# Aula 01 — Vibe Coding: Criando o Frontend do WEB3 TODO

## 🧠 Contexto da Aula

Hoje é a primeira aula do nosso projeto \*Vibe Coding\* e vamos começar desenvolvendo o \*\*frontend do WEB3 TODO\*\*, um aplicativo de gerenciamento de tarefas com integração Web3. O foco será criar a \*\*interface de usuário (UI)\*\* moderna, responsiva e funcional, que futuramente será conectada a uma carteira digital (como a MetaMask).

Estamos utilizando um \*\*prompt descritivo e completo\*\* para guiar o design e os componentes da tela. Essa abordagem nos ajuda a manter foco no usuário, na usabilidade e no propósito do projeto, mesmo antes de termos backend ou lógica de contrato inteligente funcionando.

---

## 📝 Prompt Completo do Projeto

# WEB3 TODO — Interface de Usuário

## Objetivo

Desenvolver a interface de usuário do "WEB3 TODO", um aplicativo de gerenciamento de tarefas com integração Web3. O sistema deve permitir que os usuários conectem sua carteira digital, criem e acompanhem tarefas, e visualizem métricas em tempo real, com foco em usabilidade e design moderno.

---

## Estrutura da Interface

### Cabeçalho

- Exibir no topo da página um botão de destaque: "Conectar Carteira"

- Abaixo, apresentar uma mensagem de alerta:

"Conecte sua carteira para gerenciar suas tarefas na blockchain."

- O botão e o alerta devem indicar visualmente o status da conexão com a carteira.

### Seção de Métricas

Criar uma grade horizontal com 4 cards, exibindo:

1. Total de Tarefas

2. Tarefas Concluídas

3. Tarefas Pendentes

4. Wei em Stake

Os cards devem utilizar cores que remetam à tecnologia, como azul cobalto, ciano elétrico, violeta digital, com ícones informativos.

### Seção de Tarefas

- Título da seção: "Tarefas"

- Ao lado, um botão destacado: "Nova Tarefa"

- Lista de tarefas exibida como cards horizontais, contendo:

- Nome da tarefa

- Status: Pendente ou Concluído

- Descrição

- Data de criação

- Data de conclusão (se aplicável)

- Valor em wei associado

- Ícone de ação para "Concluir Tarefa" (sem editar ou excluir)

Os cards devem realçar as ações disponíveis e diferenciar visualmente tarefas pendentes e concluídas.

---

## Diretrizes de Design

### Estilo Visual

- Interface clean e moderna, com tema claro (light mode)

- Aplicar a regra 60-30-10:

- Utilizar apenas 2 cores

- Tipografia clara, com hierarquia visual definida

- Design minimalista com espaçamento adequado

- Feedback visual para ações, erros e carregamentos

- Responsivo para dispositivos variados

### Clareza e Consistência

- Rótulos descritivos e ícones reconhecíveis

- Coerência em cores, tamanhos, espaçamentos e fontes

- Destaques visuais consistentes para componentes sensíveis à carteira

### Interatividade

- Fornecer feedback imediato com animações suaves

- Transições fluidas entre estados de interface

- Botões com estados claros: hover, ativo, desativado, carregando

### Acessibilidade

- Contraste de cores adequado

- Textos alternativos para ícones e imagens

- Navegação completa por teclado

- Evitar informações transmitidas apenas por cor

---

## Considerações Finais

A interface deve ser intuitiva para iniciantes em Web3, deixando claro quando funcionalidades exigem a carteira conectada. Todos os elementos interativos devem responder com feedback visual e manter consistência em toda a experiência do usuário.

---

## 🚀 Objetivo do Prompt

> "Criar uma interface de usuário para o 'WEB3 TODO', um aplicativo de gerenciamento de tarefas baseado em tecnologia Web3."

O objetivo é treinar a construção de \*\*UI baseada em requisitos reais de produto\*\*, algo comum no mundo do desenvolvimento front-end. Esse prompt simula o pedido de um \*\*designer de produto ou PO\*\*, e cabe a nós, como devs front-end, transformar isso em uma interface funcional.

---

## 🧩 Explicação de Cada Seção do Prompt

### 🔝 Cabeçalho

- \*\*Botão 'Conectar Carteira'\*\*:

- Introduz o conceito de integração com Web3.

- Mostra como lidar com ações que dependem de estado (ex: conectado ou não).

- \*\*Mensagem de Alerta\*\*:

- Fornece \*\*feedback contextual\*\* ao usuário sobre a necessidade de conexão.

- Ensina como trabalhar com renderização condicional no React/Vue/etc.

