformar isso em "chão de fábrica" (dev/produto) são apenas três:

O Catálogo Mínimo de Dinâmicas: Definir as 8 a 12 dinâmicas mestras que cobrem 90% das interações humanas.

A Matriz de Permissão: Definir quais Dinâmicas são permitidas em quais Estações (ex: Ranking não entra em Prática).

O Template de Tarefa: O modelo de dados (JSON/Estrutura) que define uma tarefa para o banco de dados.

Qual desses 3 você quer fechar agora para oficializar o sistema?

Perfeito. Essa imagem traz a estrutura tática do Duolingo (a interface do usuário e as mecânicas exatas). Vamos pegar esses 5 pilares visuais e traduzi-los diretamente para o seu sistema de "Reembolso Ágil".

Aqui está o mapeamento de onde cada elemento se encaixa na nossa conversa anterior e qual nome daremos a ele no seu sistema corporativo:

1. Streaks (Ofensiva)

O que é na imagem: Rastreia dias consecutivos de atividade do usuário (o fogo 🔥).

Onde se encaixa no nosso projeto: Entra na "Aversão à Perda". É o indicador visual de que o funcionário é consistente.

Como funciona no Reembolso: Não contaremos "dias", mas sim Meses Consecutivos Sem Erros ou Meses Entregues no Prazo.

Nome no Sistema: "Chama da Eficiência" ou "Sequência de Conformidade".

Regra: Se o funcionário entregar o relatório atrasado ou com erro grave em um mês, o fogo apaga e ele volta ao nível "Novato" (perdendo o reembolso rápido).

2. Experience Points / XP (Pontos de Experiência)

O que é na imagem: Pontos ganhos ao completar uma lição.

Onde se encaixa no nosso projeto: Entra nas "Recompensas Variáveis". É a moeda de troca imediata.

Como funciona no Reembolso: O usuário ganha XP a cada Nota Fiscal

válida inserida.

+10 XP: Nota enviada.

+20 XP bônus: Nota enviada no mesmo dia da compra (estimula o tempo real).

+50 XP: Fechamento do mês sem nenhuma pendência.

Nome no Sistema: "Score de Gestão" ou "Pontos de Auditoria".

3. Leaderboards / League (Ranking e Ligas)

O que é na imagem: Compara o progresso do usuário com outros.

Onde se encaixa no nosso projeto: Entra na "Comparação Social".

Como funciona no Reembolso: Para não expor financeiramente ninguém, faremos Ligas entre Departamentos.

Exemplo: A "Liga Diamante" tem os times com menos erros de notas (ex: TI e RH). A "Liga Bronze" tem os times que mais atrasam (ex: Comercial e Marketing). Isso cria uma competição saudável entre gestores.

Nome no Sistema: "Ranking de Compliance" ou "Liga da Agilidade".

4. Badges (Medalhas/Conquistas)

O que é na imagem: Ganhas ao atingir marcos específicos (milestones).

Onde se encaixa no nosso projeto: É um reforço da "Progressão e Níveis". Diferente do XP (que é contínuo), a Badge é uma honraria permanente.

Como funciona no Reembolso: São marcos de carreira dentro da ferramenta.

Badge "The Flash": Enviou 10 notas em menos de 1 hora após a compra.

Badge "Auditor Zero": Completou 1 ano sem nenhum apontamento de erro.

Badge "Liderança": O gestor que aprova os reembolsos do time em

menos de 24h.

Nome no Sistema: "Selos de Excelência".

5. Hearts (Vidas/Corações)

O que é na imagem: Número de tentativas/erros permitidos antes de "perder".

Onde se encaixa no nosso projeto: Entra no "Feedback Imediato" e tolerância a falhas.

Como funciona no Reembolso: Essa é a parte mais genial para o corporativo. O funcionário tem "Créditos de Confiança".

Todo mês o funcionário começa com 5 Corações.

Cada vez que o financeiro precisa devolver uma nota (recusar) por erro besta (foto ruim, valor errado), ele perde um coração.

Se perder todos os corações no mês: O reembolso dele vai para o fim da fila (auditoria manual pente-fino) e demora 30 dias.

Se terminar com corações cheios: Reembolso expresso.

Nome no Sistema: "Créditos de Confiança" ou "Saúde do Relatório".

Resumo da Tela do Usuário (Dashboard)

Com base nessa imagem e no que definimos, o painel do seu sistema ficaria assim:

Topo Esquerdo: 🔥 12 Meses (Sequência de Conformidade - Streaks)

Topo Direito: ❤️ 5/5 (Créditos de Confiança para este mês - Hearts)

Centro: Barra de Progresso do Nível Atual (Baseado no XP)

Abaixo: "Sua equipe está em 2º lugar na Liga Diamante"
(Leaderboard)

Rodapé: Suas Conquistas Recentes: "Selo The Flash desbloqueado"
(Badges)

Doc 2:

IDEA BUILD INFO APP

1) A ideia em 1 frase

Um gerador de “infoapps”: qualquer conteúdo (vídeo, ebook, aula, método) vira um app de aprendizado gamificado, com trilhas, desafios diários, professor/mentor na tela, pontos e recompensas — e isso pode escalar pra dezenas (80+) de apps.

2) O que isso resolve (a dor real)

Hoje, conteúdo é:

longo, solto, sem rotina

difícil de “virar prática”

a pessoa começa e larga

Você quer transformar conteúdo em:
experiência diária

progresso visível

sensação de jogo + evolução

aprendizado em micro-missões

3) Promessa simples (o resultado que a pessoa sente)

"Em vez de consumir conteúdo e esquecer, eu entro todo dia, faço um desafio rápido, ganho pontos, vejo meu progresso e sinto que tô avançando de verdade."

4) O mecanismo (a sacada que faz funcionar)

A sacada é: conteúdo vira jogo
trilha (sequência guiada)

desafios diários (microvitórias)

quiz rápido pra fixar

pontos / níveis / streak / badges

professor/mentor (avatar) conduzindo as lições

Isso transforma "aprender" em hábito + recompensa.

5) Como o usuário vive isso (fluxo em linguagem humana)

A pessoa abre o app → vê "sua trilha" e "missão de hoje"

Faz uma interação rápida (tipo duolingo: escolher, completar, responder)

Ganha pontos + sobe progresso

Desbloqueia coisas (aulas/etapas/níveis/colecionáveis)

Volta amanhã porque:

não quer perder sequência

quer acumular pontos

quer destravar níveis/recompensas

6) O "corpo" do infoapp (o que existe dentro)

Essenciais

Trilhas (caminhos prontos: do básico ao avançado)

Aulas em pedaços curtos

Desafios diários

Quizzes rápidos

Progresso visível (dias, sequência, evolução)

Vibe Duolingo (gamificação)
XP, streak, badges, ranking (opcional)

"galeria" de conquistas e de recompensas

Extra que dá retenção forte
Feed estilo TikTok/Instagram dentro do app

vídeos curtinhos (30s–1min) com dicas pontuais

a pessoa "explora" e aprende sem esforço

7) Recompensas e economia (o loop de pontos)

A pessoa ganha pontos por:

completar desafios

quizzes

consistência

Ela troca pontos por:

recompensas simples (cupons, itens, benefícios)

acesso a coisas dentro do ecossistema

O ponto-chave da tua ideia: isso vira uma “economia” onde aprender dá sensação de ganho real.

8) A tese de escala (por que isso pode virar 80 apps)

Você não está criando “um curso”.

Você está criando uma fábrica de produtos em formato app.
achou um conteúdo que tá viral?

tem um método que funciona?

tem um nicho com demanda?

→ vira um infoapp com a mesma estrutura (trilha + desafios + gamificação + feed)

9) O truque de eficiência (sem falar técnico, só a lógica)
Você comentou uma lógica muito forte:
o conteúdo/roteiro do "professor" e os desafios são gerados uma vez

depois o app só reaproveita o que já existe

e você pode lançar sem ter tudo pronto:

conforme as pessoas avançam, o conteúdo vai "sendo
criado/organizado" naturalmente

Tradução prática: o produto cresce com o uso, sem você precisar

"construir tudo antes".

10) MVP (a versão mínima que prova tudo)

Sem inventar muita coisa, o MVP pode ser:

1 infoapp (ex: "10K com MicroSaaS em 7 dias" ou o de inglês que você já tem)

1 trilha curta (7 a 14 dias)

1 desafio por dia + quiz final

progresso simples (dias + sequência + pontos)

uma "loja" simples de recompensas (mesmo que no começo seja simbólico)

Só isso já prova:

retenção diária

conclusão de trilha

desejo por recompensas

demandas por "mais apps"

11) Como você descreve a ideia pra alguém (pitch rápido)

"É um app que transforma qualquer conteúdo em uma jornada diária gamificada — tipo Duolingo, só que pra qualquer tema. Você aprende por desafios curtos, ganha pontos, vê progresso e pode trocar por recompensas. E do nosso lado, isso vira uma máquina de criar vários apps a partir de conteúdos que já existem."

Se você quiser, eu fecho isso ainda mais em 1 página de visão do produto com:
nome provisório (3 opções)

público-alvo principal

3 diferenciais que viram "tese"

e o core loop desenhado (em bullets bem simples).

BASE MÉTODO

Thought for 11s >

Perfeito — **Doc 1** é a “constituição” (motor psicológico) e **Doc 2** é o “produto” (InfoApp Factory) que vai rodar em cima desse motor.

Abaixo está o que dá pra usar imediatamente como dinâmica operacional (chão de fábrica): **catálogo de dinâmicas (10)** + matriz **Estação × Dinâmica (permissão)** + **template oficial de tarefa (JSON)** + **exemplos prontos** mostrando como a mesma dinâmica muda totalmente o estado via *tarefa*.

1) Tradução do Doc 2 para o Motor do Doc 1 (mapeamento rápido)

- **Tema (verdade neutra)**: o conteúdo em si (método/aula/ebook/vídeo).
- **Estação (intenção psicológica)**: por que o usuário entrou hoje (ansiedade, sentido, ego, segurança, fechamento).
- **Dinâmica (gatilho cognitivo)**: qual forçador mental vai obrigar ação.
- **Tarefa (experiência)**: micro-missão concreta (onde você “calibra tensão”, cria hábito e retenção).

O InfoApp é literalmente: **uma fábrica de tarefas**, com **estações e dinâmicas estáveis**.

2) Catálogo mínimo de Dinâmicas (10) — “as mestras”

1. **Quiz (Decisão)** → escolher alternativa / julgar certo-errado
2. **Escolha Forçada (Comparação)** → “qual é melhor/pior/mais provável?”
3. **Recall (Memória ativa)** → lembrar sem ver; preencher lacunas
4. **Simulação (Antecipação)** → “o que você faria agora?”

5. Ordenação (Prioridade) → ordenar passos / hierarquizar riscos
6. Matching (Associação) → conectar conceito ↔ exemplo ↔ consequência
7. Checklist/Missão (Execução) → completar micro-ação real no mundo
8. Diagnóstico (Autoimagem) → “qual nível você está?” / perfil rápido
9. Tempo (Foco/Risco) → cronômetro e pressão leve/média/alta
10. Ranking (Identidade social — opcional) → comparação controlada

Observação: Ranking é potente, mas é “químico forte”. Use com matriz e regras.

3) Matriz Estação x Dinâmica (permissão prática)

Legenda: recomendado | usar com cuidado | evitar

Estação de Prática (reduzir ansiedade / controle)

- Quiz
- Escolha
- Recall
- Simulação (sem “consequência”, mais guiada)
- Ordenação
- Matching
- Checklist/Missão (micro e previsível)
- Diagnóstico (sem julgamento, só “mapa”)
- Tempo (tempo folgado)
- Ranking

Estação de História (significado / identificação)

- Quiz  (quiz de interpretação/visão)
- Escolha  (dilemas, "qual eu faria?")
- Recall  (quebra clima)
- Simulação  (narrativa interativa)
- Ordenação 
- Matching  (conectar "causa→efeito" do personagem)
- Checklist/Missão  (ação "1% hoje")
- Diagnóstico  (espelhamento, "quem eu sou nessa história")
- Tempo 
- Ranking 

Estação de Desafio (status / prova interna)

- Quiz 
- Escolha 
- Recall 
- Simulação 
- Ordenação 
- Matching 
- Checklist/Missão  (missão "de performance")
- Diagnóstico  (nível/elo)
- Tempo 
- Ranking  (de preferência: por liga/segmento, não exposição geral)

Estação de Revisão (medo de esquecer / segurança futura)

- Quiz 
- Escolha 

- Recall
- Simulação (só para consolidar)
- Ordenação
- Matching
- Checklist/Missão (rotina de manutenção)
- Diagnóstico (apenas acompanhamento)
- Tempo
- Ranking

Estação de Avaliação (fechamento / "onde estou?")

- Quiz
- Escolha
- Recall
- Simulação (caso final)
- Ordenação
- Matching
- Checklist/Missão (se for "prova de campo")
- Diagnóstico (mapa de nível/posição)
- Tempo (moderado)
- Ranking (somente se não gerar vergonha; ranking privado ou por equipe)

4) Template Oficial de Tarefa (JSON) — “molde imutável”

json

Copy code

```
{  
    "task_id": "string",  
    "theme_id": "string",  
    "station": "PRATICA | HISTORIA | DESAFIO | REVISAO | AVALIACAO",  
    "dynamic": "QUIZ | ESCOLHA | RECALL | SIMULACAO | ORDENACAO | MATCHING | MISSAO | DIAGNOSTICO",  
    "title": "string",  
    "prompt": "string",  
    "difficulty": 1,  
    "tension_profile": "SAFE | TENSION | STATUS",  
    "time_limit_sec": 0,  
    "attempts": 1,  
    "scoring": {  
        "xp_base": 10,  
        "xp_bonus_rules": ["string"],  
        "fail_penalty": 0  
    },  
    "consequence": {  
        "type": "NONE | SOFT | HARD",  
        "description": "string"  
    },  
    "feedback": {  
        "type": "EXPLAIN | REVEAL_ABOUT_YOU | COACHING",  
        "message_correct": "string",  
        "message_wrong": "string"  
    },  
    "mentor_script": {  
        "avatar_id": "string",  
        "before": "string",  
        "after": "string"  
    }  
}
```

```
},
"content_blocks": [
  { "type": "TEXT | IMAGE | VIDEO | AUDIO", "ref": "string", "caption": "string" }
],
"hints": [
  { "trigger": "AFTER_WRONG | TIME_50 | OPTIONAL", "text": "string" }
],
"rewards_unlocks": {
  "streak_affects": true,
  "badge_ids": ["string"],
  "unlock_content_ids": ["string"],
  "shop_tokens": 0
},
"social_visibility": {
  "mode": "PRIVATE | TEAM | PUBLIC",
  "ranking_group": "string"
},
"analytics": {
  "events": ["task_started", "task_completed", "task_failed", "hint_used"]
}
}
```

Regra do Doc 1 aplicada:

- ➡ Estação + Dinâmica ficam estáveis, você escala criando novas Tarefas (mudando prompt, tempo, consequência, feedback, visibilidade, recompensa).
-

5) Exemplos prontos: mesma Dinâmica (QUIZ) → 3 tarefas (SAFE / TENSION / STATUS)

(A) SAFE — Estação Prática

- tension_profile: SAFE
- time_limit: 0
- feedback: EXPLAIN

Prompt: "Qual alternativa descreve melhor 'Big Idea' em VSL?"

