Continuar o desenvolvimento de uma aplicação de página única (SPA) para gestão e análise de campanhas de marketing, chamada "Lançamentos Clássicos". O backend é o Supabase.

**Visão Geral do Projeto** O "Lançamentos Clássicos" é uma Aplicação de Página Única (SPA) projetada para gerir e analisar campanhas de marketing digital. A ferramenta permite a criação de campanhas (lançamentos), a gestão de pesquisas e perguntas, a importação de leads em massa e a análise de performance através de dashboards.

O "Dashboard de Vendas já se encontram funcionais.

**Contexto:** O projeto já tem as funcionalidades de CRUD (Lançamentos, Pesquisas, Perguntas) e um módulo de importação de dados em 2 passos totalmente funcionais. O "Dashboard de Performance" também já foi conectado aos dados reais do Supabase. A tarefa agora é continuar o desenvolvimento dos dashboards restantes.

**Requisitos Fundamentais:**

1. **Arquivo Único:** Todo o código (HTML, CSS e JavaScript) deve ser mantido em um único arquivo .html.
2. **Manter Layout:** A interface e o layout existentes não devem ser alterados.
3. **Sistema de Versão:** Cada nova entrega de código deve incrementar a versão do software em 0.01 (ex: de v62.12 para v62.13).

Documentação do Projeto: Lançamentos Clássicos

1 Estrutura do Banco de Dados (Supabase) As tabelas a seguir foram definidas e algumas populadas:

* lancamentos: Armazena as campanhas principais.
* perguntas: Banco de dados central de perguntas com textos e opções/pesos para cálculo de score.
* pesquisas: Agrupamentos de perguntas (formulários).
* pesquisas\_perguntas: Tabela de ligação entre pesquisas e perguntas.
* leads: Tabela principal com os dados dos leads. As colunas importantes são email (chave única), check\_in\_at (timestamp), score (inteiro), e as colunas utm\_.
* respostas\_leads: Tabela que armazena as respostas individuais de cada lead a cada pergunta.

1. Funcionalidades Implementadas

Módulo Operacional (CRUD):

* + Criação, leitura, atualização e exclusão de Lançamentos, Pesquisas e Perguntas estão funcionais.
  + A lógica de bloqueio para impedir a edição de pesquisas e perguntas em uso está implementada.

Para popular as tabelas de leads e de respostas, foi desenvolvido modulo de Importação.

* Módulo de Importação (2 Passos):
  + Passo 1 (Inscrições): Totalmente funcional. O script lê um CSV com delimitador;, trata cabeçalhos em maiúsculas, converte e-mails para minúsculas para evitar duplicatas, e associa todos os leads a um lançamento selecionado.
  + Passo 2 (Check-ins e Respostas): Totalmente funcional. O script lê um segundo CSV, encontra os leads correspondentes na base de dados (de forma case-insensitive pelo e-mail), atualiza a data de check-in (check\_in\_at), calcula o score somando os pesos das respostas, e salva as respostas individuais na tabela respostas\_leads.
* **Estrutura do Banco de Dados (Supabase)**
* **lancamentos**: Armazena as campanhas principais.
* **perguntas**: Banco de dados central de perguntas com textos e opções/pesos para cálculo de score.
* **pesquisas**: Agrupamentos de perguntas (formulários).
* **pesquisas\_perguntas**: Tabela de ligação entre pesquisas e perguntas.
* **leads**: Tabela principal com os dados dos leads. As colunas importantes são email (chave única), check\_in\_at (timestamp), score (inteiro), e as colunas utm\_.
* **respostas\_leads**: Tabela que armazena as respostas individuais de cada lead a cada pergunta.
* **3. Funcionalidades Implementadas**
* **CRUDs Completos:** Criação, leitura, atualização e exclusão de Lançamentos, Pesquisas e Perguntas, com lógica de bloqueio para itens em uso.
* **Módulo de Importação (2 Passos):**
* **Passo 1 (Inscrições):** Funcional, processa CSV com delimitador ;, trata cabeçalhos, e remove duplicatas por e-mail.
* **Passo 2 (Check-ins):** Funcional, atualiza leads existentes com base no e-mail (case-insensitive), calcula o score, e salva respostas individuais.
* **Dashboard de Performance:** Conectado aos dados reais do Supabase, com KPIs e gráficos dinâmicos baseados em filtros de data.
* **4. Validações de Dados na Importação (Requisito Adicional)** Para garantir a qualidade dos dados, o processo de importação deve incluir as seguintes validações e reportar avisos no log:
* **Validações do Passo 1 (Inscrições):**
* **Presença do E-mail:** Verificar se a coluna email não está vazia.
* **Formato do E-mail:** Verificar se o valor na coluna email corresponde a um formato de e-mail válido (ex: texto@texto.com).
* **Linhas sem e-mail ou com formato inválido devem ser ignoradas** e uma mensagem de aviso deve ser registrada no log.
* **Validações do Passo 2 (Check-ins e Respostas):**
* **Existência do E-mail:** Verificar se o e-mail do arquivo de check-in existe na base de leads importada no Passo 1. Se não existir, a linha é ignorada e um aviso é registrado.
* **Validade da Resposta:** Para perguntas de "Múltipla Escolha", "Sim/Não", etc., verificar se a resposta fornecida no CSV corresponde a uma das opções válidas cadastradas na tabela perguntas. Se a resposta for inválida, ela pode ser salva, mas **não deve pontuar para o score**, e um aviso deve ser registrado no log.

