

# Documento de Briefing para Automação via GitHub

Para: Inteligência Artificial do GitHub (Copilot/Actions)

De: Cléber (via Gemini)

Assunto: Descrição do fluxo de trabalho completo para a geração automatizada de propostas e contratos da consultoria Live, para análise de viabilidade de automação.

## 1. Visão Geral do Processo

O objetivo é automatizar a criação de um pacote de entrega para novos clientes, que consiste em uma proposta comercial (página web) e um contrato de prestação de serviços. O sistema foi desenhado como um pipeline de múltiplas etapas, que combina entrada de dados humana com processamento por modelos de IA especializados.

## 2. Arquitetura do Pipeline (Etapas)

O fluxo de trabalho é sequencial e dividido em quatro etapas principais:

### Etapa 1: Entrada de Dados (Humano)

- **Ação:** Um consultor da Live preenche uma planilha de diagnóstico padronizada. Esta planilha é a fonte primária da verdade e contém dados quantitativos e qualitativos sobre diversos setores da empresa cliente (Estratégia, Financeiro, Marketing, etc.).

### Etapa 2: Análise de Diagnóstico (IA - "Manus")

- **Ação:** Os dados da planilha são fornecidos a um modelo de IA especializado, chamado "Manus".
- **Inteligência:** O Manus opera sob um prompt que lhe atribui a persona de um "consultor empresarial especialista". Sua função é analisar os dados brutos e sintetizá-los em uma apresentação de diagnóstico estruturada.
- **Referência do Processo:** O método de interação e o resultado esperado podem ser visualizados na seguinte conversa de referência:  
<https://manus.im/share/Z5oCgiMOFwhjFKHZSoRdd0>
- **Saída:** O Manus gera um texto organizado, separando os "Pontos de Melhoria" e as "Ações Propostas" para cada setor analisado. **Esta saída é a matéria-prima para a próxima etapa.**

### Etapa 3: Geração da Proposta e Contrato (IA - Gemini/Prompt Final)

- **Ação:** A "Apresentação de Diagnóstico" gerada pelo Manus é usada como entrada para um segundo processo de IA.
- **Inteligência:** Este processo utiliza um prompt-mestre que o instrui a usar as informações da apresentação para preencher dois templates simultaneamente:

1. Um **template de contrato jurídico** (live-contract-template-final), substituindo os placeholders ({{NOME\_CLIENTE}}, {{VALOR\_TOTAL\_NUMERICO}}, etc.) pelos dados correspondentes.
  2. Um **template de dados para a página web** (o objeto JavaScript dadosProposta), que estrutura todas as informações (diagnóstico, plano de ação, escopo, investimento) de forma que possam ser renderizadas dinamicamente.
- **Saída:** O resultado desta etapa é um único bloco de código (um objeto JavaScript/JSON) que contém tanto o texto completo do contrato preenchido quanto os dados estruturados para a proposta.

#### **Etapa 4: Publicação (Humano/Alvo da Automação)**

- **Ação Atual:** Atualmente, o consultor copia o objeto JavaScript gerado na Etapa 3 e o cola manualmente em um arquivo HTML padrão (index.html).
- **Estrutura do index.html:** Este arquivo é autossuficiente. Ele contém o layout, o CSS e um script que lê o objeto dadosProposta e renderiza dinamicamente todo o conteúdo da página.
- **Publicação:** O arquivo index.html é então salvo em um repositório do GitHub e publicado como uma página web via GitHub Pages.

#### **3. Objetivo da Automação (Consulta à IA do GitHub)**

A questão central é: **É possível automatizar a ponte entre a Etapa 3 e a Etapa 4 usando as ferramentas do GitHub (como GitHub Actions)?**

O fluxo ideal seria:

1. O consultor executa o processo com o Manus.
2. O Manus (ou um script intermediário) é autorizado a fazer um "push" para um repositório específico no GitHub.
3. A ação de "push" conteria o objeto de dados final (proposta + contrato).
4. Uma **GitHub Action** seria acionada por este push, pegaria os dados recebidos, os inseriria no template index.html, e faria o commit da nova versão da página, atualizando a proposta publicada no GitHub Pages.

#### **4. Artefatos de Referência**

- **Template do Contrato:** O documento live-contract-template-final é o modelo mestre para todos os contratos.
- **Template da Página Web:** O código-fonte da página index.html, que aguarda o objeto dadosProposta.

Este documento resume a arquitetura completa e o objetivo desejado. A análise da viabilidade e dos passos técnicos para criar essa automação no ecossistema do GitHub é o próximo passo desejado.