Feedback: explica o porquê e dá exemplo simples.

(B) TENSION — Estação Desafio

- tension_profile: TENSION
- time_limit: 20s
- consequence: SOFT (perde bônus de XP / streak shield)

Prompt: "Seu anúncio travou. Em 20s, qual Big Idea é mais 'crível' pro público frio?"

Feedback: "isso revela seu viés: você escolhe 'desejo' sem 'mecanismo'!"

(C) STATUS — Estação Avaliação

- tension_profile: STATUS
- ranking: TEAM (liga por turma)
- feedback: REVEAL_ABOUT_YOU

Prompt: "Você escolheu o ângulo X. Você está no 'Nível 3: Promessa sem Prova'. Próximo passo: ..."

Efeito: progresso visível + comparação controlada.

6) Como transformar isso na dinâmica que vocês vão usar (passo-a-passo curto)

- 1. Escolha 1 Tema** (ex: "Big Idea", "YouTube sem aparecer", "Reembolso Ágil", etc.)
 - 2. Selecione 3 Estações** (ex: Prática → Desafio → Revisão)
 - 3. Para cada estação, selecione 2 Dinâmicas permitidas** pela matriz
 - 4. Para cada dinâmica, crie 3 tarefas** (SAFE / TENSION / STATUS) usando o JSON
 - 5. Defina o loop do InfoApp:**
 - **Missão do dia** (1 tarefa)
 - **Progresso** (XP + streak)
 - **Fechamento** (feedback "o que isso revela sobre você")
-

Se você quiser, eu já faço a versão "oficial" pronta pra você colar no seu sistema:

- **(1)** uma **Matriz completa** em formato de tabela (Estação × Dinâmica com ✓⚠️✖️)
- **(2)** um **catálogo de 30 tarefas base** (3 por dinâmica) com variações SAFE/TENSION/STATUS
- **(3)** um **Core Loop final** (bullets) alinhado com Duolingo + economia de recompensas

Vou direto no que mais destrava o produto: **(2) catálogo de 30 tarefas base.**



A gente precisa fazer essas ideias funconarem e esse sistema realmente existir completo da melhor forma e linguagem. sempre simule as 5 pessoas mais adequadas e ideias pra ajudar esse desenvolvimento completo e a cada rodada de mensagens do usuário um deles pode chamar uma outra pessoa que ele julga ser mais adequada para entrar na conversa, e contribuir, esse sistema de simulação terá uma pessoa anotando, e tudo que o usuário sinalizar

ser importante, e isso será salvo no placeholder [resumo] que no final pode ser resgatado.

vamos começar.

usuario: oi pessoal, estamos aqui nesse conselho reunido para realmente fazer algo unico e historico, é a democratização do mundo de negocios de ensino, onde qualquer influencer ou pessoa pode transformar uma ideaia em um negocio de edtech completo. na mesa de voces tem dois documentos sobre a ideia e um pouco da filosofia e estrutura, eae, como vaoms começar?

Thought for 8s >

Perfeito. Vamos iniciar como um **Conselho** com 5 especialistas + 1 escriba (anotador). A cada rodada, um de nós pode puxar mais um especialista se fizer sentido.

Conselho (rodada 1)

1. Arq. de Produto (**EdTech/Plataforma**) – transforma filosofia em sistema operável
 2. Cientista Comportamental – garante que “estados” sejam reais e repetíveis
 3. Designer de Jogo/Economia – loop, recompensas, streak, progressão, retenção
 4. Arquiteto de Dados/Engenharia – entidades, JSON, permissões, escalabilidade (80+ apps)
 5. Estrategista de Go-To-Market (**creator economy**) – oferta para criador, linguagem, embalagem
-  **Escriba** – registra tudo importante no placeholder **[resumo]**

Resposta ao usuário (como começamos)

Arq. de Produto (abre a reunião):

"Se a tese é 'não é conteúdo, é orquestração de estados', então o primeiro passo não é código nem design. É fechar a linguagem oficial + o sistema mínimo operável. Sem isso, a fábrica vira caos."

Ordem de partida (curta e certeira)

1) Congelar o vocabulário oficial (1 página)

- Tema / Estação / Dinâmica / Tarefa
- O que é e o que *não* é cada coisa
- Regra de ouro: *variação mora na tarefa; estabilidade mora acima*

2) Definir o “Core Loop” imutável do usuário (1 tela mental)

- Abre → vê missão do dia → faz 1 interação → recebe feedback significativo → ganha progresso → volta amanhã
- Esse loop precisa funcionar em **qualquer tema** e em qualquer infoapp.

3) Definir o “Creator Loop” (a fábrica)

- Criador sobe conteúdo bruto → sistema gera/organiza tarefas → cria trilha + missão diária → publica infoapp
- Aqui mora o “80 apps”: **pipeline repetível**.

4) Fechar a governança (matriz de permissão + limites)

- O que pode em cada estação (Ranking fora de Prática etc.)
- Quanta tensão por dia (pra não virar cassino/ansiedade)

5) Definir o MVP que prova o motor

- 1 infoapp
- 1 trilha de 7–14 dias

- 1 missão diária
 - XP + streak + progresso visível
 - Loja simples (mesmo simbólica)
-

⌚ Contribuições rápidas (cada especialista dá o “primeiro tijolo”)

Cientista Comportamental:

“Pra isso existir de verdade, precisamos de uma coisa objetiva:

➡️ um ‘Tension Dial’ (SAFE / TENSION / STATUS) obrigatório em toda tarefa.

Sem isso, vocês não controlam estado psicológico; só fazem atividades.”

Designer de Jogo/Economia:

“Antes de loja e recompensas, fechem:

- **Streak** (aversão à perda)
- **XP** (progresso)
- **Badges** (identidade)

Ranking é opcional e perigoso: entra só quando vocês tiverem ‘ligas’ e ‘visibilidade controlada’”

Arq. de Dados/Engenharia:

“Para escalar 80 apps, o sistema precisa ser **conteúdo-agnóstico**.

Eu começaria definindo 6 entidades:

- InfoApp
- Tema
- Estação
- Dinâmica

- Tarefa
- Recompensa

E um JSON oficial de tarefa (vocês já estão perto)."

Go-To-Market (Creator Economy):

"O produto não pode ser vendido como 'app builder'.

Tem que ser vendido como:

➡ 'Transforma ideia em produto diário que retém e vira hábito'.

Primeira oferta: criador sai com um app 'Duolingo do meu tema' em X dias."

🔥 Proposta de plano de trabalho (a partir de agora)

Na próxima rodada, pra avançar com velocidade, eu sugiro que a gente feche **1 decisão por vez**:

Rodada 2 (agora): "1-pager oficial"

- Nome interno do sistema (provisório)
- Core Loop do usuário (em bullets)
- Creator Loop (em bullets)
- 10 dinâmicas mestras (fixo)
- Tension Dial (obrigatório)

📌 E aí a gente parte pra **Matriz Estação x Dinâmica final + templates**.

👤 Um especialista extra entrando (chamado pelo Arq. de Produto)

"Vou puxar mais um perfil essencial agora:

UX de Habit Products (apps de hábito tipo Duolingo/Strava)

porque precisamos garantir que a experiência diária seja *irresistível sem ser tóxica.*"

Pergunta objetiva pra você (pra guiar a próxima rodada)

Quando você diz "democratização do mundo de negócios do ensino", o foco principal do produto é:

1. **Criador** (fábrica de produtos/edtech para influenciadores)

ou

2. **Aluno** (experiência diária e retenção como prioridade máxima)

Eu consigo tocar os dois, mas **a ordem de decisões muda** dependendo disso.

[resumo]

- Tese central mantida: "não é conteúdo, é orquestração de estados psicológicos via arquitetura modular."
- Ordem estrutural fixada: Tema → Estação → Dinâmica → Tarefa (variação real na Tarefa).
- Proposta de início: congelar vocabulário oficial + core loop do usuário + creator loop (fábrica) + governança (permissões/limites) + MVP 7–14 dias.
- Introduzido "Tension Dial" obrigatório por tarefa: SAFE / TENSION / STATUS.
- Componentes de retenção priorizados: streak, XP, badges; ranking opcional e controlado.
- Escala (80 apps) exige modelo conteúdo-agnóstico + entidades base (InfoApp, Tema, Estação, Dinâmica, Tarefa, Recompensa).
- Posicionamento sugerido: "transforma ideia em produto diário que vira hábito", não "app builder".

- Especialista extra chamado: UX de Habit Products (Duolingo/Strava-like) para garantir loop diário saudável.



Creator Loop é uma ideia interessante, não vamos nos apegar aos detalhes e sim a necessidade de desenvolvermos uma lógica dessa pra produção, salve do placeholder essa ideia

Fechado. **Creator Loop** está oficialmente salvo no [\[resumo\]](#) como “lógica de produção” (pipeline repetível), sem apego a detalhes.

[resumo]

- **Creator Loop (Lógica de Produção)**: pipeline repetível para transformar conteúdo bruto em um InfoApp operável (trilha + missão diária + tarefas + feedback + progresso), permitindo escalar para dezenas de apps sem “reconstruir do zero”.



Salve: Escala (80 apps) exige modelo conteúdo-agnóstico + entidades base (InfoApp, Tema, Estação, Dinâmica, Tarefa, Recompensa).

Salvo no **[resumo]** exatamente como você pediu.

[resumo]

- **Escala (80 apps) exige modelo conteúdo-agnóstico + entidades base:** InfoApp, Tema, Estação, Dinâmica, Tarefa, Recompensa.



Pessoal, gostei, mas acho importante questionarmos nossa ideia pq só assim vamos avaliar.

O foco é o Criador (fábrica de produtos/edtech para influenciadores).

Acho que tudo isso vale a pena mas não quero que seja só um app de coisinhas rápidas. Acho que podem ter aulas tmb, mas numa dinâmica diferente, são aulas interativas, o usuário vai ver um player, mas pra gente serão telas já prontas com áudio narrado do elevenlabs e imagens bonitas svg animados, mas essa é uma ideia na quero limitar, esse negócio de educação precisa ser funcional para o negócio realmente gerar resultado e crescer

Pensou por 12s >

Perfeito — e é exatamente esse tipo de “stress test” que separa um brinquedo de um sistema histórico.

E fechado: o foco é o Criador (fábrica).

Conselho (rodada atual) — e uma pessoa extra entra

Hoje eu puxo mais uma cadeira: **Designer Instrucional** (learning design de produtos que geram resultado real), porque você acabou de colocar “profundidade + funcionalidade” como requisito.

1) A pergunta certa: “aula” é o quê dentro do nosso motor?

Arq. de Produto:

Aula não pode virar uma nova Estação (porque estação é intenção psicológica, não mídia).

Aula também não precisa virar uma nova Dinâmica.

➡ Aula é um tipo de Tarefa com um “container” diferente: Player Interativo.

Tradução simples do sistema:

- Estação = por que a pessoa entrou hoje (ansiedade, significado, status, etc.)
- Dinâmica = o forçador mental (decidir, lembrar, simular...)
- Tarefa = experiência concreta
- Aula Interativa = *uma tarefa composta* (com áudio+visuais+micro-interações)

Isso mantém a tese intacta: variação mora na tarefa.

2) Como não virar “app de coisinhas rápidas”

Designer Instrucional:

Se você quer educação funcional (resultado), você precisa de **3 tipos de experiência**, não só "missões":

1. Micro-missão (1–3 min)

- hábito + consistência + progressinho diário

2. Aula Interativa (6–12 min)

- entendimento + construção de modelo mental
- com *checkpoints* obrigatórios (senão vira consumo passivo)

3. Projeto/Aplicação (15–45 min, 1–2x/semana)

- transferência pro mundo real
- prova de resultado (e isso vira "case" e retenção)

Duolingo é forte em 1).

Um EdTech que muda vida precisa do combo **1 + 2 + 3**.

3) A "Aula Interativa" como tarefa composta (modelo mental)

Designer de Jogo/Economia:

"Aula interativa" funciona quando você trata como *episódio*, não como vídeo:

- Dividir em **beats** (ex: 8 beats de 45–90s)
- A cada beat, um **checkpoint** (escolha/quiz/recall/simulação)
- Recompensa: XP + desbloqueio + "sensação de avanço real"
- Sem checkpoint = vira Netflix (e aí morre o motor)

Cientista Comportamental:

O checkpoint não é "pra testar", é pra **forçar processamento ativo**.

Sem isso, não existe estado cognitivo — só entretenimento.

4) Isso encaixa no Creator Loop (produção) sem explodir complexidade?

Arq. de Dados/Engenharia:

Sim, desde que a gente defina um **Blueprint de Aula** (um formato de produção repetível) que gere telas automaticamente.

Blueprint (lógica de produção, não detalhe técnico):

- Entrada do criador:
 - roteiro (texto) ou vídeo/transcrição
 - assets (imagens/ícones) opcional
 - “objetivo da aula” + nível
- Saída da fábrica:
 - sequência de **telas** (cards) com: narração (ElevenLabs), imagem/svg/anim, e checkpoint
 - no final: “resumo + ação aplicada”

Isso vira “linha de montagem”.

5) Onde a gente deve duvidar (críticas honestas)

Go-To-Market:

Riscos reais pra não virar um “builder genérico”:

- **Complexidade pro criador** (se pedir demais, ninguém publica)
- **Qualidade inconsistente** (80 apps ruins = morte)
- **Confusão de posicionamento** (“curso? app? gamificação?”)

Arq. de Produto (antídotos):

- Criador não cria “o app”, cria **um conteúdo** e escolhe **um formato**
 - O sistema oferece **poucos formatos oficiais** (arquétipos), e pronto
-

6) Decisão prática pra hoje (pra avançar sem travar)

A gente precisa oficializar 3 “Arquétipos de Tarefa” (isso resolve seu medo do “rápido demais”):

1. **Missão** (rápida)
2. **Aula Interativa** (player + checkpoints)
3. **Aplicação** (tarefa no mundo real + prova)

Com isso, dá pra construir qualquer edtech “de resultado”.

✓ O que foi salvo no [resumo] desta rodada

[resumo]

- Foco oficial do produto: **Criador** (fábrica de produtos/edtech para influenciadores).
- “Aula” **não é Estação** (estação é intenção psicológica), e não precisa ser nova Dinâmica; “Aula Interativa” entra como **tipo de Tarefa / tarefa composta**.
- Visão de conteúdo: não limitar a “coisas rápidas”; incluir **aulas interativas** com player, áudio narrado (ex: ElevenLabs), imagens/ SVG animados (sem prender em uma tecnologia).
- Para educação funcional (resultado real), o produto precisa de **3 arquétipos de experiência**:
 1. Micro-missão (hábito), 2) Aula Interativa (modelo mental), 3) Projeto/Aplicação (transferência pro mundo real).
- Aula Interativa deve ter **checkpoints obrigatórios** (micro-interações) para evitar consumo passivo.