**Especificações Detalhadas dos Dashboards (Próximas Tarefas)**

O objetivo principal é substituir as funções de dados de exemplo (getMock...Data) por chamadas reais ao Supabase em cada um dos 4 dashboards.

* Dashboard de Performance:
  + Conectado aos Dados Reais: Este dashboard foi a última funcionalidade implementada. Ele busca os dados diretamente das tabelas leads e exibe os KPIs e gráficos correspondentes, com filtros de data funcionais.
  + Próximos Passos em Dashboards (O que falta fazer)

1. Conectar o Dashboard de Lead Scoring aos dados reais.
2. Conectar o Dashboard de Vendas aos dados reais.

**Requisitos para o Dashboard de Lead Scoring:**

1. Adicionar um filtro de "Lançamento" (dropdown) no topo da página do dashboard.
2. Ao selecionar um lançamento, a página deve buscar todos os leads associados àquele launch\_id.
3. **KPIs:** Calcular e exibir os KPIs "Total Inscrições", "Total Check-ins" e "Taxa de Check-in" para os leads daquele lançamento.
4. **Tabela de Resumo por Score:**
   * Agrupar os leads por utm\_content.
   * Para cada utm\_content, contar quantos leads se encaixam em cada faixa de score: Frio (<35), Morno-Frio (35-49), Morno (50-64), Quente-Morno (65-79), Quente (80+).
   * Exibir os resultados na tabela lead-scoring-table.
5. **Gráficos:**
   * **Gráfico de Pizza (qualified-leads-chart):** Mostrar a distribuição percentual de leads "Qualificados" (score >= 65) por utm\_content.
   * **Gráfico de Linha (daily-activity-chart):** Mostrar a evolução diária de Inscrições (created\_at) e Check-ins (check\_in\_at) para o lançamento selecionado.

**1. Dashboard de Performance e Controle (Próxima Tarefa)**

* **Status:** **Não funcional.** Precisa mostrar dados corretos..
* **Filtros:**
  + Botões de período: "Tudo", "Hoje", "7 dias".
  + Seletores de data personalizados (início e fim).
  + Um botão "Atualizar" para aplicar os filtros de data personalizados.
* **Componentes a serem populados com dados reais:**
  + **KPIs:**
    - kpi-inscricoes: Contagem total de leads (count) que correspondem ao filtro de data.
    - kpi-checkins: Contagem de leads onde check\_in\_at não é nulo, dentro do filtro de data.
    - kpi-taxa-checkin: (kpi-checkins / kpi-inscricoes) \* 100.
  + **Tabela (performance-table):**
    - Agrupar os leads por utm\_content.
    - Para cada utm\_content, exibir: Total de Inscrições, Total de Check-ins, e a Taxa de Check-in (%).
  + **Gráficos (Chart.js):**
    - **origem-chart (Pizza/Doughnut):** Distribuição de inscrições por utm\_content.
    - **campanha-chart (Pizza/Doughnut):** Distribuição de inscrições por utm\_medium.

