

Proposta de Design System para iSelfToken

1. Introdução: O que é um Design System para iSelfToken?

Um Design System para a iSelfToken será uma coleção de princípios, diretrizes, padrões e componentes reutilizáveis que garantem a consistência visual e funcional em todas as interfaces do sistema/aplicativo. Ele servirá como uma "fonte única de verdade" para designers e desenvolvedores, acelerando o processo de criação, melhorando a qualidade do produto e garantindo uma experiência de usuário coesa e intuitiva, independentemente do módulo ou perfil de usuário.

Princípios Fundamentais:

- **Consistência:** Experiência de usuário unificada em todos os módulos e plataformas (mobile e futura web).
- **Eficiência:** Acelerar o desenvolvimento e o design através da reutilização de componentes e padrões.
- **Escalabilidade:** Facilitar a adição de novas funcionalidades e módulos sem comprometer a integridade do design.
- **Acessibilidade:** Garantir que a plataforma seja utilizável por pessoas com diferentes capacidades.
- **Marca:** Refletir a identidade e os valores da iSelfToken em cada interação.
- **Modularidade:** Alinhar com a arquitetura modular do sistema, permitindo que cada módulo utilize os mesmos blocos de construção.

2. Elementos Essenciais do Design System

2.1. Cores

Uma paleta de cores bem definida é fundamental para a identidade visual e a usabilidade.

- **Paleta Primária:**
 - **Azul Principal (Brand/Ação):** #2563eb (Blue 600 - Tailwind equivalent) - Usado para botões primários, links, ícones de destaque.
 - **Azul Claro (Destaque/Fundo):** #eff6ff (Blue 50 - Tailwind equivalent) - Para fundos de seções, estados de hover.
- **Paleta Neutra (Texto, Fundos, Bordas):**
 - **Cinza Escuro (Texto Principal):** #1e293b (Slate 800 - Tailwind equivalent)
 - **Cinza Médio (Texto Secundário/Ícones):** #475569 (Slate 600 - Tailwind equivalent)

- **Cinza Claro (Bordas/Divisores):** #cbd5e1 (Slate 300 - Tailwind equivalent)
- **Branco (Fundos de Cartões/Elementos):** #ffffff
- **Fundo da Aplicação:** #f8fafc (Slate 50 - Tailwind equivalent)
- **Cores Semânticas (Feedback):**
 - **Sucesso:** Verde (#22c55e - Green 500)
 - **Aviso:** Amarelo/Laranja (#f59e0b - Amber 500)
 - **Erro:** Vermelho (#ef4444 - Red 500)
 - **Informação:** Azul (#3b82f6 - Blue 500)

2.2. Tipografia

A família tipográfica "Inter" é uma excelente escolha para clareza e modernidade.

- **Família:** Inter (sans-serif)
- **Tamanhos de Fonte (escala baseada em rem/pixels para React Native):**
 - Display/H1: 36px - 48px (para títulos de tela principais)
 - H2: 28px - 32px (para subtítulos de seção)
 - H3: 20px - 24px (para títulos de componentes/cards)
 - Corpo (Padrão): 16px
 - Pequeno: 14px (para textos auxiliares, legendas)
 - Extra Pequeno: 12px (para disclaimers, timestamps)
- **Pesos da Fonte:** Regular (400), Medium (500), SemiBold (600), Bold (700)
- **Altura da Linha (Line Height):** Consistente para legibilidade (ex: 1.5 para corpo de texto).

2.3. Espaçamento

Um sistema de espaçamento baseado em uma escala consistente (ex: múltiplos de 4px ou 8px) garante layouts harmoniosos.

- **Escala:** Utilizar uma escala incremental (ex: 4px, 8px, 12px, 16px, 24px, 32px, 48px, 64px).
- **Aplicação:** Usado para margens, paddings, espaçamento entre elementos e componentes.

2.4. Iconografia

Ícones devem ser claros, simples e consistentes no estilo.

- **Estilo:** Ícones de linha (outline) ou sólidos, dependendo do contexto, mas sempre mantendo a consistência.
- **Fonte de Ícones:** Considerar bibliotecas como Font Awesome ou Phosphor Icons (se compatíveis com React Native) ou ícones SVG customizados para maior controle.
- **Uso:** Para navegação, ações, status e elementos visuais de apoio.

2.5. Componentes da Biblioteca (React Native & React Bootstrap)

A biblioteca de componentes será a base para a construção das interfaces. Para React Native, os componentes do React Bootstrap precisarão ser adaptados ou recriados utilizando componentes nativos do React Native (View, Text, Button, Image, etc.) e estilizados para replicar o visual do Bootstrap.

Exemplos de Componentes Essenciais:

- **Botões:**
 - Primário, Secundário, Sucesso, Erro, Aviso (com estados de hover, active, disabled).
 - Tamanhos: Pequeno, Médio, Grande.
 - Ícones em botões.
- **Campos de Formulário:**
 - Input de Texto, E-mail, Senha (com toggle de visibilidade), Número.
 - Textarea.
 - Checkbox, Radio Button, Toggle Switch.
 - Select/Dropdown.
 - Estados de erro, sucesso, desabilitado.
- **Navegação:**
 - Header (cabeçalho da aplicação com logo, título, ícones).
 - Bottom Tab Navigator (para navegação principal em mobile).
 - Stack Navigator (para navegação entre telas).
 - Breadcrumbs (se aplicável em telas mais complexas).
- **Cards:**
 - Para exibir informações de startups, investimentos, perfis.
 - Com título, descrição, imagem/ícone, botões de ação.
- **Modals/Alerts:**

- Para confirmações, mensagens de erro/sucesso, pop-ups informativos.
- **Listas:**
 - Listas de itens (ex: transações, documentos, notificações).
 - Com ícones, textos primários/secundários.
- **Indicadores:**
 - Spinners de carregamento.
 - Barras de progresso.
- **Avatares:** Para perfis de usuário.
- **Badges/Tags:** Para status ou categorias.