- Creator Loop deve incluir um **Blueprint de Aula** como lógica de produção (roteiro/transcrição → telas com narração + visual + checkpoints).
-

Próxima fala (sua): escolha 1 coisa

Pra gente não dispersar, me diga qual você quer fechar agora:

- A) Definir os 3 arquétipos oficialmente (nome + regra + quando usar)
- B) Desenhar o Blueprint da Aula Interativa (em linguagem humana, produção)
- C) Governança de qualidade pra escala (80 apps) (como impedir "apps ruins")



Salve: Dividir em beats (ex: 8 beats de 45–90s)

- A cada beat, um checkpoint (escolha/quiz/recall/simulação)



Salvo no [resumo]:

[resumo]

- **Aula Interativa (formato):** dividir em beats (ex: 8 beats de 45–90s).
- **Regra de engajamento:** a cada beat, inserir um checkpoint obrigatório (ex: escolha / quiz / recall / simulação).



Salve: O checkpoint não é “pra testar”, é pra **forçar processamento ativo**.

Sem isso, não existe estado cognitivo — só entretenimento.

Salvo no **[resumo]** exatamente.

[resumo]

- **Checkpoint (função real)**: não é “pra testar”; é para **forçar processamento ativo**.
- **Regra**: sem checkpoint, **não existe estado cognitivo**, só **entretenimento/consumo passivo**.



Salve: Arq. de Dados/Engenharia:

Sim, desde que a gente defina um Blueprint de Aula (um formato de produção repetível) que gere telas automaticamente.

Blueprint (lógica de produção, não detalhe técnico):

- Entrada do criador:
- roteiro (texto) ou vídeo/transcrição
- assets (imagens/ícones) opcional
- “objetivo da aula” + nível
- Saída da fábrica:

- sequência de telas (cards) com: narração (ElevenLabs), imagem/svg/anim, e checkpoint
 - no final: "resumo + ação aplicada"

Isso vira "linha de montagem".

Salvo no **[resumo]** como especificação oficial do **Blueprint de Aula** (lógica de produção).

[resumo]

- **Blueprint de Aula (produção repetível)**: gera telas automaticamente (linha de montagem).
 - **Entrada do criador:**
 - roteiro (texto) **ou** vídeo/transcrição
 - assets (imagens/ícones) **opcional**
 - "objetivo da aula" + **nível**
 - **Saída da fábrica:**
 - sequência de **telas (cards)** com: **narração (ElevenLabs) + imagem/svg/anim + checkpoint**
 - fechamento com **"resumo + ação aplicada"**



Salve: Arq. de Produto (antídotos):

- Criador não cria "o app", cria um conteúdo e escolhe um formato
- O sistema oferece poucos formatos oficiais (arquetipos), e pronto

Salvo no [resumo].

[resumo]

- **Antídotos (produto/escala):**
 - Criador não cria “o app”; cria **um conteúdo** e escolhe **um formato**.
 - O sistema oferece **poucos formatos oficiais (arquétipos)** — e pronto.



Salve: A gente precisa oficializar 3 “Arquétipos de Tarefa” (isso resolve seu medo do “rápido demais”):

1. Missão (rápida)
2. Aula Interativa (player + checkpoints)
3. Aplicação (tarefa no mundo real + prova)

Salvo no [resumo] como decisão estrutural.

[resumo]

- **3 Arquétipos Oficiais de Tarefa:**
 1. **Missão** (rápida)
 2. **Aula Interativa** (player + checkpoints)
 3. **Aplicação** (tarefa no mundo real + prova)

Salve: Aula" não é Estação (estaçao é intenção psicológica), e não precisa ser nova Dinâmica; "Aula Interativa" entra como tipo de Tarefa / tarefa composta.

- Visão de conteúdo: não limitar a "coisas rápidas"; incluir aulas interativas com player, áudio narrado (ex: ElevenLabs), imagens/ SVG animados (sem prender em uma tecnologia).
- Para educação funcional (resultado real), o produto precisa de 3 arquétipos de experiência:
 1. Micro-missão (hábito), 2) Aula Interativa (modelo mental), 3) Projeto/Aplicação (transferência pro mundo real).
- Aula Interativa deve ter checkpoints obrigatórios (micro-interações) para evitar consumo passivo.
- Creator Loop deve incluir um Blueprint de Aula como lógica de produção (roteiro/transcrição → telas com narração + visual + checkpoints).

Salvo no **[resumo]** como bloco consolidado (fundação do sistema).

[resumo]

- **"Aula" não é Estação** (Estação = intenção psicológica) e não precisa ser nova Dinâmica; **Aula Interativa** entra como **tipo de Tarefa / tarefa composta**.
- **Visão de conteúdo:** não limitar a "coisas rápidas"; incluir **aulas interativas** com player, **áudio narrado** (ex: ElevenLabs), **imagens / SVG animados** (sem prender em tecnologia específica).

- **Educação funcional (resultado real)** exige 3 arquétipos de experiência:
 1. **Micro-missão** (hábito), 2) **Aula Interativa** (modelo mental), 3) **Projeto/Aplicação** (transferência pro mundo real).
- **Aula Interativa** deve ter **checkpoints obrigatórios** (micro-interações) para evitar consumo passivo.
- **Creator Loop** deve incluir um **Blueprint de Aula** como lógica de produção (**roteiro/transcrição** → **telas com narração + visual + checkpoints**).



Ok, antes de avançar. Me envie tudo que temos salvos até agr em
[resumo]

[resumo]

- **Creator Loop (Lógica de Produção)**: pipeline repetível para transformar conteúdo bruto em um InfoApp operável (trilha + missão diária + tarefas + feedback + progresso), permitindo escalar para dezenas de apps sem “reconstruir do zero”.
- **Escala (80 apps)** exige modelo conteúdo-agnóstico + entidades base: **InfoApp, Tema, Estação, Dinâmica, Tarefa, Recompensa**.
- **Aula Interativa (formato)**: dividir em **beats** (ex: 8 beats de 45–90s).
- **Regra de engajamento**: a cada beat, inserir um **checkpoint obrigatório** (ex: **escolha / quiz / recall / simulação**).
- **Checkpoint (função real)**: não é “pra testar”; é para **forçar processamento ativo**.
 - **Regra**: sem checkpoint, **não existe estado cognitivo**, só entretenimento/consumo passivo.
- **Blueprint de Aula (produção repetível)**: gera telas automaticamente (linha de montagem).

- Entrada do criador:
 - roteiro (texto) **ou** vídeo/transcrição
 - assets (imagens/ícones) **opcional**
 - “objetivo da aula” + **nível**
- Saída da fábrica:
 - sequência de **telas (cards)** com: **narração (ElevenLabs)** + **imagem/svg/anim** + **checkpoint**
 - fechamento com “**resumo + ação aplicada**”
- Antídotos (produto/escala):
 - Criador não cria “o app”; cria **um conteúdo** e escolhe **um formato**.
 - O sistema oferece **poucos formatos oficiais (arquétipos)** — e pronto.
- 3 Arquétipos Oficiais de Tarefa:
 1. **Missão** (rápida)
 2. **Aula Interativa** (player + checkpoints)
 3. **Aplicação** (tarefa no mundo real + prova)
- Fundação consolidada:
 - “Aula” **não é Estação** (Estação = intenção psicológica) e não precisa ser nova Dinâmica; **Aula Interativa** entra como **tipo de Tarefa / tarefa composta**.
 - **Visão de conteúdo:** não limitar a “coisas rápidas”; incluir **aulas interativas** com player, áudio narrado (ex: ElevenLabs), **imagens / SVG animados** (sem prender em tecnologia específica).
 - **Educação funcional (resultado real)** exige **3 arquétipos de experiência**:
 1. **Micro-missão** (habito), 2) **Aula Interativa** (modelo mental), 3) **Projeto/Aplicação** (transferência pro mundo real).
 - **Aula Interativa** deve ter **checkpoints obrigatórios** (micro-interações) para evitar consumo passivo.

- Creator Loop deve incluir um Blueprint de Aula como lógica de produção (roteiro/transcrição → telas com narração + visual + checkpoints).



Perfeito vamos seguir a disfunção, abaixo vou mandar alguns recursos que tem em um protótipo rascunho mas a ideia é descontínua e pegarmos só oq faz sentido, avalie tudo e veja oq podemos usar e onde colocaríamos isso baseado no que já estamos definindo.

Ideias recursos:

Resumo de Recursos, Atividades e Quizzes

Guia rápido para entender o que existe no English Strategies, como cada parte funciona e qual a lógica por trás das atividades e quizzes.

Conteúdos Principais

- Conversation Strategies: lições em slides com emoji/visual/texto + áudio TTS, finalizadas por um quiz e com XP base de 50 quando completadas. F:README.md†L114-L121

F:docs/SISTEMA_XP_MULTIPLICADORES.md†L38-L47

- TPRS Stories: histórias com slides, áudio, contagem de palavras-alvo e conclusão com XP base de 75; após ler, o aluno pode fazer um quiz de circling para reforço.

F:docs/guides/COMO_TESTAR.md†L188-L283

F:docs/SISTEMA_XP_MULTIPLICADORES.md†L38-L47

- Vocabulary + Daily Review:vocabulário é importado via template CSV e serve de base para revisões espaçadas (10 XP por palavra na daily review) usando o algoritmo SRS (SM-2).

F:templates/README.md[†]L22-L185

F:docs/SISTEMA_XP_MULTIPLICADORES.md[†]L38-L46

F:src/utils/srsAlgorithm.ts[†]L12-L97

- Gamificação: streaks, badges automáticos, leaderboards e gráficos de progresso já ativos no dashboard do aluno.

F:README.md[†]L123-L129

Fluxos de Atividade

- Lição de Strategy: escolher uma strategy no dashboard, assistir aos slides com áudio, completar o quiz; ao finalizar, o usuário ganha XP, atualiza streak e vê confetti de conclusão.

F:docs/guides/COMO_TESTAR.md[†]L94-L107

- História TPRS: navegar pelos slides com imagens e destaque de target words, tocar áudio, concluir a história para XP; se houver circling questions, o quiz aparece em seguida.

F:docs/guides/COMO_TESTAR.md[†]L188-L283

- Daily Review (SRS): planejada para rodar com base no algoritmo SM-2; respostas corretas rápidas aumentam o intervalo de revisão, erros reiniciam para 1 dia. F:README.md[†]L142-L148

F:src/utils/srsAlgorithm.ts[†]L12-L97

Tipos de Quiz

- Quiz de Strategy: múltipla escolha com feedback imediato; registra tempo por questão, acertos/erros, dispara confetti no final e pode atualizar o SRS de cada vocábulo associado.

F:src/components/lessons/Quiz.tsx[†]L41-L85

F:src/components/lessons/Quiz.tsx[†]L104-L169

- Circling Quiz (TPRS): respostas abertas focadas em repetição de palavras-alvo, barra de progresso, indicador de dificuldade e feedback com correção; o score final complementa o XP da história.

F:src/components/tprs/CirclingQuiz.tsx[†]L23-L179

F:docs/guides/COMO_TESTAR.md[†]L235-L283

🎮 Lógica de XP e Recompensas

- Bases de XP: lesson 50 XP, quiz 30 XP, daily review 10 XP por palavra, TPRS 75 XP.

F:docs/SISTEMA_XP_MULTIPLICADORES.md[†]L38-L47

- Multiplicadores: streaks de 3 a 100 dias rendem 1.05x–1.5x; bônus fixos para 100% de acerto (+50 XP), primeira lição do dia (+25) e conclusão rápida (+25).

F:docs/SISTEMA_XP_MULTIPLICADORES.md[†]L50-L82

- Feedback visual: modal de breakdown pós-atividade mostra total, bônus e mensagens motivacionais; aparece ao fechar uma lição ou história TPRS.

F:docs/SISTEMA_XP_MULTIPLICADORES.md[†]L90-L124

- Badges e Leaderboards: desbloqueios automáticos por XP, streak, palavras e lições; rankings semanais/mensais/all-time exibem quem mais avançou. F:README.md[†]L123-L129

F:src/utils/badgeDetection.ts[†]L20-L129

🛠 Admin e Importação

- Dashboard Admin: autenticação dedicada, estatísticas e gestão de conteúdo; Strategies Manager pronto com import em massa, demais managers com placeholders prontos para UI rápida.

F:README.md[†]L135-L177

- Templates de Conteúdo: JSON/CSV para strategies, CSV para vocabulary e JSON para TPRS, com regras de nível, fases, slides mínimos e validações automáticas. F:templates/README.md[†]L8-L210

Pensou por 19s >

Beleza — isso aí é ouro porque **não é “ideia solta”**: é um protótipo com peças que já batem com o Motor (Doc 1) + a Fábrica (Doc 2). Vamos avaliar como “Conselho” e mapear **o que entra, onde entra e o que cortar** pra não virar Frankenstein.

Conselho (rodada): avaliação do protótipo “English Strategies”

1) Arq. de Produto (visão sistêmica)

O protótipo já tem **3 blocos que viram formatos oficiais** (arquétipos) do seu sistema:

- **Conversation Strategies** → Aula Interativa (player + checkpoints)
- **TPRS Stories** → Aula Interativa na Estação História (narrativa + checkpoints “circling”)
- **Vocabulary + Daily Review (SRS)** → Missão de Revisão (Estação Revisão)
E por cima:
 - **Gamificação + XP/multiplicadores + Dashboard/Admin/import** → são “camadas de plataforma” (não são conteúdo).

Ou seja: dá pra absorver quase tudo sem mudar nossa filosofia: **Tema→Estação→Dinâmica→Tarefa.**

Mapeamento: “o que é” cada recurso dentro do Motor

2) Cientista Comportamental (estado psicológico)

O que vale muito manter:

- **Daily Review com SRS (SM-2)** é a sua Estação de Revisão perfeita:
 - Dinâmicas: Recall / Matching / Quiz rápido

- Estado: **segurança futura + medo de esquecer**
 - Isso cria retenção "limpa" (não cassino).
 - **Slides + áudio + checkpoints** é exatamente seu modelo de "Aula Interativa":
 - Sem checkpoint vira consumo passivo (entretenimento).
 - Com checkpoint vira **processamento ativo**, ou seja, estado cognitivo real.
- ⚠ Cuidado: **tempo por questão + bônus de velocidade** pode distorcer o estado (vira pressa/ansiedade). Mantém como opcional, não padrão.
-

Economia: XP, streak, badges, ranking (o que fica e como fica)

3) Game Designer / Economia

O protótipo tem um sistema de XP bem completo. A pergunta é: **isso serve a aprendizagem ou vira só grind?**

- ✓ Eu manteria como "mínimo saudável":
- **XP base por arquétipo**
 - Missão (rápida) → base menor
 - Aula Interativa → base média/alta
 - Aplicação (mundo real) → base maior (porque é "resultado real")
 - **Streak + multiplicador leve** (ótimo pra hábito)
 - **Badges automáticos** (ótimo pra identidade)
 - **Breakdown pós-atividade** (feedback significativo)

⚠ Eu controlaria fortemente:

- Leaderboards/Ranking → só em modo TEAM/PRIVATE, e como "liga" (sem humilhação pública).
Ranking é "químico forte".
-

Fluxos: o que reaproveitar exatamente

4) Arq. de Dados/Engenharia (onde encaixa no Creator Loop)

O que você descreveu já é quase um "Creator Loop" pronto, só falta oficializar como **formatos de produção**.