---

Elaboração do diagnóstico a partir de um áudio ou documento com as informações levantadas pelo consultor

<https://manus.im/share/Z5oCgjMOFwhjFKHZSoRdd0?replay=1>

## [PLANNER MODULE]

1. Analisar a planilha modelo localizada no link:

[https://docs.google.com/spreadsheets/d/1dzYmJ--YZ6tzUCbHDo3Dms4K\\_LfTaUY7\\_Ro0hoeRfjc/edit?gid=162999720#gid=162999720](https://docs.google.com/spreadsheets/d/1dzYmJ--YZ6tzUCbHDo3Dms4K_LfTaUY7_Ro0hoeRfjc/edit?gid=162999720#gid=162999720)

2. Identificar e mapear todas as abas existentes, com foco na estrutura de dados, fórmulas e lógica de preenchimento.

3. Compreender a relação entre os dados coletados, os cálculos automáticos e os campos que exigem preenchimento manual.

4. Criar um modelo de diagnóstico automatizado baseado nas fórmulas e estrutura da planilha.

5. A cada novo cliente, coletar os dados necessários e preencher automaticamente todas as abas e colunas da planilha.

6. Realizar análise interpretativa dos dados, especialmente nos campos que exigem elaboração consultiva, como a aba "Plano de Ação":

- Preencher o campo \*\*\*"Como"\*\*\* com recomendações estratégicas baseadas nos diagnósticos identificados.

- Definir o campo \*\*\*"Urgência"\*\*\* com base na gravidade, impacto e necessidade imediata de cada ação proposta.

7. Gerar um plano de ação estruturado, pronto para ser incluído tanto na apresentação quanto na proposta.

8. Gerar uma apresentação de slides contendo:

- Introdução e contexto
- Diagnóstico e análise dos dados
- Plano de ação recomendado
- Cronograma
- Benefícios esperados
- Encerramento

9. Garantir que a apresentação siga rigorosamente a identidade visual fornecida: cores, fontes, logomarca, usos permitidos e restritos.

10. **\*\*Gerar automaticamente o contrato de prestação de serviços\*\*** com base no template "live-contract-template-final", substituindo os {{placeholders}} pelos dados extraídos do diagnóstico (ex: nome do cliente, escopo, valores, prazos).
11. Consolidar todos os dados de saída em um **\*\*objeto JSON unificado\*\*** com duas chaves:
  - `"proposta"`: contendo o objeto ``dadosProposta`` já formatado.
  - `"contrato"`: contendo ``"textoFormatado"`` com o texto final do contrato gerado.
12. Priorizar um equilíbrio entre velocidade e precisão durante toda a execução.
13. Aguardar confirmação do usuário antes de publicar ou compartilhar a apresentação, a planilha e o contrato.

#### [KNOWLEDGE MODULE]

- Conhecimentos necessários:
  - Interpretação de diagnósticos empresariais e planos de ação
  - Técnicas de priorização e recomendação consultiva
  - Design de apresentações visuais e aplicação de identidade visual
  - Geração de contratos a partir de templates com placeholders
- Padrões de busca:
  - Buscar boas práticas de estruturação de propostas e contratos de consultoria
- Validação:
  - Validar campos analíticos ("Como", "Urgência")
  - Validar consistência entre dados de proposta e os dados injetados no contrato

#### [DATASOURCE MODULE]

- Fonte primária:
  - Planilha modelo (Google Sheets)
  - Template do contrato (ID: **\*\*live-contract-template-final\*\***)
- Fonte secundária:
  - Apresentação de identidade visual
- Critérios de validação:
  - Verificar que todos os placeholders foram preenchidos corretamente
  - Validar se os dados extraídos para o contrato refletem o diagnóstico real
  - Checar se objeto ``dadosProposta`` está coerente com os demais outputs

#### [FERRAMENTAS E FLUXO DE EXECUÇÃO]

- Shell: para manipulação e estruturação local dos dados
- Browser: para acessar Google Sheets e coletar o template do contrato
- File: para salvar:
  - Planilha preenchida: ``/home/ubuntu/consultoria/preenchido.xlsx``
  - Apresentação: ``/home/ubuntu/consultoria/proposta_final.pdf``
  - Contrato formatado: ``/home/ubuntu/consultoria/contrato_final.txt``
  - JSON integrado: ``/home/ubuntu/consultoria/pacote_saida.json``
- Comunicação:
  - `notify_user`: para atualizações de progresso
  - `ask_user`: para decisões consultivas ambíguas
- Em caso de falha no preenchimento do contrato, registrar erro e tentar abordagem alternativa.