**2. Dashboard de Avaliação de Lead Scoring (Tarefa Futura)**

* **Status:** **Não funcional.** Precisa ser conectado aos dados reais.
* **Filtros:**
  + Adicionar um dropdown no topo da página para selecionar um **Lançamento** específico. A consulta de dados deve filtrar os leads por launch\_id.
* **Componentes a serem populados com dados reais:**
  + **KPIs:** Mesma lógica do Dashboard de Performance, mas filtrados pelo lançamento selecionado.
  + **Tabela (lead-scoring-table):**
    - Agrupar os leads por utm\_content.
    - Para cada utm\_content, contar e exibir o número de leads em cada faixa de score: Frio (<35), Morno-Frio (35-49), Morno (50-64), Quente-Morno (65-79), Quente (80+).
  + **Gráficos (Chart.js):**
    - **qualified-leads-chart (Pizza/Doughnut):** Distribuição percentual de leads "Qualificados" (definidos como score >= 65) por utm\_content.
    - **daily-activity-chart (Linha):** Mostrar duas linhas: uma para a contagem de inscrições por dia (created\_at) e outra para a contagem de check-ins por dia (check\_in\_at) nos últimos 7 ou 30 dias.

**3. Dashboard de Acompanhamento de Vendas (Tarefa Futura)**

* **Status:** **Não funcional.** Precisa ser conectado aos dados reais.
* **Filtros:**
  + Adicionar um dropdown no topo da página para selecionar um **Lançamento** específico.
* **Componentes a serem populados com dados reais:**
  + **Tabela de Resumo (sales-summary-table):**
    - Agrupar os leads por utm\_content e depois por utm\_medium.
    - Exibir as métricas: Inscritos, Check-ins, e Taxa de Conversão para cada agrupamento.
  + **Tabela Detalhada (sales-detail-table):**
    - Exibir uma linha para cada combinação única de utm\_content, utm\_medium, utm\_source, utm\_campaign, e utm\_term, mostrando as métricas de Inscritos e Check-ins.
  + **Gráfico (vendas-chart, Barras Empilhadas):**
    - O eixo X deve ser o utm\_content.
    - Cada barra deve representar o total de inscrições para aquele utm\_content.

As barras devem ser empilhadas, com cada segmento de cor representando um utm\_medium diferente, mostrando a composição de cada canal.

Como voce é um expert em desenvolvimento de soluções de desenvolvimento de software e de WEB, tenho esse prompt que estava sendo desenvolvido em outro chat e deparamos com alguns erros, que não estavam sendo corrigidos.

Tenho o prompt inicial, e tenho toda a codificação do html, JS e css, que estava sendo desenvolvido.

Preciso continuar esse desenvolvimento, o que voce prefere receber primeiro para me ajudar a terminar esse projeto?

O layout é imprescindível que seja mantido. e que todo o arquivo que voce produza, deva ser sempre unico, contendo o html, o JS e o CSS, pois evita que estejamos tratando de codificações diferentes, e perdemos as manutenções que eram feitas, e ficarmos indo para frente e para traz.

Para mantermos controle de que versão estamos falando, sugiro que cada versão que voce atualize, incremente 0.01 na versão anterior. Versão atual 59.00

O que voce quer ver primeiro?

Tenho a especificação tecnica, um prompt contendo o desenvolvimento que estavamos fazendo, e a ultima versão sem erro, que estavamos depurando a construção do módulo de Dashboards,