Onde colocar cada coisa:

A) Conversation Strategies

Arquétipo: Aula Interativa

Estação típica: Prática (ou Desafio se tiver tensão/status)

Dinâmicas: Quiz / Escolha / Recall (checkpoints)

Tarefa: sequência de cards (beats) + quiz final + XP + streak + feedback

Isso é literalmente um "Blueprint de Aula Interativa" no estilo slide-player.

B) TPRS Stories

Arquétipo: Aula Interativa (em História)

Estação: História

Dinâmicas: Simulação leve / Escolha / Circling (que é recall guiado)

Tarefa: narrativa em beats + reforço de target words + quiz ao final

Aqui a emoção vem do "significado/identificação", não da pressão.

C) Vocabulary + Daily Review (SRS)

Arquétipo: Missão (rápida)

Estação: Revisão

Dinâmicas: Recall / Matching / Quiz curto

Tarefa: revisão espaçada com SM-2 (intervalo aumenta quando acerta, reinicia quando erra)

Isso é uma “máquina de retenção” que você pode aplicar em qualquer tema que tenha “itens revisáveis”.

D) Gamificação (streak/badges/leaderboards/gráficos)

Camada: Economia e Feedback

Aplicação: transversal a todos os arquétipos

E) Admin + Importação + Templates (JSON/CSV)

Camada: Creator Loop / Linha de montagem

Aplicação: o criador não mexe em tudo; ele escolhe “formato oficial” e alimenta template.

O que cortar ou rebaixar de prioridade (pra não complicar a fábrica)

5) Go-To-Market (produto pra criador)

Se a promessa é “transformar ideia em edtech”, o criador não pode enfrentar 20 opções. Eu reduziria para:

Formatos oficiais (poucos):

1. Aula Interativa (slides + áudio + checkpoints)
2. Story Interativa (narrativa + checkpoints)

3. Review SRS (itens revisáveis)

4. Aplicação (missão no mundo real + prova)

E todo o resto vira "configuração avançada".

⚠️ Eu rebaixaria:

- bônus por velocidade (pode gerar comportamento ruim)
 - excesso de multiplicadores e exceções (vira cassino/grind)
 - ranking público
-

Tabela final: "recurso do protótipo → onde entra no sistema"

Recurso do protótipo	No Motor (Doc 1)	No Produto (Doc 2)	Eu usaria como
Conversation Strategies (slides+áudio+quiz)	Estação Prática/Desafio + Dinâmicas Quiz/Recall	Aula Interativa	Formato oficial #1
TPRS Stories (história + circling)	Estação História + Dinâmicas Escolha/Recall guiado	Aula Interativa (Story)	Formato oficial #2
Daily Review com SM-2	Estação Revisão + Dinâmica Recall/Matching	Missão diária de retenção	Formato oficial #3
XP base + multiplicadores	Economia/Feedback	Retenção e hábito	Manter, simplificar
Confetti + breakdown pós-atividade	Feedback significativo	"sensação de avanço real"	Manter

Recurso do protótipo	No Motor (Doc 1)	No Produto (Doc 2)	Eu usaria como
Badges + gráficos	Identidade/Progresso	Colecionáveis	Manter
Leaderboards	Status/social	Opcional perigoso	Só TEAM/PRIVATE
Admin + import + templates	Creator Loop	Linha de montagem 80 apps	Manter (núcleo da fábrica)

O “pulo do gato” pra você reaproveitar isso em QUALQUER nicho

English Strategies é só um caso do formato.

O que você está reaproveitando é:

- **Blueprints oficiais** (Aula / Story / Review / Aplicação)
- **Economia transversal** (XP/streak/badges)
- **Importação por templates** (creator loop de produção)

Isso é exatamente “edtech factory”.

Se você quiser, eu sigo a partir daqui com a próxima peça que torna isso “existente de verdade”:

Entregável da próxima rodada (escolho o mais útil):

- **Lista oficial dos “Formatos Oficiais”** (4 formatos) +
- **Blueprint de produção de cada um** (em linguagem humana, estilo checklist) +
- **Quais estações e dinâmicas cada formato permite** (governança simples)

Só me diga se você quer que eu comece por:

1. Formatos Oficiais + governança (mais produto)

OU

2. Blueprints de produção (mais fábrica / operacional).



Sabe toda a tabela; Tabela final: "recurso do protótipo → onde entra no sistema"

Claro — aqui está **toda a tabela** "recurso do protótipo → onde entra no sistema":

Recurso do protótipo	No Motor (Doc 1)	No Produto (Doc 2)	Eu usaria como
Conversation Strategies (slides + emoji/visual/texto + áudio TTS + quiz + XP)	Estação: Prática (ou Desafio) • Dinâmicas: Quiz / Escolha / Recall • Tarefa: Aula em beats + checkpoint + quiz final	Aula Interativa (player + checkpoints)	Formato oficial #1 (Aula Interativa)
TPRS Stories (histórias em slides + áudio + target words + circling quiz)	Estação: História • Dinâmicas: Escolha / Recall guiado (circling) / Simulação leve • Tarefa: narrativa em beats + checkpoints	Aula Interativa (Story)	Formato oficial #2 (Story Interativa)
Vocabulary + Daily Review (SRS / SM-2) (10 XP por palavra + espaçamento)	Estação: Revisão • Dinâmicas: Recall / Matching / Quiz curto •	Missão diária de retenção	Formato oficial #3 (Review SRS)

Recurso do protótipo	No Motor (Doc 1)	No Produto (Doc 2)	Eu usaria como
	Tarefa: revisão espaçada (intervalo ↑ acerto, reinicia erro)		
Gamificação (streaks, badges automáticos, leaderboards, gráficos)	Camada transversal: Atração/engajamento via tensão calibrada (streak = aversão à perda; badges = identidade; ranking = social/status)	Economia do app (progresso visível + retorno diário)	Manter (com governança: ranking opcional/controlado)
Fluxo Strategy (escolhe no dashboard → slides → quiz → XP/streak/confetti)	Dinâmica força ação; Tarefa fecha ciclo com feedback e recompensa	Core Loop do usuário	Core Loop padrão para o infoapp
Fluxo Story TPRS (slides + áudio + target words → XP → circling opcional)	História ancora valor + checkpoints garantem processamento ativo	Core Loop (variante Story)	Manter como "modo história"
Daily Review planejada (SM-2) (tempo/resposta altera intervalo)	Revisão = segurança futura; o algoritmo sustenta consistência	Rotina diária (retenção limpa)	Manter como "pilar de retenção"
Quiz de Strategy (múltipla escolha, feedback imediato, tempo por questão, confetti, pode atualizar SRS)	Dinâmica: Quiz • Tarefa: variações SAFE/TENSION/STATUS via tempo/feedback/consequência	Mecanismo de fixação	Manter (tempo/velocidade como opcional)
Circling Quiz (TPRS) (aberto, repetição de target words, progresso, dificuldade, correção)	Dinâmica: Recall guiado / repetição ativa (processamento ativo)	Fixação pós-história	Manter como "checkpoint narrativo"

Recurso do protótipo	No Motor (Doc 1)	No Produto (Doc 2)	Eu usaria como
Bases de XP (lesson 50, quiz 30, daily 10/palavra, TPRS 75)	Camada de progresso (não é pedagogia)	Economia	Manter, mas recalibrar por arquétipo (missão/aula/aplicação)
Multiplicadores (streak 1.05x–1.5x, 100% acerto +50, primeira do dia +25, rápido +25)	Modula tensão/aversão à perda e reforço	Reforço comportamental	Simplificar (evitar “grind” e ansiedade)
Modal breakdown pós-atividade (total + bônus + motivacional)	Feedback significativo (identidade, clareza)	Fechamento do loop	Manter (padrão obrigatório)
Badges e Leaderboards (XP/streak/palavras/lições; ranking semanal/mensal/all-time)	Badges = identidade ✅ · Ranking = status⚠️	Camada social/competitiva	Badges: manter · Ranking: só TEAM/PRIVATE
Dashboard Admin + importação em massa	Governança + produção	Creator Loop / Fábrica	Manter (núcleo)
Templates de conteúdo (JSON/CSV) (validações, níveis, fases, slides mínimos)	“Tarefa composta” + regras	Linha de montagem	Manter como padrão oficial de criação

Se quiser, eu transformo essa tabela em uma versão mais “constitucional” (com nomes oficiais e regras de permissão por formato) pra já virar documento do produto.



Gostei da ideia de organizar em tabelas, acho bom organizar tudo em tabelas cada membro pode fazer isso, de organizar suas ideias em tabelas e acho que por todos os membros serem experientes, todos podem trazer 1 ou 2 pessoas mais adequadas para criar tabelas em áreas relevantes

Fechado. Vamos operar daqui pra frente assim: **cada membro entrega suas contribuições em 1–2 tabelas**, e cada um puxa **1–2 especialistas** só pra ajudar a “tabular” as áreas que normalmente viram bagunça (economia, governança, produção, qualidade, etc.).

Abaixo está a **rodada 1 de tabelas**, já com **novos convidados**.

Conselho + convidados (para tabular)

- Arq. de Produto
 - convidado: PMO/Operações de Produto (pra transformar em “chão de fábrica”)
- Cientista Comportamental
 - convidado: Neuro/Aprendizagem (pra validar processamento ativo)
- Game Designer/Economia
 - convidado: Economista de jogos / LiveOps (pra simplificar XP/multiplicadores)
- Arq. de Dados/Engenharia
 - convidado: Arquiteto de Conteúdo / CMS (templates e import)
- Go-To-Market

- convidado: **Especialista em Creator Monetization** (produto vendável pro criador)



Escriba segue anotando.

1) Arq. de Produto — “Constituição do Sistema” (tabela)

Camada	Definição (oficial)	Estável ou variável?	Exemplo	Regra de ouro
Tema	Verdade neutra (conhecimento)	Estável	“Copy: Big Idea”	Tema não muda por formato
Estação	Intenção psicológica	Estável	Prática / História / Desafio / Revisão / Avaliação	Estação não é mídia
Dinâmica	Forçador cognitivo	Estável	Quiz / Recall / Simulação...	Dinâmica pressiona ação
Tarefa	Experiência concreta	Variável	missão, aula interativa, aplicação	Variação mora aqui
Formato Oficial	“Embalagem” de tarefa (arquétipo)	Estável (poucos)	Missão / Aula Interativa / Aplicação (+ Story/Review como variações)	Poucos formatos = escala
Economia	XP/streak/badges/recom pensa	Estável com calibração	XP base, multiplicadores	Serve o hábito, não o grind

Camada	Definição (oficial)	Estável ou variável?	Exemplo	Regra de ouro
Creator Loop	Linha de montagem	Estável	templates → build → publish	Criador escolhe formato, não "constrói app"

2) PMO/Operações — “Cadências e entregáveis” (tabela)

Sprint	Entregável	Saída em tabela	Dono	Critério de pronto
S0	Vocabulário + Constituição	Tabela de definições + regras	Produto	1 página, sem ambiguidades
S1	Formatos Oficiais	Tabela “Formato→Blueprint”	Produto/Eng	4 formatos no máx.
S2	Matriz de Permissão	Tabela “Estação×Dinâmica”	Comportamental	✓ /⚠️ /🚫 fechado
S3	Economia mínima	Tabela XP/multiplicadores	Game/Econ	simples, não cassino
S4	Templates (Creator)	Tabela de inputs/outputs	Eng/CMS	import fácil
S5	Qualidade + publicação	Tabela de QA + gates	Produto/PMO	bloqueia apps ruins

3) Cientista Comportamental — “Estados & Tension Dial” (tabela)

Tension Profile	Objetivo psicológico	O que permite	O que evita	Sinais de sucesso
SAFE	reduzir ansiedade / controle	sem tempo; feedback explicativo	punição, ranking	conclusão alta, baixa evasão
TENSION	foco / risco simbólico	tempo moderado; consequência leve	vergonha pública	atenção alta, retorno amanhã
STATUS	validação / ego	ranking controlado (TEAM/PRIVATE), badges	exposição pública	prova interna, orgulho

4) Neuro/Aprendizagem — “Aula Interativa em Beats” (tabela)

Elemento	Padrão	Por quê
Beats	8 beats de 45–90s	mantém ritmo sem virar “vídeo longo”
Checkpoint	1 por beat (escolha/quiz/recall/simulação)	força processamento ativo
Finalização	resumo + ação aplicada	transferência pro mundo real
Falha do sistema	aula sem checkpoint	vira entretenimento/passivo

5) Game Designer – “Economia mínima saudável” (tabela)

Arquétipo	XP base (sugestão)	Bônus permitidos	Bônus a evitar
Missão (rápida)	10–20	primeira do dia, streak	velocidade agressiva
Aula Interativa	50–80	100% checkpoints	grind de multiplicadores
Aplicação (mundo real)	120–200	prova enviada, consistência semanal	comparação pública

Ranking: só TEAM/PRIVATE por liga/segmento.

6) Economista de jogos/LiveOps – “Multiplicadores (simplificado)” (tabela)

Item	Regra simples	Motivo
Streak	1.05x (3d), 1.1x (7d), 1.2x (30d)	reforço sem vício
First action/day	+ bônus fixo pequeno	habito
Perfect completion	bônus fixo	qualidade
Speed bonus	opcional e leve	não criar ansiedade

7) Arq. de Dados/Engenharia — “Entidades base” (tabela)

Entidade	Serve pra	Relaciona com
InfoApp	produto publicado	Temas, trilhas, economia
Tema	conteúdo neutro	Estações, tarefas
Estação	intenção psicológica	permite dinâmicas
Dinâmica	gatilho cognitivo	usada em tarefas
Tarefa	unidade executável	formato, XP, feedback
Recompensa	economia	badges/loja/unlocks

8) Arq. CMS (convidado) — “Templates de produção (Creator Loop)” (tabela)

Formato oficial	Entrada do criador	Saída da fábrica
Aula Interativa	roteiro/transcrição + objetivo + nível + (assets opcional)	cards com narração + visual + checkpoints + resumo/ação
Story Interativa	história + target points	cards narrativos + checkpoints “circling”
Review SRS	CSV itens revisáveis	agenda SM-2 + missões diárias

Formato oficial	Entrada do criador	Saída da fábrica
Aplicação	checklist de tarefa + prova	missão no mundo real + upload/validação

9) Go-To-Market — “Como vender pro criador” (tabela)

Dor do criador	Promessa	Prova	Oferta simples
curso não retém	vira hábito diário	streak + progresso + conclusão	“seu Duolingo do seu tema”
criar app é difícil	poucos formatos oficiais	templates + publish	“só entra com conteúdo”
qualidade varia	governança + gates	QA automático	“não publica app ruim”

✓ O que eu entendi que você quer com “tabelas”

- “Tabelas” viram o **índio oficial** do conselho.
- A gente passa a “fechar decisões” sempre em tabelas (não em texto longo).