\*\*Aprendizado:\*\* como construir interfaces dinâmicas com base em estados lógicos.

---

### 📊 Seção de Métricas

- \*\*Total de Tarefas, Tarefas Concluídas, Tarefas Pendentes, Wei em Stake\*\*:

- Simula \*\*indicadores de performance\*\*, comuns em dashboards.

- Trabalha com \*\*exibição de dados em blocos de forma organizada\*\*.

- Introduz conceitos de layout responsivo (4 cards em linha).

\*\*Aprendizado:\*\* criar layouts flexíveis com componentes reutilizáveis e foco na legibilidade.

---

### ✅ Seção de Lista de Tarefas

- \*\*Título 'Tarefas' e botão 'Nova Tarefa'\*\*:

- Mostra como \*\*habilitar ou desabilitar botões dinamicamente\*\* com base em permissões.

- Interface simplificada mas com lógica de dependência da carteira.

- \*\*Lista em formato de cards horizontais\*\*:

- Ensina criação de componentes estilizados com hierarquia visual clara.

- Pratica \*\*boas práticas de UI para visualização de dados estruturados\*\*.

Cada card inclui:

- Nome, status (pendente/concluído), descrição, datas, valor em wei

- \*\*Ícone de ação único\*\*: apenas para concluir tarefas, sem opções de deletar ou editar

- \*\*Abordagem minimalista\*\*: reduz complexidade e foca na ação principal

- \*\*UI simplificada\*\*: evita sobrecarga de opções desnecessárias

\*\*Aprendizado:\*\* trabalhar com listas de dados, manipulação de estados focados em \*\*ações essenciais\*\* e \*\*design minimalista\*\*.

---

### 🎨 Considerações de Design

- \*\*Design Minimalista e Tema Light\*\*:

- \*\*Abordagem minimalista\*\* para evitar excesso de elementos

- \*\*Tema light\*\* como base visual principal

- \*\*Foco na usabilidade\*\*: destaque dos componentes principais

- \*\*Hierarquia visual clara\*\* sem sobrecarga de informações

- \*\*Paleta Tecnológica e Regra 60-30-10\*\*:

- Utilização de \*\*paleta que remeta à tecnologia\*\* com aplicação da regra de proporcionalidade 60-30-10.

- \*\*60%\*\* cor dominante, \*\*30%\*\* cor secundária, \*\*10%\*\* cor de destaque.

- \*\*Clareza e Consistência\*\*:

- Rótulos descritivos e ícones intuitivos

- Linguagem de design consistente (tipografia, espaçamento, cores)

- \*\*Menos é mais\*\*: priorização de elementos essenciais

- \*\*Feedback e Interação\*\*:

- Estados visuais para ações do usuário

- Indicadores de carregamento para operações assíncronas

- Interface responsiva e fluida

- \*\*Acessibilidade\*\*:

- Contraste de cores adequado

- Textos alternativos para ícones

- Navegação por teclado

\*\*Aprendizado:\*\* criar UIs \*\*minimalistas e funcionais\*\*, priorizando \*\*usabilidade\*\* e \*\*clareza visual\*\* através de design limpo e focado no essencial.

---

### 🔐 Dependência da Carteira

- Componentes que exigem carteira conectada têm \*\*indicação visual clara\*\*.

- Ensina como guiar o usuário quando uma ação está indisponível.

- Importância de \*\*UX explicativo\*\*, não só funcional.

\*\*Aprendizado:\*\* UX writing, micro-copy e posicionamento estratégico de elementos.

---

## 💡 Por que começar pelo Frontend?

- \*\*Validação visual rápida:\*\* já conseguimos "ver" o produto tomando forma.

- \*\*Base sólida para conexão futura com Web3:\*\* já deixamos os ganchos prontos para integração com MetaMask.

- \*\*Ensina lógica de componentes, estados, listas e interações\*\* – pilares fundamentais do front-end moderno.

---

## 🛠️ Tecnologias Sugeridas

Você pode construir essa interface com qualquer stack frontend moderna. Recomenda-se:

- \*\*React com TypeScript\*\*

- \*\*Styled-components ou TailwindCSS\*\*

- \*\*Componentização e design responsivo\*\*

---

## ✨ Missão da Aula

Desenvolver uma \*\*interface limpa, funcional e bonita\*\*, pronta para ser conectada a uma blockchain. Mais que código, queremos criar \*\*experiência e usabilidade\*\*.

> \*"Frontend não é só o que se vê, é como se sente."\* — vibe coding 🧘‍♂️

---