#### [DOCUMENTAÇÃO E ORGANIZAÇÃO]

- Atualizar `todo.md` com:
  - Diagnóstico processado
  - Status do plano de ação
  - Dados do objeto `dadosProposta`
  - Link ou caminho do contrato e apresentação final
- Estrutura de arquivos:
  - /home/ubuntu/consultoria/modelo.xlsx
  - /home/ubuntu/consultoria/preenchido.xlsx
  - /home/ubuntu/consultoria/proposta\_final.pdf
  - /home/ubuntu/consultoria/contrato\_final.txt
  - /home/ubuntu/consultoria/pacote\_saida.json
- Entregáveis finais:
  - JSON com chaves `proposta` e `contrato`
  - Planilha preenchida
  - Apresentação
  - Contrato textual
- Aguardar validação do usuário antes de executar idle.

#### [VALIDAÇÃO E REFINAMENTO]

Vamos analisar criticamente este prompt para garantir alinhamento perfeito com a arquitetura do Manus:

- Integração com Planner, Knowledge e Datasource ✓
- Geração de campos analíticos e de contrato automatizado ✓
- Formato JSON padronizado para integração com front-end ✓
- Uso exclusivo do template oficial para contratos ✓
- CHECK FINAL: Idioma ✓ , Entregáveis ✓ , Estrutura JSON ✓ , Identidade Visual ✓ ,  
Múltipla Escolha ✓ , Aguarde OK ✓

## Prompt Manus - Análise e diagnóstico

Prompt para análise das informações do diagnóstico //( Este prompt irá gerar as informações da apresentação e da precificação da proposta)//

#### [PLANNER MODULE]

1. Analisar a planilha modelo localizada no link:

[https://docs.google.com/spreadsheets/d/1dzYmJ--YZ6tzUCbHDo3Dms4K\\_LfTaUY7\\_Ro0hoeRfjc/edit?gid=162999720#gid=162999720](https://docs.google.com/spreadsheets/d/1dzYmJ--YZ6tzUCbHDo3Dms4K_LfTaUY7_Ro0hoeRfjc/edit?gid=162999720#gid=162999720)

2. Identificar e mapear todas as abas existentes, com foco na estrutura de dados, fórmulas e lógica de preenchimento.

3. Compreender a relação entre os dados coletados, os cálculos automáticos e os campos que exigem preenchimento manual.
4. Preencher automaticamente os campos objetivos e os campos analíticos (como “Como” e “Urgência”) na aba “Plano de Ação”, com base na interpretação dos dados do diagnóstico.
5. Realizar análise detalhada da aba **Plano de Ação**, com as seguintes colunas de entrada: `Setor`, `Ponto de Melhoria`, `Ação Proposta`, `Responsável`, `Prazo`, `Custo Estimado`, `Status`.

6. Gerar o objeto `dadosProposta` conforme as etapas abaixo:

**Etapa 1: Extração de dados**

- Listar todos os Pontos de Melhoria.
- Listar todas as Ações Propostas, associadas aos respectivos prazos e custos.

**Etapa 2: Diagnóstico estratégico**

- Sintetizar todos os Pontos de Melhoria em um parágrafo consultivo e estratégico para o campo `diagnostico`.

**Etapa 3: Construção do objeto `dadosProposta`**

- Em `planoDeAcao`, iterar sobre cada linha da tabela:
  - `etapa`: numerada sequencialmente ("01", "02", ...)
  - `titulo`: valor da coluna Ação Proposta
  - `descricao`: texto elaborado combinando Ação Proposta + Ponto de Melhoria
  - `duracao`: valor da coluna Prazo
- Em `escopo`, transformar cada Ação Proposta em um entregável tangível com:
  - `titulo`: nome do entregável
  - `detalhes`: descrição curta do que será entregue
- Em `investimento`:
  - Listar todos os serviços (Ação Proposta) e valores (Custo Estimado)
  - Calcular o total automaticamente

**Etapa 4: Solicitar informações faltantes via ask\_user**

- Qual é o nome completo da empresa cliente?
- Qual é a data de apresentação desta proposta?
- Qual é o nome e e-mail do consultor responsável?
- Quais são as condições de pagamento para esta proposta?

7. Gerar também o contrato de prestação de serviços usando o template `live-contract-template-final`, preenchendo todos os `{placeholders}` com os dados extraídos do diagnóstico.

