

Pet Virtual Interativo: Estrutura Essencial em React Native

Este aplicativo simula um **mascote digital dinâmico** em React Native. Ele reage ao tempo e às interações do usuário, mudando de emoções e necessidades para uma experiência divertida e envolvente.

Ferramentas Principais

O app usa os Hooks essenciais do React e componentes do React Native:

- **useState**: Gerencia informações que mudam, como o estado do pet ('fome', 'feliz') e textos de fala.
 - **useEffect**: Executa lógicas importantes em momentos específicos, como carregar fontes, gerenciar *timers* e exibir falas.
 - **useRef**: Mantém uma referência a valores que não precisam re-renderizar o componente, como IDs de *timers*, para garantir que sejam limpos corretamente.
 - **Componentes RN (View, Text, Image, TouchableOpacity, ImageBackground, StyleSheet)**: São os blocos de construção para a interface visual e interativa do app.
 - **expo-font**: Permite carregar fontes personalizadas para um visual único.
-

O Coração do App: Gerenciamento de Estado

Vários estados controlam o comportamento e a aparência do pet:

- **estadoPet**: Define a emoção/necessidade atual ('normal', 'fome', 'feliz', 'coco').
 - **fala**: O texto temporário no balão de fala do pet.
 - **fonteCarregada**: Confirma que a fonte personalizada foi carregada.
 - **fundoAtual / chapéuAtual**: Índices para controlar qual fundo e chapéu estão ativos nas suas respectivas listas.
 - **quantidadeCoco**: Conta quantos "cocs" estão na tela.
 - **nomePet**: O nome do mascote, "Salt".
-

Dados Essenciais

O aplicativo organiza informações importantes em listas e objetos:

- **fundos**: Lista de imagens para os diferentes planos de fundo.
- **chapeus**: Lista de imagens para os acessórios do pet, incluindo uma opção "sem chapéu".

- **estilosChapeus:** Mapeia estilos específicos para cada chapéu, garantindo o ajuste correto.
 - **imagemPet:** Associa cada estado do pet a uma imagem GIF correspondente (ex: pet chorando quando com fome).
 - **efeitos:** Guarda imagens para efeitos visuais, como corações e os "cocs".
 - **mensagens:** Objeto com falas específicas para cada estado do pet.
 - **TEMPO_PARA_SENTIR_FOME:** Uma constante que define o tempo para o pet sentir fome automaticamente (9 segundos).
-

Lógica Chave do Aplicativo

- **Carregamento de Fonte:** Um `useEffect` carrega a fonte "pixel" uma única vez ao iniciar o app, garantindo que o texto tenha o estilo desejado.
 - **Falas Dinâmicas:** Outro `useEffect` exibe uma fala aleatória no balão do pet por 2 segundos, sempre que o `estadoPet` muda. Um `useRef` garante que o timer da fala seja limpo corretamente.
 - **Fome Automática:** Se o pet está `normal` por 9 segundos, um `useEffect` o muda para o estado `fome`, simulando o tempo.
 - **alimentarPet():** Se o pet está com fome ou "sujo", alimentá-lo o deixa `feliz`. Após 8 segundos, ele "faz cocô", incrementando `quantidadeCoco` e mudando para o estado `coco`.
 - **limparCoco():** Reseta `quantidadeCoco` para zero e retorna o pet ao estado `normal`.
 - **trocarFundo() / trocarChapeu():** Funções que ciclam entre as opções de fundos e chapéus usando o operador de módulo (%), mudando a aparência do cenário e do pet.
-

Como o App é Exibido (Renderização)

A interface é construída dinamicamente:

- **Fundo:** Uma imagem de fundo que muda com a função `trocarFundo()`.
- **Botões de Ação:** Ícones clicáveis para alimentar, limpar, trocar fundo e chapéu. Alguns botões são **desativados condicionalmente** (`disabled`) se a ação não for aplicável (ex: limpar sem cocôs).
- **Balão de Fala:** Aparece e desaparece conforme o pet "fala" (`fala !== ''`).
- **Pet Interativo:** Sua imagem (um GIF) muda conforme o `estadoPet`.
- **Acessórios e Efeitos:** Chapéus e efeitos visuais (corações, cocôs) aparecem ou somem dependendo do estado do pet e das ações do usuário. Os "cocs" são renderizados em lista e posicionados para dar a ideia de acúmulo.