[resumo]

- Decisão de processo: **o sistema será organizado e evoluído via tabelas** (cada membro contribui em 1–2 tabelas por rodada).
- Conselho pode **chamar 1–2 especialistas adicionais** por rodada para tabular áreas específicas (economia, governança, produção, qualidade etc.).

Se você mandar o próximo bloco de ideias/recursos do protótipo, eu já encaixo **na mesma estrutura de tabelas** e a gente vai fechando o sistema peça por peça.



Bom pessoal, nossa discussão está produtiva. Qual o nosso próximo passo? Falta algo para fechar os esse sistema completo e interface?

Sim — falta. E dá pra fechar isso de um jeito bem “conselho experiente”: **o conceito já está forte**, agora precisamos transformar em **sistema completo + interface publicável** com 6 fechamentos inevitáveis.

Abaixo vai o **próximo passo** em tabelas, com “o que falta” + “ordem” + “o que é pronto”.

1) Arq. de Produto — O que falta pra “fechar o sistema” (tabela)

Bloco	O que falta fechar	Por quê é crítico	Saída (artefato)
A. Formatos Oficiais	lista final de 4 formatos + nomes	reduz complexidade pro criador	Tabela “Formato → Blueprint”
B. Matriz de Permissão	Estação x Dinâmica	governança do motor psicológico	Tabela de permissão final
C. Tension Dial	SAFE/TENSION/STATUS como campo obrigatório em tarefa	controla o estado emocional	Regras + defaults por estação

Bloco	O que falta fechar	Por quê é crítico	Saída (artefato)
D. Economia mínima	XP base + bônus permitidos	evita grind/cassino	Tabela "XP & bônus"
E. Creator Loop (produção)	templates oficiais + pipeline de publicação	escala 80 apps	Checklist de produção
F. Interface (UX)	3 telas-chave do aluno + 3 do criador	sem interface não existe	Mapa de telas (tabela)

2) UX Habit Products — O que falta pra “fechar a interface” (tabela)

A) Interface do Aluno (3 telas que fazem o produto existir)

Tela	Objetivo	Elementos obrigatórios	Critério de pronto
Home (Missão do dia)	retorno diário	missão + streak + XP + “voltar amanhã”	usuário completa em <60s
Player (Aula Interativa)	aprendizado ativo	beats + checkpoint por beat + progresso	sem checkpoint = bloqueia avanço
Progresso/Mapa	“onde estou?”	trilha + badges + gráficos simples	clareza de próximo passo

B) Interface do Criador (3 telas que fazem a fábrica existir)

Tela	Objetivo	Elementos obrigatórios	Critério de pronto
Create InfoApp	iniciar produto	nome + nicho + formato oficial	criado em 2 min
Import/Build	gerar conteúdo	upload roteiro/transcrição + nível	preview de cards
Publish/QA	qualidade + release	checklist + preview + "publish"	não publica app ruim

3) PMO/Operações — Ordem correta do próximo passo (tabela)

Ordem	Fechamento	Por que vem agora	Tempo lógico
1	Formatos Oficiais	define o que o criador pode fazer	base do resto
2	Blueprints por formato	transforma visão em produção	"linha de montagem"
3	Matriz Estação×Dinâmica	governa o motor	evita erros psicológicos
4	Economia mínima (XP/streak/badges)	retenção saudável	sem cassino
5	UX: 6 telas-chave	materializa produto	interface publicável
6	Gates de qualidade (QA)	protege escala	impede apps ruins

4) Arq. de Dados/Engenharia — “o mínimo técnico que fecha o sistema” (tabela)

Item	Precisamos fechar	Resultado
Modelo de dados	entidades + campos obrigatórios (tension, estação etc.)	motor implementável
Templates	formatos de entrada e validação	creator loop escalável
Geração de cards	regras de split em beats + checkpoints	aula interativa real
Eventos/analytics	task_started/completed/failed/hint_used	otimização e QA

✓ Próximo passo (decisão única)

Se você quer que isso “vire produto” rápido e completo, o próximo passo é:

Fechar os Formatos Oficiais + Blueprints

Porque é isso que:

- torna o Creator Loop real
- torna a interface desenhável
- permite o time construir sem ambiguidade

O que eu faço já na próxima mensagem (sem enrolar)

Eu entrego uma tabela “final” com:

1. 4 Formatos Oficiais (nomes oficiais)
2. Blueprint de produção de cada um (inputs → outputs)
3. Estação padrão + dinâmicas permitidas por formato
4. XP base sugerido por formato (simples)

E isso já vira “documento mestre” pra produto/UX/dev.

[resumo]

- Para fechar sistema + interface faltam 6 fechamentos: Formatos Oficiais; Blueprints; Matriz de Permissão; Tension Dial obrigatório; Economia mínima; Interface (3 telas aluno + 3 criador) + QA gates.



Ok, precisamos pensar em cada tela e detalhe, primeiro vamos listar todas as telas e tudo que precisa ter em cada tela,

Pensou por 12s >

Fechado. Vou listar **todas as telas** (Aluno + Criador/Admin) e **o que cada uma precisa ter** — em tabelas, já no formato “chão de fábrica”. (Depois a gente entra tela por tela pra desenhar detalhes, estados e micro-interações.)

1) App do Aluno (Learner) – mapa completo de telas

1.1 Telas essenciais do MVP

Tela	Objetivo	Precisa ter (mínimo)	Precisa ter (detalhes)
Login / Cadastro	entrar no app	email/social, termo, recuperar senha	bloqueio anti-fraude básico, "continuar de onde parou"
Onboarding rápido	posicionar + reduzir atrito	objetivo, nível, rotina (min/dia)	escolher "estilo": SAFE/TENSION (opcional), tutorial de streak
Home (Missão do Dia)	retorno diário	missão do dia (1 CTA), streak, XP do dia	"começar agora" em 1 clique, preview de recompensa, fallback se missão já feita
Mapa/Trilha	clareza do caminho	trilha com etapas, bloqueios/desbloqueios	"próxima etapa", indicação de estação (Prática/História/...)
Player de Aula Interativa	aprender ativo	beats (8), progresso, áudio + visual	checkpoint por beat (obrigatório), botão repetir, legenda opcional
Checkpoint/Quiz (overlay)	forçar ação	escolha/quiz/recall/simulação	feedback imediato (EXPLAIN/COACHING/REVEAL), tentativa/ayuda
Resumo + Ação Aplicada	transferência	resumo (3 bullets), "ação 1% hoje"	botão "marcar como feito", opcional "prova"/nota
Review (SRS)	retenção	lista do dia, cards de revisão	acerto/erro, timer opcional leve, "voltar amanhã"

Tela	Objetivo	Precisa ter (mínimo)	Precisa ter (detalhes)
Aplicação (mundo real)	resultado real	tarefa, checklist, campo de prova	upload/link/nota, rubrica simples, status "pendente/aprovado"
Tela de Conclusão (Breakdown)	reforço e fechamento	XP total, bônus, streak update	confetti, badge/unlock, mensagem motivacional "o que isso revela"
Progresso	"onde estou?"	nível, streak, XP semanal	gráficos simples, etapas concluídas, próximas metas
Badges / Galeria	identidade	lista de badges + condições	badges recentes, próximos badges ("quase lá")
Recompensas / Loja	economia	saldo/moeda, catálogo	resgate, histórico, regras claras
Perfil/Configurações	controle	notificações, idioma, privacidade	reset onboarding, export básico, suporte

1.2 Telas opcionais (V2+)

Tela	Objetivo	Precisa ter
Leaderboard/Ligas	status social	modo TEAM/PRIVATE, ligas por segmento, sem humilhação
Feed (short tips)	exploração leve	vídeos/cards curtos, salvar, "ver mais"
Biblioteca	acesso ao histórico	aulas, histórias, aplicações concluídas

Tela	Objetivo	Precisa ter
Notificações	retenção	lembretes, streak em risco, unlocks

2) Studio do Criador/Admin — mapa completo de telas

2.1 Telas essenciais do MVP (fábrica existir)

Tela	Objetivo	Precisa ter (mínimo)	Precisa ter (detalhes)
Login Admin / Workspace	entrar + organizar	autenticação, selecionar workspace	perfis: owner/editor/reviewer
Dashboard de Apps	ver tudo	lista de infoapps, status (draft/published)	métricas rápidas: DAU, completion, streak avg
Criar InfoApp (Wizard)	iniciar produto	nome, descrição, nicho, idioma	branding básico, público-alvo, nível inicial
Escolher Formato Oficial	reduzir complexidade	(Aula, Story, Review SRS, Aplicação)	exemplos + "quando usar", limitações claras
Importer / Build (Aula Interativa)	gerar cards	colar roteiro / subir transcrição	definir objetivo + nível, preview em beats, checkpoints sugeridos
Importer / Build (Story)	narrativa	colar história + target points	contagem de palavras-alvo (opcional), circling checkpoints

Tela	Objetivo	Precisa ter (mínimo)	Precisa ter (detalhes)
Importer / Build (SRS Review)	subir itens	upload CSV itens revisáveis	validação, tags/nível, agendamento SRS
Builder de Aplicação	tarefas reais	template: objetivo, checklist, prova	rubrica, critérios de aceite, tempo sugerido
Preview do Aluno	ver como fica	simular Home/Trilha/Player	simular SAFE/TENSION/STATUS
QA / Checklist de Publicação	qualidade	checklist obrigatório (sem checkpoint não publica)	lint de conteúdo, "tension dial" consistente
Publish / Versioning	lançar	publicar, rollback básico	versão v1/v2, changelog
Analytics básico	medir	conclusão, drop-off por beat, acerto checkpoint	heatmap simples "onde cai"
Gestão de Recompensas (simples)	economia	criar badges, configurar XP base	catálogo simples, regras de resgate
Configurações do InfoApp	controle	nome, branding, domínio	integrações: ElevenLabs (chave), assets

2.2 Telas opcionais (V2+)

Tela	Objetivo	Precisa ter
Assets Manager	organizar mídias	upload, tags, versões

Tela	Objetivo	Precisa ter
A/B de tarefas	otimizar	variações SAFE/TENSION/STATUS, métricas comparativas
Usuários/Comunidade	suporte	cohort view, suporte, feedback
Automations	retenção	regras: reminder, streak risk, "volta amanhã"
Billing/Planos	negócio	planos por apps/MAU, upgrades
Roles avançadas	time	permissões finas, auditoria

3) “O que precisa ter” por tela – checklist transversal (vale para TODAS)

Categoria	Precisa ter
Estados	loading, empty state, erro, offline (opcional), "sem missão hoje"
Navegação	back consistente, salvar progresso, retomar onde parou
Feedback	micro-feedback (acertou/errou), macro-feedback (breakdown final)
Regras do motor	estaçao e tension dial visíveis (interno), checkpoints obrigatórios em aula
Telemetria mínima	started/completed/failed, hint_used, drop_off_beat, time_spent

Categoria	Precisa ter
Governança	bloqueios de publicação se violar blueprint (ex: aula sem checkpoints)

Próximo passo (pra avançar sem dispersar)

Agora que o “mapa” está listado, o melhor próximo passo é **fechar 2 coisas em sequência:**

- 1. IA (Information Architecture) final:** quais telas entram no MVP vs V2+ (pra não explodir escopo)
- 2. Detalhar 3 telas núcleo do aluno** (Home, Player, Conclusão) e **3 do criador** (Create, Build/Import, QA/Publish) com estados e componentes.

Se você mandar “vamos de MVP”, eu já devolvo a **lista enxuta (MVP)** + a **ordem de implementação** em tabela.



Chame para o conselho as pessoas mais adequadas para avaliar as páginas e UX ui, é baseado em tudo que foi discutido revisar essa etapa e ajustar oq for necessário m. E acho importante termos claro que não vamos trabalhar com MVP, vamos fazer o trabalho completo e final

Pensou por 8s >

Fechado — **sem MVP**. Vamos tratar como **produto final (v1 completo)**, com todas as telas, flows e governança fechados de forma publicável.

Abaixo eu **chamo o conselho ampliado** (os perfis mais adequados pra UX/UI) e já faço a **revisão crítica** do mapa de telas que listamos — ajustando o que precisa pra ficar coerente com:

- Motor (Tema→Estação→Dinâmica→Tarefa)
 - Formatos Oficiais
 - Aula Interativa em beats + checkpoints
 - Creator Loop (linha de montagem)
 - Escala (80+ apps) conteúdo-agnóstico
-

Conselho ampliado (UX/UI) — “só gente de produto final”

Núcleo (5):

1. Head of Product Design (EdTech/Creator Tools)
2. UX Architect (IA + navegação)
3. Design Systems Lead (UI escalável + componentes)
4. Growth & Retention Designer (loops, notificações, economia saudável)
5. UX Writer / Content Designer (microcopy, instruções, feedback)

Convidados especializados (para fechar “final”):

- 6) Accessibility & Inclusive Design (WCAG, legibilidade, motion-safe)
- 7) Motion/Interaction Designer (beats, transições, confetti, microfeedback)
- 8) Research/Insights (Jobs-to-be-Done) (criador e aluno)

9) Anti-abuse/Trust UX (ranking, prova, upload, fraude, moderação)

10) B2B/Admin UX (workspaces, roles, billing, auditoria)

1) Revisão do mapa de telas – ajustes necessários (produto final)

1.1 Problema do mapa anterior (diagnóstico do Conselho)

Ele estava bom, mas “parecia MVP” porque:

- misturava “telas” com “funcionalidades” (ex: checkpoint overlay é componente, não tela)
- não fechava **Workspaces/Roles, Billing, Versionamento, QA gates, Moderation, Distribuição/Store**
- não explicitava **Navegação oficial** (como o usuário circula)

Ajuste: vamos organizar em **3 Apps** (Learner / Creator Studio / Platform Admin) + **camadas transversais**.