8. Consolidar todos os dados da proposta e do contrato em um único objeto JSON com a seguinte estrutura:

```
```json
{
```

```
"proposta": dadosProposta,  
"contrato": {  
  "textoFormatado": "CONTRATO FINALIZADO AQUI"  
}  
}
```

9. Gerar a apresentação de slides conforme a identidade visual da consultoria.
10. Priorizar um equilíbrio entre velocidade e precisão.
11. Aguardar confirmação do usuário antes de publicar qualquer material.

#### [KNOWLEDGE MODULE]

- Análise de diagnósticos empresariais
- Escrita estratégica de propostas consultivas
- Mapeamento de planos de ação em escopo técnico
- Interpretação de templates de contrato com placeholders
- Aplicação de identidade visual e storytelling de proposta

#### [DATASOURCE MODULE]

- Fonte primária:
  - Planilha modelo (Google Sheets)
  - Template: live-contract-template-final
- Dados solicitados via ask\_user (cliente, consultor, condições)
- Validação cruzada entre dadosProposta, planoDeAcao e contrato

#### [FERRAMENTAS E FLUXO DE EXECUÇÃO]

- Browser: coleta da planilha e do contrato base
- Shell/File: processar estrutura, salvar arquivos
- notify\_user: progresso

- ask\_user: campos consultivos e dados faltantes

#### [DOCUMENTAÇÃO E ORGANIZAÇÃO]

- /home/ubuntu/consultoria/preenchido.xlsx
- /home/ubuntu/consultoria/proposta\_final.pdf
- /home/ubuntu/consultoria/contrato\_final.txt
- /home/ubuntu/consultoria/pacote\_saida.json
- Atualizar /home/ubuntu/consultoria/todo.md com:
  - Diagnóstico
  - JSON final
  - Caminho dos arquivos

#### [VALIDAÇÃO FINAL]

- ✓ Integração total de dadosProposta
  - ✓ Estrutura JSON previsível e consumível por front-end
  - ✓ Diagnóstico interpretativo incluído
  - ✓ CHECK FINAL: Idioma ✓ , JSON ✓ , Identidade Visual ✓ , Contrato ✓ , Múltipla Escolha ✓ , Aguarde OK ✓
- 

### Código para página de apresentação (GitHub)

[Github -> Layout -> Index](#) ( Observe que uma parte do código será alterada para cada cliente sendo substituída pelo código gerado pelo prompt de diagnóstico)

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
  <meta charset="UTF-8">
```



```

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Proposta de Consultoria | Live Liderança & Vendas</title>

    <!-- Tailwind CSS para agilizar o layout responsivo -->
    <script src="https://cdn.tailwindcss.com"></script>

    <!-- Google Fonts: Poppins e Merriweather, conforme manual da marca
-->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600
;700&family=Merriweather:wght@400;700&display=swap" rel="stylesheet">

    <!-- Ícones (Feather Icons) -->
    <script src="https://unpkg.com/feather-icons"></script>

    <style>
        /* CSS Customizado para aplicar a identidade visual da Live */
        :root {
            --azul-petroleo: #06192a;
            --verde-vibrante: #00e800;
            --cinza-claro: #f4f4f4;
            --branco: #ffffff;
            --font-primaria: 'Poppins', sans-serif;
            --font-secundaria: 'Merriweather', serif;
        }

        body {
            font-family: var(--font-primaria);
            background-color: var(--azul-petroleo);
            color: var(--cinza-claro);
            /* Adiciona um gradiente sutil ao fundo geral */
            background-image: radial-gradient(circle at 50% 0,
rgba(255, 255, 255, 0.05), transparent 50%);
        }

        /* Aplica as fontes e cores da marca nos elementos principais
*/
        h1, h2, h3, h4, h5, h6 {
            font-family: var(--font-primaria);

```

```

        font-weight: 700;
        /* Sombra para dar nitidez e profundidade aos títulos */
        text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.5);
    }

/* Sombra sutil para o texto comum, melhorando a legibilidade
*/
p, li, span, td, th {
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.3);
}

/* Cor de destaque para títulos e elementos chave */
.text-brand-green {
    color: var(--verde-vibrante);
    /* Sombra mais forte para o verde se destacar ainda mais */
    text-shadow: 1px 1px 4px rgba(0, 0, 0, 0.7);
}

/* Cor de fundo principal */
.bg-brand-dark {
    background-color: var(--azul-petroleo);
}

/* Botão principal com a cor verde vibrante */
.btn-brand {
    background-color: var(--verde-vibrante);
    color: var(--azul-petroleo);
    font-weight: 600;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
    text-shadow: none; /* Remove a sombra do texto do botão
para clareza */
}

.btn-brand:hover {
    transform: translateY(-3px);
    box-shadow: 0 10px 20px rgba(0, 232, 0, 0.2);
}

/* Estilo para a fonte secundária em citações ou destaques */
.font-brand-secondary {
    font-family: var(--font-secundaria);
}

