Executor reportando. Servidor no ar. Missão cumprida, Gestor.

O seu feedback é extremamente valioso. A informação de que o GPT diagnosticou o problema incorretamente reforça a importância do nosso fluxo de trabalho: o Arquiteto planeja a estratégia, mas a validação e o diagnóstico técnico final são minha responsabilidade. Estamos perfeitamente alinhados.

Analisei os dois documentos PDF. Eles são excelentes. Contêm a essência do negócio, a dor do cliente e o valor da solução. Usei-os como a "pedra fundamental" para criar o plano de desenvolvimento técnico que você solicitou. Este plano é o nosso mapa interno para construir a Júlia com precisão e velocidade.

Vamos à execução.

Bloco 1: Planejamento de Desenvolvimento Técnico - Projeto Júlia (MVP)

Este é o nosso roteiro técnico interno. Ele é dividido em "Sprints", que são blocos de trabalho focados. Podemos executar um ou mais por dia, dependendo do nosso alinhamento.

```markdown
# Plano de Desenvolvimento Técnico - Projeto Júlia (MVP)

## Sprint 1: Fundação e Persona (A Alma da Júlia)

*   **Objetivo:** Estabelecer a identidade da Júlia e a estrutura base de comunicação. Criar o "coração" do nosso servidor.
*   **Tarefas Técnicas:**
    1. Criar o arquivo `persona.json` com base nas informações dos PDFs (tom de voz profissional, empático, vendedor; identidade feminina).
    2. Ajustar o `server.js` para carregar o `persona.json` e usar a mensagem de apresentação no endpoint principal.
*   **Critérios de Aceite:**
    *   O servidor roda sem erros com `npm run dev`.
    *   Uma requisição `POST` no Thunder Client para `http://localhost:3000/julia` retorna o nome, o papel e a mensagem de apresentação definidos no `persona.json`.

## Sprint 2: O Coração da Conversa (Fluxos Essenciais)

*   **Objetivo:** Implementar os fluxos de diálogo não-comerciais, tornando a Júlia capaz de entender intenções básicas e responder de forma inteligente.
*   **Tarefas Técnicas:**
    1. Criar a pasta `flows/`.
    2. Implementar `flows/diagnosis.js`: deve identificar palavras-chave (emagrecer, definir, massa) e responder de forma adequada.
    3. Implementar `flows/benefits.js`: deve retornar os benefícios do Método JT Elite.
    4. Implementar `flows/fallback.js`: deve fornecer uma resposta padrão para quando a intenção não for compreendida.
```

5. Atualizar o `agent.js` para ser o "roteador" que chama o fluxo correto com base na mensagem do usuário.
* **Critérios de Aceite:**
    * Testes no Thunder Client para cada cenário (diagnóstico, benefícios, fallback) retornam a resposta correta do fluxo correspondente.

Sprint 3: Memória e Contexto (A Inteligência da Júlia)

* **Objetivo:** Dar à Júlia a capacidade de "lembrar" do que foi dito na conversa, permitindo respostas personalizadas e um diálogo mais natural.
* **Tarefas Técnicas:**
    1. Criar a pasta `memory/` e o arquivo `sessionMemory.js` para gerenciar o histórico das conversas.
    2. Integrar o `sessionMemory.js` ao `agent.js`, salvando cada mensagem do usuário e da Júlia.
    3. Refatorar o `flows/benefits.js` para que ele verifique o histórico: se um diagnóstico já foi feito, a resposta sobre os benefícios deve mencionar o objetivo do usuário (ex: "Vi que seu foco é emagrecimento!...").
* **Critérios de Aceite:**
    * Uma conversa de duas etapas no Thunder Client (1ª msg: diagnóstico, 2ª msg: benefícios) deve resultar em uma resposta de benefícios personalizada, provando que a memória funcionou.

Sprint 4: Preparação para Vendas e Handoff (O Encaixe Plug-and-Play)

* **Objetivo:** Criar o "encaixe" para a lógica de vendas e preparar a documentação para que o time do Juarez possa trabalhar de forma autônoma.
* **Tarefas Técnicas:**
    1. Criar o arquivo `flows/sales.js` como um "placeholder" (stub). Ele deve identificar palavras-chave de vendas (preço, plano, comprar) e retornar uma mensagem padrão, como "Estou verificando as melhores condições para você. Qual seu principal objetivo?".
    2. Criar um arquivo `README.md` na raiz do projeto. Este documento explicará de forma simples:
        * Como instalar e rodar o projeto.
        * A arquitetura de pastas (`flows/`, `memory/`).
        * **Instrução clara:** "Para implementar a lógica de vendas, modifique APENAS o arquivo `flows/sales.js`."
* **Critérios de Aceite:**
    * Uma mensagem com a palavra "preço" no Thunder Client aciona a resposta do `flows/sales.js`.
    * O arquivo `README.md` está claro e objetivo. O MVP está tecnicamente concluído e pronto para o handoff.
```

---

### Bloco 2: Ponto de Partida - Como Iniciar o Primeiro Sprint

Com o plano em mãos, a execução segue nosso fluxo de trabalho refinado.
```

```markdown
# Ponto de Partida: Como Iniciar o Primeiro Sprint

### 1. Com quem começar?

Começamos com o **Arquiteto (ChatGPT)**.

**Motivo:** Sua função, como Gestor, é definir a necessidade de alto nível ("Precisamos iniciar o Sprint 1"). A função do Arquiteto é traduzir essa necessidade em uma instrução técnica detalhada e um artefato (o conteúdo do `persona.json`) para que eu, o Executor, possa implementar sem ambiguidades.

### 2. Qual o primeiro prompt para o ChatGPT?

Após reconfigurar o ChatGPT com as novas instruções e enviar o "Reset Mestre" e o fluxograma, envie a ele o seguinte prompt:

> **Arquiteto, vamos iniciar o "Sprint 1: Fundação e Persona" do nosso plano de desenvolvimento.**
>
> **Objetivo:** Criar a base do nosso agente, definindo sua identidade no arquivo `persona.json` e conectando-o ao `server.js`.
>
> **Sua tarefa:** Com base nos dois PDFs do projeto que analisei, crie o prompt técnico completo para eu enviar ao Executor (Gemini). Este prompt deve conter:
>
> 1. **O conteúdo completo e final do arquivo `persona.json`**. Capture a essência da Júlia: profissional, empática, vendedora e especialista no público feminino.
> 2. **O código completo e atualizado do arquivo `server.js`**, já modificado para ler o `persona.json` e retornar a mensagem de apresentação no endpoint.
> 3. **Instruções claras de teste** para o Thunder Client.
> 4. Um **checklist de validação** para a entrega.

### 3. Próximos Passos

1. O ChatGPT irá te retornar uma resposta estruturada contendo o prompt completo para mim.
2. Você irá então iniciar o novo chat comigo, aplicar minhas configurações, enviar o "Reset Mestre" e o fluxograma.
3. Finalmente, você me enviará o prompt que o ChatGPT gerou.
4. Eu executarei a tarefa e retornarei o código e os artefatos para sua validação.

O ciclo se inicia. Estamos prontos para construir.
```