2) Arquitetura final de informação (IA) – visão completa (tabela)

2.1 Learner App (Aluno) – seções finais

Seção	Telas	Observações UX (final)
Entrada	Login, Signup, Recuperar, Consentimentos	login social + SSO opcional
Onboarding	Objetivo, Nível, Rotina, Preferências	“safe by default”; notifs opt-in

Seção	Telas	Observações UX (final)
Core Diário	Home (Missão do Dia), "Continuar", "Próximo Passo"	1 CTA dominante
Conteúdo	Player Aula Interativa, Player Story, Review SRS, Aplicação	Player único com "modes"
Progresso	Mapa/Trilha, Progresso, Badges/Galeria, Histórico	"onde estou / o que falta"
Economia	Recompensas/Loja, Carteira/Saldo, Histórico de resgates	regras claras e anti-frustração
Social (governado)	Ligas/Ranking (TEAM/PRIVATE), Equipe/Grupo	sem vergonha pública
Feedback & Suporte	Feedback rápido, Reportar problema, Ajuda/FAQ, Contato	dentro do fluxo, não escondido
Conta	Perfil, Configurações, Privacidade, Notificações	controle total

 **Mudança importante:** "Checkpoint/Quiz overlay" vira **componente do Player**, não tela solta.

2.2 Creator Studio (Criador) — seções finais

Seção	Telas	Observações UX (final)
Workspace	Login, Selecionar workspace, Criar workspace	multi Produto, multi Marca
Dashboard	Visão geral, Apps (lista), Atalhos (criar/importar/publish)	métricas de produto
Criar InfoApp	Wizard: nome, nicho, idioma, branding, público	criador não "monta app"
Build (Linha de montagem)	Import/Builder por Formato Oficial	templates e validação fortes
Preview	Preview learner (Home/Player/Trilha)	preview por perfil (SAFE/TENSION/STATUS)
QA & Governança	Checklist, Warnings, Bloqueios, "Corrigir agora"	gates obrigatórios (ex: aula sem checkpoint bloqueia)
Publish & Versões	Publish, Versioning, Changelog, Rollback	release profissional
Conteúdo & Biblioteca	Content Library, Assets, Reuso de blocos	escala (80 apps)
Economia & Recompensas	XP rules, Badges, Loja	presets + avançado
Analytics	Funil por beat, drop-off, cohorts, heatmaps	otimização contínua

Seção	Telas	Observações UX (final)
Usuários & Comunidade	usuários, cohorts, feedback, suporte	moderação e insights
Configurações	domínio, integrações (TTS), notificações, SEO/Store	distribuição
Billing	plano, limites, faturas	v1 completo exige isso
Roles & Audit	permissões, logs, trilha de auditoria	obrigatório pra escalar time

2.3 Platform Admin (Operador da plataforma) — seções finais

Seção	Telas	Pra quê
Tenants/Workspaces	gestão global	suporte e compliance
Moderation/Trust	denúncias, bloqueios, revisão de conteúdo	evitar abuso
Catálogo/Marketplace	listagem de infoapps	descoberta
Infra/Status	saúde do sistema	operação
Pagamentos	planos, cobrança, chargebacks	negócio
Auditoria	logs e incidentes	segurança

3) Navegação oficial (para não virar “app labirinto”)

3.1 Navegação do Aluno (tabela)

Padrão	Tabs principais	Entradas rápidas	Regra
Bottom tabs (5)	Home • Trilha • Review • Progresso • Perfil	“Continuar” sempre visível	1 CTA dominante por tela
Player	modal/stack interno	checkpoint dentro do player	checkpoint bloqueia avanço

3.2 Navegação do Criador (tabela)

Padrão	Menu lateral	Barra superior	Regra
Sidebar (8–10)	Apps • Build • Preview • QA • Publish • Analytics • Users • Rewards • Settings • Billing	workspace switch + search	“Create → Build → QA → Publish” sempre linear

4) Componentes UI obrigatórios (Design System) — produto final

Componente	Onde usa	Observação
Mission Card	Home	estado: pendente/concluída
Progress Bar (beats)	Player	por beat e total

Componente	Onde usa	Observação
Checkpoint Module	Player	escolha/quiz/recall/simulação
Feedback Module	Player/Conclusão	EXPLAIN / COACHING / REVEAL
XP Breakdown Modal	final	obrigatório
Streak Widget	Home/Progresso	"em risco" com notificação
Badge Toast + Gallery	app todo	identidade
SRS Card	Review	acerto/erro e intervalo
QA Gate Banner	Studio	"bloqueado até corrigir"
Template Importer	Studio	CSV/JSON + validação
Preview Emulator	Studio	simula telas do aluno

5) Ajustes recomendados pelo Conselho (o que mudar agora)

5.1 Ajustes de estrutura (tabela)

Item	Antes	Agora (ajuste final)	Por quê
Aula/Story	telas separadas	1 Player com modos	consistência + escala
Checkpoint	"tela"	componente do Player	reduz complexidade

Item	Antes	Agora (ajuste final)	Por quê
Review	opcional	seção fixa	retenção estrutural
Aplicação	"extra"	formato oficial	resultado real
QA/Publish	"pode depois"	obrigatório	protege a fábrica
Billing/Roles	v2	v1 completo	produto final exige

6) Próximo passo (concreto) — sem MVP, mas com ordem inteligente

Agora que as telas estão "completas", o próximo passo é **especificar cada tela** com:

- **Componentes**
- **Estados** (loading/empty/error/success)
- **Ação primária**
- **Regras do motor** (estaçao/tension/checkpoint)
- **Eventos (analytics)**

Entregável imediato da próxima rodada

Eu posso te devolver, em tabela, o detalhamento completo das **6 telas núcleo** (que definem a experiência final):

Aluno: Home • Player • Conclusão

Criador: Create InfoApp • Build/Import • QA/Publish

E depois a gente expande pra demais com o mesmo template.

Só me diga: você quer começar detalhando **Aluno** ou **Criador** primeiro?



Ok, vamos organizar então, pq é aplicação, oq é a aba do creator, e pensa num modelo para importação em massa, numa linguagem simples que pode ser feito em volume usando ia, salva num formato específico e já sobe e assim a aplicação funciona

Pensou por 22s >

Perfeito — vamos organizar isso como **3 definições fechadas + um modelo de importação em massa** que dá pra gerar com IA e subir em volume sem dor.

1) O que é “Aplicação”

Aplicação é o arquétipo de tarefa que força transferência pro mundo real (resultado), com prova. Ela não é “uma aula” e não é “um quiz”. Ela é o momento em que o usuário faz algo fora do app e volta com evidência.

Aplicação, na prática (tabela)

Peça	O que é	Pra que serve
Brief	contexto + objetivo em 3 linhas	clareza imediata
Checklist	3–7 passos	reduzir incerteza
Entrega	"o que enviar/registrar"	prova objetiva
Prova	texto / link / upload / áudio / screenshot	validação do mundo real
Rubrica	critérios simples (0/1 ou 0–2)	julgamento consistente
Feedback	coaching + "o que isso revela sobre você"	identidade + evolução
Recompensa	XP alto + badge (às vezes)	reforço de valor real

Tipos de Aplicação (pra ficar completo)

Tipo	Duração	Prova típica	Estação mais comum
Aplicação Rápida	5–10 min	texto curto	Prática
Aplicação de Campo	20–60 min	screenshot/link/áudio	Desafio / Avaliação
Projeto	2–7 dias	entregável + antes/depois	Avaliação

2) O que é "Aba do Creator"

"Aba do Creator" é o **modo Studio** (painel do criador).

Pra produto final, eu recomendo isso existir como **um espaço separado dentro da mesma plataforma**, acessível por um botão tipo "**Modo Criador**" (no Perfil do usuário) — sem misturar com a navegação do aluno.

Aba do Creator (seções finais)

Seção	O que tem	Pra que serve
Meus InfoApps	lista + status (draft/published)	controlar portfólio
Criar InfoApp	wizard rápido	iniciar produto
Importar / Build	upload de pacote + validação	linha de montagem
Preview	simula Home/Player/Trilha	ver como o aluno verá
QA / Publicação	checklist + bloqueios	impedir app ruim
Versões	changelog + rollback	evolução profissional
Analytics	drop por beat, conclusão, coortes	otimizar retenção
Recompensas	XP/badges/loja	economia do app
Configurações	branding, domínio, integrações TTS	operacional
Workspace/Roles/Billing	time + permissões + cobrança	escala real

3) Modelo de importação em massa (simples, em volume, com IA)

A ideia é: o criador (ou a IA dele) gera um **Pacote InfoApp** em um formato único, sobe, a plataforma valida, pré-visualiza e já “vira app”.

Formato recomendado: Pacote InfoApp (.zip)

Porque permite volume + assets + organização (e é fácil pra IA gerar).

Estrutura do pacote:

bash

 Copy code

```
infoapp_pack.zip
  manifest.yaml
  tracks.yaml
  lessons/lesson_001.yaml
  lessons/lesson_002.yaml
  missions/mission_001.yaml
  applications/apply_001.yaml
  srs/vocab.csv
  assets/
    img_001.svg
    img_002.png
```

O que vai em cada arquivo (linguagem humana)

Arquivo	Conteúdo	Observação
manifest.yaml	identidade do app + idioma + níveis + regras básicas	“capa” do infoapp

Arquivo	Conteúdo	Observação
tracks.yaml	trilhas/módulos e ordem	mapa do progresso
lessons/*.yaml	Aula Interativa em beats + checkpoints	segue regra dos beats
missions/*.yaml	missões rápidas	habito diário
applications/*.yaml	aplicações com prova + rubrica	resultado real
srs/vocab.csv	itens revisáveis (SRS)	retenção estrutural
assets/	imagens/svg/anim	opcional

4) Especificação mínima do arquivo de Aula Interativa (o mais importante)

Exemplo (bem simples, ideal pra IA gerar):

yaml

 Copy code

```

id: lesson_001
title: "Como criar uma Big Idea que prende atenção"
theme: "VSL / Big Idea"
station: "PRATICA"
tension: "SAFE"
xp_base: 60

beats:
  - beat: 1

```

```
narration: "A Big Idea é a promessa com mecanismo: algo novo e crível."
visual_ref: "assets/img_001.svg"
checkpoint:
    type: "ESCOLHA"
    prompt: "Qual opção define melhor Big Idea?"
    options:
        - "Uma promessa genérica"
        - "Uma promessa com mecanismo crível"
        - "Um título bonito"
    correct: 2
    feedback_correct: "Boa. Você entendeu o núcleo."
    feedback_wrong: "Isso é promessa solta. Falta mecanismo."
- beat: 2
    narration: "Sem mecanismo, o cérebro trata como 'mais do mesmo'."
    visual_ref: "assets/img_002.svg"
    checkpoint:
        type: "RECALL"
        prompt: "Complete: Big Idea = Promessa + ____"
        answer: "mecanismo"
        feedback_correct: "Perfeito."
        feedback_wrong: "Era 'mecanismo'."
```

closing:

summary:

- "Big Idea não é slogan."
- "Elas une promessa + mecanismo crível."
- "O objetivo é quebrar o 'já ouvi isso'."

```
applied_action:  
    prompt: "Escreva 1 Big Idea do seu tema com promessa + mecanismo."
```

- Isso já nasce “rodável” e consistente com o motor.
-

5) Fluxo de importação (como vira app de verdade)

Etapa	O que acontece	Regra de governança
Upload	criador sobe o .zip	valida estrutura
Validação	checa campos obrigatórios	aula sem checkpoint = bloqueia
Preview	mostra como aluno verá	simula SAFE/TENSION/STATUS
QA Gate	checklist final	impede lixo em escala
Publish	publica versão v1	cria versão e changelog

O que eu ajustaria no que você pediu (pra ficar “completo e final”)

- A importação não pode ser “só subir e pronto” sem **QA gate**, senão o seu “80 apps” vira “80 apps ruins”.
 - O Studio precisa ter **Workspace/Roles/Versioning/Billing**, senão não é produto final.
-

Próximo passo (pra gente fechar de vez)

Eu te entrego a **tabela final de campos obrigatórios** (manifest, lesson, mission, application, srs) — tipo “dicionário oficial” — pra virar especificação do produto.

[resumo]

- **Aplicação** = arquétipo de tarefa de mundo real com **prova + rubrica + feedback**, para gerar **resultado** (não só “coisa rápida”).
- **Aba do Creator = Modo Studio** (Criar → Importar/Build → Preview → QA/Publish → Versões → Analytics → Recompensas → Configurações + Workspace/Roles/Billing).
- **Importação em massa (produto final): Pacote InfoApp (.zip)** com `manifest.yaml`, `tracks.yaml`, conteúdos em YAML por tipo (lessons/missions/applications), `srs/vocab.csv` e `assets/`.
- **Aula Interativa no pacote:** beats + checkpoint por beat; sem checkpoint, **bloqueia publicação** via QA gate.



Antes de avançar, me diga, esse produto é algo superficial? Coisa genérica ou através da importação pode ser completo e amplo, tudo dependendo da qualidade das importações e qualidade do tema?
Como podemos garantir que tenha qualidade e seja realmente uma edtech o resultado

Não é superficial por natureza — ele **pode virar superficial** se vocês tratarem como “gerador de microcards”.

Mas, do jeito que vocês estão desenhando (motor + formatos oficiais + checkpoints + aplicação + QA gate), ele pode ser **um sistema completo de EdTech**, porque o “tamanho” do aprendizado passa a ser **uma propriedade do pacote importado + governança**, não da plataforma.

A verdade nua:

- A plataforma é um **motor**.
 - A profundidade vem de **(1) formato certo, (2) blueprint certo, (3) governança de qualidade, (4) aplicação com prova**.
-

1) O que define se é “edtech de resultado” (tabela)

Se faltar...	Vira o quê	Se existir...	Vira o quê
Checkpoints	consumo passivo	checkpoints por beat	processamento ativo
Aplicação com prova	“aprendi na teoria”	prática no mundo real	transferência e resultado
Trilha progressiva	pílulas soltas	sequência guiada	competência construída
Rubrica/critério	subjetivo	critério objetivo	evolução mensurável
Analytics por beat	achismo	drop-off e gaps	melhoria contínua
QA gate	“80 apps ruins”	padrão mínimo	catálogo forte

2) Sim: pode ser completo e amplo via importação

A importação (pacote) pode conter:

- **Trilhas longas** (módulos, níveis, pré-requisitos)
- **Aulas interativas profundas** (8+ beats, cenários, simulações)
- **Revisão espaçada** (SRS para memória)
- **Aplicações e projetos** (tarefas reais com prova)
- **Assets ricos** (SVG/anim, narrativas, exemplos)

Ou seja: o produto suporta “curso inteiro”, só que **reformatado em experiência ativa**.

3) Como garantir qualidade (sem depender de “bom senso”)

Aqui entra o que um conselheiro experiente faria: **qualidade por design**, não por promessa.

3.1 “Quality Gates” automáticos (bloqueiam publicação)

Gate	Regra objetiva	Se falhar...
Gate 1 — Atividade	aula tem checkpoint em todo beat	não publica
Gate 2 — Transferência	cada módulo tem pelo menos 1 aplicação	não publica (ou publica em “beta”)
Gate 3 — Progressão	trilha tem pré-requisitos claros / ordem	não publica
Gate 4 — Feedback	todo checkpoint tem feedback (não só “certo/errado”)	não publica
Gate 5 — Tempo/escopo	beats 45–90s (tolerância)	alerta/ajuste automático

Gate	Regra objetiva	Se falhar...
Gate 6 — Clareza	objetivo da aula + “o que você saberá fazer”	não publica

3.2 Score de Qualidade (para ranquear apps e proteger a marca)

Dimensão	Como medir	Alvo
Atividade	% beats com checkpoint	100%
Transferência	apps com prova / módulo	≥1
Profundidade	nº de simulações/aplicações por trilha	mínimo por nível
Retenção	SRS presente quando faz sentido	sim/não
Clareza	objetivo + resumo + ação aplicada	100%
Engajamento real	completion rate + retorno D1/D7	thresholds

Apps abaixo de um score:

- ficam “não listados” no marketplace
- ou aparecem com selo “beta”
- ou exigem revisão manual

4) Garantia humana: revisão e “padrões editoriais”

Automação segura + padrão editorial é o que transforma isso em edtech de verdade.