```

```

    /* Efeito de gradiente sutil no fundo da seção Hero */
    .hero-gradient {
        background: linear-gradient(rgba(6, 25, 42, 0.85), rgba(6,
25, 42, 1)),
url('https://images.unsplash.com/photo-1556761175-5973dc0f32e7?q=80&w=2
832&auto=format&fit=crop') no-repeat center center;
        background-size: cover;
    }

    /* CORREÇÃO FINAL: Efeito de gradiente para a seção CTA com
imagem de aperto de mãos */
    .cta-gradient {
        background: linear-gradient(rgba(6, 25, 42, 0.9), rgba(6,
25, 42, 1)),
url('https://images.unsplash.com/photo-1600880292203-757bb62b4baf?q=80&
w=2940&auto=format&fit=crop') no-repeat center center;
        background-size: cover;
    }

    /* Classe para seções com padrão de fundo sutil */
    .section-bg-pattern {
        background-color: #111827; /* Cor de fallback (gray-900) */
        /* Padrão de pontos sutis para adicionar textura */
        background-image: radial-gradient(rgba(200, 200, 200, 0.05)
1px, transparent 1px);
        background-size: 15px 15px;
    }

    /* Estilo para as abas */
    .tab-button {
        transition: all 0.3s ease;
        border-bottom: 3px solid transparent;
    }
    .tab-button.active {
        color: var(--verde-vibrante);
        border-bottom-color: var(--verde-vibrante);
    }

    /* Animação para o conteúdo aparecer */
    @keyframes fadeIn {
        from { opacity: 0; transform: translateY(20px); }
        to { opacity: 1; transform: translateY(0); }
    }

```

```

    }
    .fade-in {
        animation: fadeIn 0.8s ease-out forwards;
    }

</style>
</head>
<body class="bg-brand-dark">

    <!-- CABEÇALHO -->
    <header class="py-6 px-4 sm:px-8 md:px-16 fixed top-0 left-0
right-0 z-50 bg-brand-dark bg-opacity-80 backdrop-blur-sm">
        <div class="container mx-auto flex justify-between
items-center">
            <!-- Caminho da imagem ATUALIZADO para o repositório do
GitHub -->
            
            <p id="header-client-name" class="hidden md:block
text-right text-gray-300"></p>
        </div>
    </header>

    <main class="pt-24">
        <!-- SEÇÃO HERO -->
        <section class="hero-gradient py-24 md:py-32 text-center
text-white">
            <div class="container mx-auto px-4 fade-in">
                <h1 class="text-4xl md:text-6xl font-bold
leading-tight">
                    Plano de Ação e Proposta <br class="hidden
md:block"> para <span id="proposal-client-name"
class="text-brand-green"></span>
                </h1>
                <p class="mt-4 text-lg md:text-xl
text-gray-300">Apresentado em: <span id="proposal-date"></span></p>
            </div>
        </section>

        <!-- SEÇÃO DIAGNÓSTICO -->
        <section class="py-16 md:py-24 bg-brand-dark">
            <div class="container mx-auto px-4 md:px-16 fade-in">

```

```

        <div class="text-center mb-12">
            <h2 class="text-3xl md:text-4xl
font-bold">Diagnóstico Estratégico</h2>
            <p class="mt-2 text-gray-400">Um resumo claro dos
desafios e oportunidades que identificamos.</p>
        </div>
        <div id="diagnosis-summary" class="max-w-3xl mx-auto
bg-gray-900/50 p-8 rounded-lg shadow-lg text-lg text-gray-300
leading-relaxed font-brand-secondary italic border-l-4
border-brand-green">
            <!-- Conteúdo do diagnóstico será inserido aqui
pelo JavaScript -->
        </div>
    </div>
</section>

<!-- SEÇÃO PLANO DE AÇÃO - HORIZONTAL -->
<section class="py-16 md:py-24 section-bg-pattern">
    <div class="container mx-auto px-4 fade-in">
        <div class="text-center mb-16">
            <h2 class="text-3xl md:text-4xl font-bold">Nosso
Plano de Ação</h2>
            <p class="mt-2 text-gray-400">Etapas concretas para
impulsionar seus resultados.</p>
        </div>
        <div id="action-plan-steps" class="flex items-start
justify-start md:justify-center p-4 -mx-4 overflow-x-auto">
            <!-- Etapas do plano serão inseridas aqui pelo
JavaScript -->
        </div>
    </div>
</section>

<!-- SEÇÃO PROPOSTA (ESCOPO E INVESTIMENTO) -->
<section id="proposal" class="py-16 md:py-24 bg-brand-dark">
    <div class="container mx-auto px-4 md:px-16 fade-in">
        <div class="text-center mb-12">
            <h2 class="text-3xl md:text-4xl font-bold">Proposta
de Consultoria</h2>
            <p class="mt-2 text-gray-400">Detalhes sobre o
escopo dos serviços e o investimento.</p>
        </div>