3. Diretrizes de Design e Uso

3.1. Acessibilidade (A11y)

- **Contraste de Cores:** Garantir contraste suficiente entre texto e fundo.
- **Tamanhos de Fonte:** Permitir redimensionamento de texto.
- **Navegação por Teclado:** Assegurar que todos os elementos interativos sejam navegáveis via teclado (para PWA e emuladores).
- **Rótulos Semânticos:** Usar `aria-labels` e `accessibilityLabel` (React Native) para leitores de tela.
- **Estados Visuais:** Indicar claramente estados de foco, hover e ativo.

3.2. Responsividade

- **Layouts Flexíveis:** Utilizar flexbox e grid (no caso de React Native, flexbox é o principal) para criar layouts que se adaptam a diferentes tamanhos de tela.
- **Imagens e Mídia:** Otimizar imagens para mobile e garantir que se ajustem aos contêineres.
- **Pontos de Quebra (Breakpoints):** Definir como os componentes se comportam em diferentes tamanhos de tela (embora em React Native o foco seja mais em layout flexível do que em breakpoints CSS tradicionais).

3.3. Internacionalização (i18n)

- **Textos Traduzíveis:** Todos os textos, rótulos e mensagens devem ser externalizados e gerenciados por um sistema de i18n (ex: `react-i18next`).
- **Layout RTL (Right-to-Left):** Considerar suporte a idiomas RTL se o escopo futuro incluir mercados como o árabe ou hebraico.

- **Formato de Datas e Moedas:** Adaptar para o locale do usuário.

3.4. Branding e Voz da Marca

- **Tom de Voz:** Definir a personalidade da comunicação (formal, amigável, informativo).
- **Linguagem:** Usar terminologia consistente e clara.
- **Elementos Visuais:** Garantir que o logo, cores e ícones sejam usados de forma consistente para reforçar a marca iSelfToken.

4. Implementação e Manutenção do Design System

4.1. Ferramentas

- **Documentação:** Storybook (para documentar e visualizar componentes React Native), ou uma documentação interna baseada em Markdown.
- **Gerenciamento de Pacotes:** Publicar os componentes do Design System como um pacote npm privado para fácil reutilização em diferentes projetos ou módulos.
- **Versionamento:** O Design System terá seu próprio repositório Git e ciclo de versionamento.

4.2. Processo de Contribuição

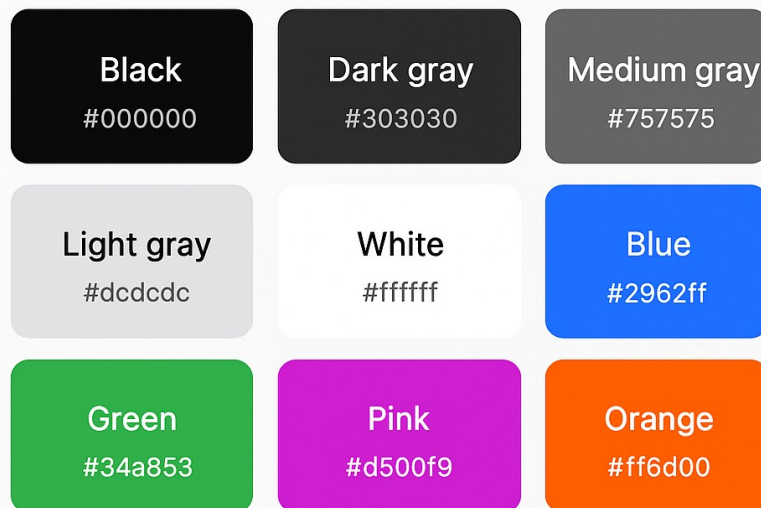
- **Guidelines:** Criar diretrizes claras para a contribuição de novos componentes ou padrões.
- **Revisão:** Todo novo componente ou alteração deve passar por revisão de design e código.
- **Comunicação:** Manter a equipe atualizada sobre as mudanças e novas adições ao Design System.

5. Próximos Passos

1. **Definição Detalhada:** Criar especificações detalhadas para cada componente (estados, variações, propriedades).
2. **Criação da Biblioteca de Componentes:** Iniciar o desenvolvimento dos componentes básicos em React Native, utilizando React Bootstrap como referência de estilo e estrutura.
3. **Documentação:** Começar a documentar os componentes e diretrizes no Storybook ou plataforma similar.
4. **Integração:** Integrar a biblioteca de componentes nos módulos do aplicativo iSelfToken.

Este Design System fornecerá a base para construir uma experiência de usuário excepcional e consistente em toda a plataforma iSelfToken, garantindo que o desenvolvimento seja escalável e eficiente.

CORES



FONTES

Inter

Inter Medium

Inter SemiBold

Inter Bold