Padrões editoriais (simples, mas fortes)

Padrão	Exigência
Learning outcomes	toda trilha define "o que você faz no final"
Exemplos reais	1 exemplo real por aula (mínimo)
Aplicação	sempre existir "ação aplicada"
Rubrica	prova com critérios objetivos
Linguagem	clareza > floreio, sem jargão desnecessário
Tension dial	SAFE/TENSION/STATUS coerente com estação

E aí sim entra um "review board" leve:

- revisão por amostra (5% das aulas)
- revisão obrigatória nos apps que vão pro marketplace

5) O que impede o risco "genérico"

O genérico nasce quando:

- o criador importa texto e o sistema só quebra em cards
- não há exercícios com consequência
- não há aplicação com prova

- não há rubrica

Antídoto: importação sempre gera "estrutura de competência", não "conteúdo fatiado".

6) Garantia final: o produto promete "resultado"

Se vocês querem "edtech de resultado", a plataforma precisa ter **uma noção explícita de competência**.

Competência (tabela)

Campo	Exemplo
Competência	"Escrever Big Idea crível"
Evidência	3 Big Ideas criadas + 1 aplicada em anúncio
Rubrica	mecanismo presente? promessa clara? prova?
Nível	iniciante/intermediário/avançado

Isso permite dizer:

"você não só consumiu — você **provou**."

[resumo]

- O produto **não é superficial por natureza**; ele se torna superficial se for apenas "microcards sem checkpoints e sem aplicação".

- Profundidade depende de: **qualidade do pacote importado + governança (quality gates) + aplicação com prova + rubrica + progressão.**
- Garantia de qualidade: **QA gates automáticos + score de qualidade + padrões editoriais + revisão humana por amostra/marketplace.**
- Para ser “edtech de resultado”, adicionar no sistema o conceito de **competência** (evidência + rubrica + nível).



Ok, então vamos seguir essas garantias., precisamos agora da definição de todas as telas, tem em todas as telas e como deve ser o visual. Da aba, e oq sera usado em cada tela. É importante contratar tamb um sound designer que vai usar a API do elevemlabs que permite colocar efeito sonoso e feedback auditivo, nenhum detalhe pode ficar de fora,

Pensou por 10s >

Fechado. Pra nenhum detalhe ficar de fora, eu vou tratar isso como **especificação de produto final** (UX/UI + Design System + Sound System + Screen Specs). E sim: **Sound Designer entra como membro fixo do conselho** porque áudio/haptics vira parte do “motor de estados”.

Conselho ampliado (UX/UI + áudio)

- Head Product Design (EdTech/Creator tools)
- UX Architect (IA + flows)

- Design Systems Lead (componentes/tokens)
 - Motion/Interaction Designer
 - UX Writer (microcopy/feedback)
 - Growth/Retention Designer (loops)
 - **Sound Designer (novo) + Implementação ElevenLabs/FX** (feedback auditivo)
 - Accessibility & Inclusive Design
-

0) Regra-mãe do visual

Não é “app bonito”. É “app claro e inevitável”.

O visual precisa reforçar:

- 1 CTA dominante por tela
 - progresso visível
 - feedback imediato
 - sem poluição / sem gimmick
-

1) Design System final (visual) — tokens e estilo

1.1 Direção de arte (tabela)

Item	Diretriz final	Por quê
Estética	premium, limpo, “tool-like” (não infantil)	escala pra qualquer nicho
Densidade	baixa/média, muito respiro	reduz carga cognitiva

Item	Diretriz final	Por quê
Hierarquia	título forte + 1 CTA	evita indecisão
Cards	cantos arredondados, sombra leve	leitura e toque
Ilustrações	SVG/anim sutil (não exagerado)	reforça conceito sem distrair
Modo	claro + escuro desde o início	produto final

1.2 Tipografia e layout (tabela)

Token	Padrão
Fonte	system (rápida) ou Inter (profissional)
Tamanhos	H1 24–28, H2 18–20, body 14–16
Grid	8pt grid, spacing consistente
Botões	primário sólido, secundário ghost
Ícones	set único (line icons), consistentes

1.3 Componentes obrigatórios do Design System (lista curta)

- **MissionCard** (CTA do dia)
- **ProgressHeader** (streak + XP do dia)
- **TrailMap** (etapas + bloqueios)
- **PlayerFrame** (áudio + visual + beat progress)

- **CheckpointModule** (Escolha/Quiz/Recall/Simulação)
 - **FeedbackCard** (Explain/Coaching/Reveal)
 - **XPBreakdownModal** (sempre no final)
 - **BadgeToast + BadgeGallery**
 - **SRSReviewCard**
 - **ProofUploader / ProofForm**
 - **QA Gate Banner** (bloqueia publish)
 - **Importer** (CSV/JSON/YAML + validação)
 - **Preview Emulator** (simula aluno)
 - **AppStatusPill** (draft/beta/published)
-

2) Sound System final (áudio + haptics) — obrigatório

Sound não é "enfeite": é **feedback de estado** (segurança, tensão, conquista).

2.1 Biblioteca de sons (tabela)

Cue	Quando toca	Estado psicológico	Observação
tap_soft	clique/ação normal	controle	curtinho, discreto
checkpoint_open	abre checkpoint	foco	"atenção agora"
correct_light	acerto comum	reforço	não infantil
wrong_soft	erro com SAFE	segurança	sem punição emocional

Cue	Quando toca	Estado psicológico	Observação
<code>wrong_tension</code>	erro com TENSION	alerta	mais seco, não agressivo
<code>streak_saved</code>	mantém streak	alívio	"ufa"
<code>streak_lost</code>	perde streak	perda	triste, mas leve
<code>level_up</code>	subiu de nível	orgulho	assinatura do produto
<code>badge_unlock</code>	novo badge	identidade	curto + marcante
<code>activity_complete</code>	concluiu tarefa	fechamento	prepara XP modal
<code>confetti_pop</code>	confetti	vitória	opcional (toggle)
<code>upload_success</code>	prova enviada	progresso real	"feito"
<code>publish_success</code>	app publicado	conquista do criador	assinatura forte

2.2 Regras de implementação (tabela)

Regra	Padrão
Volume	baixo por padrão + controle no perfil
Mute	toggle global + por contexto
Haptics	leve (tap), médio (correct), forte (level_up)
Acessibilidade	"reduce motion" + "reduce sound cues"

Regra	Padrão
Coerência	sons variam por tension_profile (SAFE/TENSION/STATUS)
ElevenLabs	usado para narrativa ; FX via biblioteca local/CDN

2.3 Papel do Sound Designer (brief de contratação)

Ele vai entregar:

- **Sound Kit v1** (biblioteca + naming + guidelines)
- **Mix padrão** (volumes relativos por cue)
- **Mapeamento por estado** (SAFE vs TENSION vs STATUS)
- **Assinatura sonora** (level_up / publish_success)
- **Testes em devices** (celular/com fone/sem fone)

3) Especificação de todas as telas (com "o que tem" + "visual" + "o que usa")

Para ficar completo e não virar texto infinito, eu vou usar um **template único** e preencher tudo.

Template de tela (padrão)

- **Objetivo**
- **Componentes (Design System)**
- **Ação primária**
- **Estados (loading/empty/error/success)**
- **Sons/Haptics**

- Eventos (analytics)
-

3.1 Learner App — todas as telas

Entrada & conta

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Login	entrar	inputs, primary button, SSO	"Entrar"	tap_soft	auth_start/success/fail
Signup	criar conta	inputs, termo, SSO	"Criar conta"	tap_soft	signup_success
Recuperar senha	acesso	input, CTA	"Enviar link"	tap_soft	reset_requested
Perfil	controle	settings list	—	tap_soft	settings_open
Configurações	preferências	toggles (sound/haptics/motion)	"Salvar"	tap_soft	settings_save

Onboarding

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Objetivo	intenção	cards choice	"Continuar"	tap_soft	onboarding_goal
Nível	calibrar	slider/choices	"Continuar"	tap_soft	onboarding_level

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Rotina	habito	time picker	"Definir rotina"	tap_soft	onboarding_routine
Tutorial streak	evitar perda	modal/coachmarks	"Entendi"	checkpoint_open	tutorial_done

Núcleo diário

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Home (Missão do Dia)	retorno diário	MissionCard, ProgressHeader	"Começar missão"	tap_soft	mission_start
Mapa/Trilha	caminho	TrailMap, AppStatusPill	"Ir para próxima"	tap_soft	trail_open
Progresso	onde estou	charts simples, stats	—	tap_soft	progress_open
Badges/Galeria	identidade	BadgeGallery	—	badge_unlock (quando novo)	badge_view

Player (unificado)

Tela	Objetivo	Componentes	Primário	Sons	Eventos
PlayerFrame (Aula/Story)	aprender ativo	PlayerFrame, Progress (beats)	"Continuar"	tap_soft	lesson_open, beat_view

Tela	Objetivo	Componentes	Primário	Sons	Eventos
CheckpointModule (overlay)	processamento ativo	CheckpointModule, FeedbackCard	"Responder"	checkpoint_open	checkpoint_answer
Resumo + ação aplicada	transferência	closing summary, applied action	"Marcar como feito"	upload_success (se prova)	action_submitted

Revisão e aplicação

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Review (SRS)	retenção	SRSReviewCard stack	"Responder próxima"	correct_light/wrong_soft	srs_answer
Aplicação (mundo real)	resultado	checklist + ProofForm	"Enviar prova"	upload_success	proof_submit
Histórico/Biblioteca	retomada	list + filters	"Continuar"	tap_soft	content_resume

Fechamento

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Conclusão (XP breakdown)	reforçar	XPBreakdownModal, BadgeToast	"Fechar"	activity_complete + (badge_unlock)	activity_complete
Streak em risco	salvar	modal + CTA	"Fazer missão"	streak_saved	streak_risk

Social (governado)

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Ligas/Ranking	status saudável	leaderboard (TEAM/PRIVATE)	"Ver liga"	tap_soft	leaderboard_view
Equipe/Grupo	contexto	group card	"Entrar"	tap_soft	group_open

3.2 Creator Studio — todas as telas

Workspace & Dashboard

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Login Studio	acesso	auth UI	"Entrar"	tap_soft	studio_auth
Workspace switch	multi-marca	selector	"Selecionar"	tap_soft	workspace_select
Dashboard	visão geral	cards + KPIs	"Criar InfoApp"	tap_soft	dashboard_view

Criar / Build

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Create InfoApp (Wizard)	iniciar	form + presets	"Criar"	tap_soft	infoapp_create
Escolher Formato	reduzir opções	format cards	"Continuar"	tap_soft	format_select

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Import/Build (Pacote .zip)	linha de montagem	Importer + validação	"Validar"	tap_soft	import_upload
Build Aula Interativa	gerar cards	beats preview + checkpoints	"Gerar"	tap_soft	lesson_generate
Build Story	narrativa	preview + circling	"Gerar"	tap_soft	story_generate
Build Review SRS	retenção	CSV validator	"Importar"	tap_soft	srs_import
Build Aplicação	tarefa real	checklist + rubrica	"Salvar"	tap_soft	apply_create

Preview / QA / Publish

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Preview Emulator	ver como aluno	emulator + persona	"Testar fluxo"	tap_soft	preview_run
QA Checklist	qualidade	QA Gate Banner + checklist	"Corrigir"	wrong_soft (se falha)	qa_fail/pass
Publish	lançar	version + notes	"Publicar"	publish_success	publish_done
Versioning	evoluir	versions list	"Criar nova versão"	tap_soft	version_create

Operação (final)

Tela	Objetivo	Componentes	Primário	Sons	Eventos
Analytics	melhorar	funnels + drop beats	—	tap_soft	analytics_view
Rewards/Economia	retenção	rules + badges	"Salvar regras"	tap_soft	economy_save
Users/Cohorts	suporte	cohorts table	—	tap_soft	cohort_view
Assets Library	reuso	asset manager	"Upload"	tap_soft	asset_upload
Settings/Integrations	TTS/branding	keys + toggles	"Salvar"	tap_soft	settings_save
Roles & Audit	controle	roles table	"Convidar"	tap_soft	role_update
Billing	comercial	plan + invoices	"Atualizar plano"	tap_soft	billing_update

4) "O que será usado em cada tela" (resumo por componentes)

Se você quiser uma visão ultra objetiva, aqui vai o "mapa componente → telas":

Componente	Telas
MissionCard	Home
PlayerFrame	Player Aula/Story
CheckpointModule	dentro do Player

Componente	Telas
XPBreakdownModal	Conclusão
TrailMap	Trilha
SRSReviewCard	Review
ProofForm/Uploader	Aplicação
QA Gate Banner	QA
Importer	Import/Build
Preview Emulator	Preview
BadgeToast/Gallery	Conclusão/Badges
ProgressHeader	Home/Progresso

Próximo passo (pra avançar de verdade)

Escolhe por onde você quer que eu detalhe tela por tela com layout textual (wireframe em bullets + estados + microcopy + sons):

1. **Learner primeiro** (Home → Player → Checkpoints → Conclusão → Review → Aplicação)
2. **Creator Studio primeiro** (Create → Import Pack → Preview → QA → Publish → Analytics)

Eu recomendo começar por **Learner Home + Player**, porque isso fixa o “idioma visual” do produto.

Ok, antes de fecharmos então as definições, alguém do conselho quer adicionar ou tirar algo?

Sim — o conselho inteiro tem ajustes finais. Vou colocar **só o que muda de verdade** (adicionar/remover/ajustar), em tabelas, sem repetição.