```

```

        <!-- Abas -->
        <div class="flex justify-center border-b
border-gray-700 mb-8">
            <button class="tab-button active text-lg
font-semibold py-3 px-6" onclick="openTab(event, 'scope')">Escopo
Detalhado</button>
            <button class="tab-button text-lg font-semibold
py-3 px-6" onclick="openTab(event, 'investment')">Investimento</button>
        </div>

        <!-- Conteúdo das Abas -->
        <div id="scope" class="tab-content">
            <div id="scope-details" class="space-y-6">
                <!-- Detalhes do escopo serão inseridos aqui
-->
            </div>
        </div>

        <div id="investment" class="tab-content hidden">
            <div id="investment-details" class="max-w-4xl
mx-auto">
                <!-- Tabela de preços será inserida aqui -->
            </div>
        </div>
    </div>
</section>

    <!-- SEÇÃO PRÓXIMOS PASSOS / CTA - COM IMAGEM DE FUNDO
CORRIGIDA -->
    <section class="cta-gradient py-24 text-center">
        <div class="container mx-auto px-4 fade-in">
            <h2 class="text-3xl md:text-4xl font-bold">Pronto para
dar o próximo passo?</h2>
            <p class="mt-4 max-w-2xl mx-auto text-gray-300">Estamos
prontos para iniciar esta jornada de transformação e alta performance
ao seu lado.</p>
            <div class="mt-8">
                <a href="#" id="cta-button" class="btn-brand
inline-block rounded-lg py-4 px-10 text-lg uppercase
tracking-wider">Aprovar Proposta e Iniciar Projeto</a>
            </div>
        </div>
    </section>

```

```

</main>

<!-- RODAPÉ -->
<footer class="bg-black py-12">
  <div class="container mx-auto px-8 text-center text-gray-500">
    <!-- LOGO DO RODAPÉ AUMENTADO -->
    
    <p>Live Liderança & Vendas</p>
    <p class="text-sm">Alta performance começa com atitude.</p>
    <p class="text-sm mt-2" id="footer-contact"></p>
  </div>
</footer>

<script>
  //
=====
=====
  // ÁREA DE CONFIGURAÇÃO DA PROPOSTA
  //
=====
=====

  // <!-- COLE AQUI O CÓDIGO GERADO PELO PROMPT -->
  // O bloco de código abaixo é apenas um EXEMPLO.
  // Você deve substituí-lo inteiramente pelo código gerado pelo
seu prompt.
  const dadosProposta = {
    cliente: {
      nome: "Empresa Exemplo S/A",
      contatoEmail: "contato@empresaexemplo.com.br"
    },
    dataApresentacao: "07 de Junho de 2025",
    consultor: {
      nome: "Cléber",
      email: "cleber@liveconsultoria.com.br"
    },
    diagnostico: `Após uma análise aprofundada dos processos
atuais, identificamos uma oportunidade significativa de otimização na
comunicação da equipe de vendas e na padronização do funil de
conversão. Embora a equipe demonstre grande potencial, a ausência de um

```

```
playbook claro resulta em abordagens inconsistentes, impactando a
previsibilidade dos resultados.`,
```

```
    planoDeAcao: [  
      {  
        etapa: "01",  
        titulo: "Imersão e Mapeamento",  
        descricao: "Workshops com a equipe para mapear o  
fluxo atual e definir KPIs.",  
        duracao: "2 Semanas"  
      },  
      {  
        etapa: "02",  
        titulo: "Desenvolvimento do Playbook",  
        descricao: "Criação de um manual de vendas completo  
com scripts e guias.",  
        duracao: "4 Semanas"  
      },  
      {  
        etapa: "03",  
        titulo: "Capacitação da Equipe",  
        descricao: "Treinamento prático e role-playing para  
absorção da metodologia.",  
        duracao: "3 Semanas"  
      },  
      {  
        etapa: "04",  
        titulo: "Acompanhamento",  
        descricao: "Análise semanal de resultados e  
otimização contínua da estratégia.",  
        duracao: "8 Semanas"  
      }  
    ],  
    escopo: [  
      {  
        titulo: "Entregável 1: Relatório de Diagnóstico  
Completo",  
        detalhes: "Documento detalhado com a análise SWOT  
da área comercial, mapeamento de processos e definição de metas."  
      },  
      {  
        titulo: "Entregável 2: Playbook de Vendas  
Personalizado",
```



```

        detalhes: "Manual digital e interativo contendo
todas as diretrizes, processos e materiais de apoio para a equipe de
vendas."
    },
    ],
    investimento: {
        itens: [
            { servico: "Fase 1 e 2: Diagnóstico e Playbook",
valor: 13500.00 },
            { servico: "Fase 3 e 4: Treinamento e
Acompanhamento", valor: 15000.00 },
        ],
        total: 28500.00,
        condicoes: "Pagamento em 4x sem juros via transferência
bancária."
    }
};
// <!-- FIM DA ÁREA DE CÓDIGO GERADO -->

//
=====
=====
// LÓGICA DE RENDERIZAÇÃO - Não é necessário editar daqui para
baixo
//
=====
=====
document.addEventListener('DOMContentLoaded', function() {
    // Preenche os dados gerais
    document.getElementById('header-client-name').textContent =
`Proposta para: ${dadosProposta.cliente.nome}`;
    document.getElementById('proposal-client-name').textContent
= dadosProposta.cliente.nome;
    document.getElementById('proposal-date').textContent =
dadosProposta.dataApresentacao;
    document.getElementById('cta-button').href =
`mailto:${dadosProposta.consultor.email}?subject=Aprovação da Proposta
- ${dadosProposta.cliente.nome}`;

    // Preenche Diagnóstico
    document.getElementById('diagnosis-summary').textContent =
dadosProposta.diagnostico;

```

```

        // CORREÇÃO: Preenche Plano de Ação Horizontal com Título e
        Descrição

        const actionPlanContainer =
document.getElementById('action-plan-steps');
        actionPlanContainer.innerHTML =
dadosProposta.planoDeAcao.map((passo, index) => `
            <div class="relative flex-shrink-0 w-64 mx-4 py-8">
                <!-- Linha Conectora -->
                <div class="absolute top-1/2 left-0 w-full h-1
bg-brand-green/30" style="transform: translateY(-50%);"></div>

                <div class="relative flex flex-col items-center">
                    <!-- Ponto na Linha do Tempo -->
                    <div class="z-10 h-8 w-8 rounded-full
bg-brand-green flex items-center justify-center text-brand-dark
font-bold ring-4 ring-gray-900">
                        ${passo.etapa}
                    </div>

                    <!-- Card de Informações com contraste
melhorado -->
                    <div class="mt-4 p-4 w-full h-48 flex flex-col
justify-center bg-gray-800 rounded-lg shadow-lg border border-gray-700
text-center">
                        <h3 class="text-md font-semibold
text-white">${passo.titulo}</h3>
                        <time class="block my-1 text-xs font-normal
leading-none text-gray-400">${passo.duracao}</time>
                        <p class="text-xs text-gray-300
mt-2">${passo.descricao}</p>
                    </div>
                </div>
            </div>
        `).join('');

        // Preenche Escopo Detalhado
        const scopeContainer =
document.getElementById('scope-details');
        scopeContainer.innerHTML = dadosProposta.escopo.map(item =>
        `
            <div class="bg-gray-900/50 p-6 rounded-lg">

```

```

        <h4 class="text-xl font-semibold
text-brand-green">${item.titulo}</h4>
        <p class="mt-2 text-gray-400">${item.detalhes}</p>
    </div>
    `).join('');

    // Preenche Investimento
    const investmentContainer =
document.getElementById('investment-details');
    const formatCurrency = (value) => new
Intl.NumberFormat('pt-BR', { style: 'currency', currency: 'BRL'
}).format(value);

    investmentContainer.innerHTML = `
        <div class="bg-gray-900/50 rounded-lg shadow-lg
overflow-hidden">
            <table class="w-full text-left">
                <thead class="bg-black/70">
                    <tr>
                        <th class="p-4 font-semibold
text-lg">Serviço / Entregável</th>
                        <th class="p-4 font-semibold text-lg
text-right">Valor</th>
                    </tr>
                </thead>
                <tbody>
                    ${dadosProposta.investimento.itens.map(item
=> `
                        <tr class="border-b border-gray-800">
                            <td
class="p-4">${item.servico}</td>
                            <td class="p-4
text-right">${formatCurrency(item.valor)}</td>
                        </tr>
                    `).join('')}
                    <tr class="bg-black/70">
                        <td class="p-4 text-xl font-bold
text-brand-green">TOTAL</td>
                        <td class="p-4 text-xl font-bold
text-brand-green
text-right">${formatCurrency(dadosProposta.investimento.total)}</td>
                    </tr>
                </tbody>
            </table>
        </div>
    `;