1) Head of Product Design — ajustes de produto (tabela)

Proposta	Adicionar / Tirar	Motivo	Impacto
Marketplace/Store público	Adicionar	produto final precisa de descoberta/distribuição	cria "catálogo" e SEO
Modo "Demo/Preview público"	Adicionar	creator vende/compartilha antes do usuário logar	conversão e viralização
Home do aluno: 1 CTA apenas	Reforçar	evita indecisão	melhora completion

2) UX Architect — ajustes de navegação e IA (tabela)

Proposta	Adicionar / Tirar	Motivo	Detalhe
Search global (aluno e creator)	Adicionar	produto completo precisa achar conteúdo rápido	search + filtros
Histórico “Continuar” sempre no topo	Adicionar	reduz fricção	botão persistente
Checkpoint como tela	Tirar	é componente do Player	mantém consistência

3) Design Systems Lead — ajustes de consistência visual (tabela)

Proposta	Adicionar / Tirar	Motivo
Tokens por “tension_profile” (SAFE/TENSION/STATUS)	Adicionar	muda o clima sem redesign
Estados padrão (empty/error/loading) com componentes prontos	Adicionar	escala e velocidade de dev
Excesso de animação	Tirar/limitar	foco > espetáculo

4) Motion/Interaction Designer — ajustes de microinterações (tabela)

Proposta	Adicionar / Tirar	Motivo	Nota
Transição entre beats (snap + progress)	Adicionar	sensação de avanço real	suave, curto
Confetti sempre	Tirar (tornar condicional)	enjoa e infantiliza	só level_up/badge
Micro-feedback no checkpoint (shake leve, glow)	Adicionar	reforço sem texto extra	respeitar "reduce motion"

5) UX Writer — ajustes de linguagem e feedback (tabela)

Proposta	Adicionar / Tirar	Motivo
Feedback "o que isso revela" padrão	Adicionar	identidade > certo/errado
Tom infantil (parabéns!!!)	Tirar	produto multi-nicho e premium
Texto curto + claro por beat	Reforçar	áudio narra; texto guia

6) Growth/Retention Designer — ajustes de retenção (tabela)

Proposta	Adicionar / Tirar	Motivo
Notificações por contexto (missão, streak risk, unlock)	Adicionar	v1 completo precisa retenção
Calendário/agenda semanal (planejamento)	Adicionar	sensação de controle e compromisso
Ranking público	Tirar	risco de vergonha/toxicidade

7) Accessibility — ajustes finais obrigatórios (tabela)

Proposta	Adicionar / Tirar	Motivo
Controles: som / haptics / motion	Adicionar	acessibilidade real
Legendas no player	Adicionar	inclusivo + ambientes sem som
Contraste mínimo	Reforçar	WCAG e leitura

8) Sound Designer — ajustes no sistema de áudio (tabela)

Proposta	Adicionar / Tirar	Motivo	Observação
Som por “estado” (SAFE/TENSION/STATUS)	Adicionar	áudio vira parte do motor	mesma ação, cue diferente

Proposta	Adicionar / Tirar	Motivo	Observação
Sound toggle rápido no Player	Adicionar	usuário controla no momento	sem sair da aula
Erros com som agressivo	Tirar	punição emocional mata aprendizado	erro = coaching

9) Trust & Safety / Anti-abuse — ajustes de plataforma final (tabela)

Proposta	Adicionar / Tirar	Motivo
Moderação de conteúdo + reports	Adicionar	marketplace + creators exigem
Anti-spam e anti-fraude em prova	Adicionar	aplicações com upload pedem proteção
Auditoria de ações no Studio	Adicionar	escala com times/roles

Lista final do que o conselho quer MUDAR (resumo curto)

Adicionar (essenciais v1 completo):

- Marketplace/Store + “demo/preview público”
- Search global
- Calendário/agenda semanal + notificações contextuais

- Tokens visuais por tension_profile
- Acessibilidade (som/haptics/motion + legendas)
- Trust: moderação + anti-fraude + auditoria

Tirar/limitar:

- confetti sempre
 - ranking público
 - erro com som agressivo
 - checkpoint como "tela" separada (fica dentro do player)
-

[resumo]

- Conselho adiciona itens de produto final: **Marketplace/Store, demo/preview público, search global, agenda semanal, notificações contextuais, tokens por tension_profile, acessibilidade completa, trust/moderação/anti-fraude/auditoria.**
- Conselho remove/limita: **confetti sempre, ranking público, sons punitivos, checkpoint como tela separada.**

Se você disser "aprovado", eu consolido tudo em um **Documento Mestre de UX/UI** (tabelas finais por tela + componentes + tokens + sons + eventos) pronto pra repassar pra design/dev/sound.



Perfeito, agora precisamos fechar esse projeto, qual o melhor caminhão, quais documentos precisamos criar para fechar tudo?

O melhor “caminho” é fechar em **3 camadas de documentação**: (1) *produto e regras*, (2) *experiência e design*, (3) *engenharia e operação*. PRD/SRS ajudam, mas sozinho não fecha esse tipo de plataforma (creator + marketplace + importação + QA + áudio).

Abaixo vai um pacote de documentos **completo e final**, em **ordem de criação**, com o que cada um precisa conter.

1) Produto e Governança (fecha o “o que” e o “por quê”)

1. Vision / One-Pager

- tese, público (criador), promessa, diferenciais, o que “não é”.

2. PRD (Product Requirements Document)

- escopo completo (Learner/Studio/Admin/Marketplace), formatos oficiais, regras do motor (Tema→Estação→Dinâmica→Tarefa), “tension dial”, aplicação com prova, qualidade.

3. Governance & Quality Gates Spec

- gates automáticos (ex: aula sem checkpoint bloqueia publish), score de qualidade, regras do marketplace (beta/unlisted/listed), revisão humana por amostra.

4. Content Standards (Guia Editorial)

- outcomes por trilha, exemplo mínimo por aula, rubrica de aplicação, tom de voz, anti-jargão, padrões de feedback “o que isso revela”.

5. Monetization & Plans Spec

- billing, planos, limites (apps/MAU/uso de TTS), marketplace (revshare), trials, roles.
-

2) UX/UI e Experiência (fecha “como parece e funciona”)

6. IA + User Flows (Arquitetura de Informação)

- mapa final de telas + navegação oficial + flows principais (aluno e criador).

7. UX Spec por tela (Wireframe textual + estados)

- para cada tela: componentes, ação primária, empty/error/loading, regras do motor, eventos.

8. Design System Spec

- tokens (incluindo tokens por SAFE/TENSION/STATUS), componentes obrigatórios, estados padrão, acessibilidade.

9. Sound & Haptics Spec (com Sound Designer)

- biblioteca de cues, regras por tension profile, mix/volumes, toggles, “reduce sound/motion”, assinatura sonora (level_up/publish).

10. Accessibility & Localization Spec

- WCAG, legendas no player, contraste, idiomas, formatos regionais.
-

3) Engenharia e Operação (fecha “como implementa e escala”)

11. SRS (Software Requirements Spec)

- requisitos funcionais e não funcionais: performance, offline (se houver), segurança, privacidade, logging, auditoria.

12. ERS (Engineering Requirements / Architecture Spec)

- arquitetura de alto nível, serviços, filas, armazenamento de assets, versão, rollback, observabilidade, custo.

13. Data Model (ERD) + Schema Dictionary

- entidades base (InfoApp/Tema/Estação/Dinâmica/Tarefa/Recompensa etc.), campos obrigatórios (tension, station...), versionamento e audit.

14. Import Pack Spec (Pacote .zip)

- manifest.yaml , tracks.yaml , lessons/missions/applications , srs/vocab.csv , assets/ ; validações; mensagens de erro; lint.

15. API Spec (OpenAPI)

- endpoints do Learner, Studio, Admin; auth; upload; publish; analytics; roles.

16. Event Taxonomy & Analytics Plan

- eventos: beat_view, checkpoint_answer, drop_off_beat, proof_submit, qa_fail/pass, publish_done; dashboards e métricas-alvo.

17. Trust/Safety & Moderation Spec

- reports, abuso, fraude de prova, conteúdo sensível, auditoria, bloqueios, revisão.

18. QA/Test Plan + Release Runbook

- testes por formato, testes de importação em massa, critérios de aceite, playbook de incidentes.

19. Ops: SLO/SLA + Monitoring

- uptime, latência, filas, falhas do TTS, quotas, alertas.

Ordem recomendada pra “fechar” sem voltar atrás

1. Vision/One-Pager
2. PRD
3. IA + User Flows
4. UX Spec por tela
5. Design System + Sound Spec
6. Governance & Quality Gates
7. Import Pack Spec
8. ERD/Schema + API Spec

9. Event Taxonomy + Moderation + QA/Runbook

10. SRS + ERS + SLO/SLA

O que tem além de PRD/SRS/ERS (o “pulo do gato”)

Se você só fizer PRD/SRS/ERS, fica faltando o que protege qualidade e escala:

- Governance & Quality Gates
- Import Pack Spec
- Sound & Haptics Spec
- Event Taxonomy
- Moderation/Trust

Esses 5 são os que transformam “builder genérico” em **edtech de resultado**.

Se você quiser, eu começo imediatamente pelo **documento #2 (PRD)** já no formato final (tabelas, seções e critérios), porque ele vira a âncora do resto.



ok, conselho e convidados temos tudo?

Temos o essencial, mas pra “fechar tudo” (produto final + marketplace + importação em massa + áudio + moderação + escala) eu faria **2 ajustes no conselho: adicionar 3 perfis fixos e 1 convidado sob demanda**.

Conselho fixo (completo)

Área	Pessoa
Produto/Visão	Head of Product (ou PM Principal)
UX/IA	UX Architect
UI/Design System	Design Systems Lead
Interação/Motion	Motion/Interaction Designer
Conteúdo/UX Writing	UX Writer / Content Designer
Retenção/Economia	Growth & Retention Designer + Game Economy
Psicologia/Aprendizagem	Learning Scientist (processamento ativo)
Creator Tools	Specialist em Creator Studio
Engenharia/Arquitetura	Staff Engineer / Architect
Dados/Analytics	Data/Tracking Lead
Trust/Safety	Trust & Safety Lead
Áudio	Sound Designer (fixo)
Acessibilidade	Accessibility Lead
Operações	PMO/Program Manager

Convidados que eu adicionaria (fixos)

1. Head of Content Ops / Editorial Lead

- porque “qualidade” vai morrer sem padrão editorial + revisão por amostra + guidelines para criadores.

2. Legal/Compliance (privacidade + marketplace + conteúdo + UGC)

- produto final com upload, marketplace e pagamentos precisa de regras, termos e moderação bem desenhados.

3. FinOps/Infra Cost

- ElevenLabs + assets + vídeos + geração + storage podem explodir custo em escala (80 apps + usuários). Precisa governar quota, caching e limites por plano.

Convidado sob demanda (quando for implementar a importação)

4. Developer Experience (DX) / Tooling

- pra garantir que o “Import Pack” seja realmente fácil, validado e com mensagens de erro ótimas (senão ninguém usa em volume).

Veredito do conselho

Sim, agora está completo para fechar PRD/SRS/ERS + Design System + Sound Spec + Import Spec + Moderation + Billing.

[resumo]

- Conselho “completo final” adiciona 3 perfis fixos: Editorial/Content Ops, Legal/Compliance, FinOps/Infra Cost; e 1 convidado sob demanda: DX/Tooling (para import pack e validação).



ok, então conselho e convidados, deem sua mensagem final antes de tudo ser enviado para o desenvolvimento

Abaixo vai a “mensagem final” do conselho e convidados — **cada um com o que aprova, o que exige como não-negociável, e o risco principal** antes de ir pra desenvolvimento.

1) Head of Product (Produto)

Aprovo: motor (Tema→Estação→Dinâmica→Tarefa), 3 arquétipos (Missão/Aula Interativa/Aplicação), Creator Loop via Import Pack.

Não-negociável: Quality Gates bloqueando publish (checkpoint por beat; aplicação por módulo; feedback obrigatório).

Risco principal: virar “builder genérico” se **governança** ficar opcional.

2) UX Architect (IA/Flows)

Aprovo: separação Learner vs Creator Studio vs Platform Admin, Player unificado com modos.

Não-negociável: navegação oficial (tabs aluno + sidebar studio) + “Continuar” persistente + search global.

Risco: excesso de caminhos; **1 CTA dominante** por tela é lei.

3) Head Product Design (UX/UI)

Aprovo: direção premium, multi-nicho, tokens por tension_profile.

Não-negociável: Design System com componentes fechados antes de “pixel polish” solto.

Risco: inconsistência visual entre apps (Learner/Studio/Admin).

4) Design Systems Lead (UI Scale)

Aprovo: lista de componentes obrigatórios + estados padrão.

Não-negociável: tokens (spacing/type/color) + variantes SAFE/TENSION/STATUS + dark mode + acessibilidade.

Risco: “cada tela inventa um padrão” → explode manutenção.

5) Motion/Interaction Designer

Aprovo: beats + checkpoints como ritmo do produto.

Não-negociável: motion util, condicional (reduce motion), confetti só em marcos.

Risco: animação demais virar distração/infantilização.

6) UX Writer / Content Designer

Aprovo: feedback “o que isso revela” e linguagem premium.

Não-negociável: microcopy padrão para erros, checkpoints, QA gates e publish.

Risco: se o texto ficar genérico, a experiência vira “curso fatiado”.

7) Learning Scientist (Aprendizagem)

Aprovo: checkpoint como processamento ativo + aplicação com prova.

Não-negociável: aula sem checkpoint = **não publica**; cada trilha precisa outcome claro.

Risco: se tolerarem consumo passivo, perde o diferencial científico.

8) Growth & Retention Designer (Loops)

Aprovo: streak/badges/recompensas com governança, agenda semanal, notificações contextuais.

Não-negociável: ranking público fora; ligas só TEAM/PRIVATE; "streak em risco" bem feito.

Risco: economia virar grind/cassino se multiplicadores forem complexos.

9) Game Economy / LiveOps

Aprovo: XP base por arquétipo + bônus simples.

Não-negociável: economia mínima e previsível; sem bônus agressivo de velocidade.

Risco: incentivo errado gera comportamento errado (pressa, trapaça, churn).

10) Sound Designer (Áudio)

Aprovo: sons como parte do motor de estado.

Não-negociável: sound kit v1 + mix + cues por tension_profile + toggle no player.

Risco: som punitivo ou alto mata aprendizagem; acessibilidade é mandatória.

11) Accessibility Lead

Aprovo: controles de som/haptics/motion e legendas.

Não-negociável: contraste, foco de teclado, legibilidade, reduce motion/sound.

Risco: “corrigir depois” vira dívida impossível; fazer certo agora.

12) Trust & Safety / Moderation

Aprovo: reports, auditoria e anti-fraude de prova.

Não-negociável: moderação para marketplace + políticas de conteúdo + logs.

Risco: sem trust, marketplace vira risco reputacional e legal.

13) Staff Engineer / Architect

Aprovo: Import Pack (.zip) + versioning + QA gates + preview emulator.

Não-negociável: ERD/Schema + OpenAPI + event taxonomy antes do build grande.

Risco: começar “na pressa” sem contratos = rework caro.

14) Data/Analytics Lead

Aprovo: analytics por beat + drop-off + cohorts.

Não-negociável: taxonomia de eventos e dashboards-alvo desde o dia 1.

Risco: sem telemetria, vocês não conseguem provar “edtech de resultado”.

15) Content Ops / Editorial Lead

Aprovo: padrões editoriais e revisão por amostra.

Não-negociável: guide editorial + rubricas + QA editorial para apps listados.

Risco: escala sem editorial = catálogo fraco e perda de confiança.

16) Legal/Compliance

Aprovo: separação de papéis, auditoria, políticas de conteúdo.

Não-negociável: termos (UGC, uploads, marketplace), privacidade (LGPD), consentimentos.

Risco: upload/prova + marketplace sem base legal = bomba.

17) FinOps/Infra Cost

Aprovo: quotas por plano + caching e limites de geração.

Não-negociável: política de custo (TTS, storage, assets) + alertas.

Risco: custo explode silenciosamente com ElevenLabs e assets.

18) PMO/Program Manager

Aprovo: pacote de documentos e ordem de entrega.

Não-negociável: milestones com "Definition of Done" por documento (PRD/IA/DesignSystem/ImportSpec/ERD/OpenAPI).

Risco: construir antes de alinhar docs = atrasos e divergência.

Mensagem final unificada (para dev)

Construam isso como um motor governado, não como um editor solto.

O diferencial é inegociável:

1. Aula Interativa = beats + checkpoint obrigatório
2. Aplicação = prova + rubrica + feedback
3. Creator Loop = Import Pack em massa + QA gates bloqueando publish
4. Economia simples + áudio por estado + acessibilidade + trust/moderação

Se esses quatro estiverem no código como "leis", vocês terão uma edtech real e escalável.