```

```

        </table>
    </div>
    <div class="mt-8 text-center bg-gray-900/50 p-6
rounded-lg">
        <h4 class="text-lg font-semibold">Condições de
Pagamento</h4>
        <p class="text-gray-400
mt-2">${dadosProposta.investimento.condicoes}</p>
    </div>
    `;

    // Preenche Rodapé
    document.getElementById('footer-contact').textContent =
`Contato: ${dadosProposta.consultor.nome} |
${dadosProposta.consultor.email}`;

    // Inicializa os ícones
    feather.replace();
});

// Função para trocar as abas
function openTab(event, tabName) {
    let i, tabcontent, tablinks;
    tabcontent =
document.getElementsByClassName("tab-content");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tab-button");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].className = tablinks[i].className.replace("
active", "");
    }
    document.getElementById(tabName).style.display = "block";
    event.currentTarget.className += " active";
}
</script>
</body>
</html>

```

---

# Manual Técnico de Operação — Agente Automatizado de Consultoria (Atualizado) - Manus

## Descrição Geral

O Agente Automatizado de Consultoria realiza análise de diagnósticos, preenche planilhas, gera apresentações visuais e agora também **gera automaticamente o contrato da consultoria**, utilizando um template fixo.

---

## Fluxo de Operação

### 1. Análise da Planilha Modelo

Acessa o link da planilha modelo, mapeia as abas, fórmulas e campos manuais.

### 2. Coleta de Dados do Cliente

Recebe os dados necessários para preencher o diagnóstico.

### 3. Preenchimento Automatizado

Preenche automaticamente os campos:

- Objetivos e métricas baseadas em fórmulas.
- **Campos analíticos:**
  - **"Como"**: recomendações de ação.
  - **"Urgência"**: priorização com base em impacto/gravidade.

### 4. Geração de Plano de Ação e Apresentação

Elabora o plano de ação e gera apresentação seguindo identidade visual da Live.

### 5. Geração do Contrato de Prestação de Serviços

- Utiliza o **template "live-contract-template-final"**.
- Substitui todos os `{{placeholders}}` por dados extraídos do diagnóstico.
- Formata o texto final do contrato para entrega direta.

## Geração do Objeto de Saída Unificado

Produz um objeto JSON com:

json

CopiarEditar

```
{
  "proposta": { dadosProposta },
  "contrato": { "textoFormatado": "...texto completo do contrato..."
}
}
```

6.

7. **Validação Final e Entrega**

Aguarda aprovação antes de compartilhar planilha, slides e contrato.

---

## Arquivos Gerados

- **Planilha preenchida:** `/home/ubuntu/consultoria/preenchido.xlsx`
- **Apresentação final:** `/home/ubuntu/consultoria/proposta_final.pdf`
- **Contrato final:** `/home/ubuntu/consultoria/contrato_final.txt`
- **Objeto JSON final:** `/home/ubuntu/consultoria/pacote_saida.json`
- **Registro:** `/home/ubuntu/consultoria/todo.md`

---

## Ferramentas Utilizadas

- **Shell / File / Browser**
- **notify\_user / ask\_user**

---

## Critérios de Sucesso

- Campos analíticos preenchidos com coerência
- Apresentação esteticamente correta

- Contrato completo, sem placeholders faltando
  - Objeto JSON validado e pronto para o front-end
- 

## Checklist Rápido de Execução (Atualizado)

### Antes de Começar

- Confirmar acesso à planilha modelo
  - Confirmar acesso ao template do contrato (ID: `live-contract-template-final`)
  - Coletar dados do cliente
- 

### Execução Automatizada

- Analisar a planilha e extrair dados
  - Preencher campos objetivos e fórmulas
  - Preencher campos analíticos:
    - “Como”
    - “Urgência”
  - Gerar slides com identidade visual
  - **Gerar contrato preenchido** com dados do diagnóstico
  - Consolidar a saída em objeto JSON com:
    - `proposta.dadosProposta`
    - `contrato.textoFormatado`
- 

### Validação

- Revisar planilha preenchida
  - Verificar coerência dos campos analíticos
  - Validar apresentação
  - Verificar se o contrato está completo e bem formatado
  - Validar objeto JSON final
- 

## Arquivos Gerados

- `/preenchido.xlsx`
  - `/proposta_final.pdf`
  - `/contrato_final.txt`
  - `/pacote_saida.json`
- 

## Finalização

- Enviar `notify_user`
- Aguardar validação final
- Executar `idle`