FOUNDATION

IOS

ANA CLAUDIA SANTANA

GUSTAVO NASCIMENTO

BRUNA MORAIS

1 Explicação Geral

Nosso projeto é um aplicativo de Pokedex criado em SwiftUI. Ele permite visualizar todos os pokémons da 1ª geração (151 no total), com nome e tipo.

O app possui duas telas principais:

- Pokedex: exibe uma lista com os nomes e tipos dos pokemons.
- Estatísticas: mostra as características e status de cada Pokemon

2 Trechos Importantes

1
```
struct ContentView: View {
  @State var selectedTab:Int = 0
  @State var selectedPokemon: Pokemon?

  var body: some View {
    TabView {
      PokedexView()
        .tabItem {
          Label("Pokedex", systemImage: "magazine.fill")
        }.tag(0)

      StatisticsView(pokemon: selectedPokemon ??
Pokemon(id:0,name: "dd",types: [ElementType.normal]))
        .tabItem {
          Label("Estatísticas", systemImage: "chart.xyaxis.line")
        }.tag(1)
    }
  }
}
```

2 Trechos Importantes

2

```
var primaryType: ElementType {    pokemon.types.first ?? .normal  }

 var body: some View {    VStack(spacing: 12) {
    AsyncImage(url: URL(string: "https://www.pokemon.com/static-assets/content-assets/cms2/img/pokedex/full/\(String(format: "%03d", pokemon.id)).png")) { image in image
        .resizable()        .scaledToFit()        .frame(width: 120, height: 120)
    } placeholder: {
      ProgressView()        .frame(width: 120, height: 120)
.background(Color.gray.opacity(0.1))       .cornerRadius(60)
    }

    HStack(spacing: 6) {
      ForEach(pokemon.types, id: \.self) { type in
        Text(type.rawValue.capitalized)
.          ...
      }
   .background(
    LinearGradient(
      colors: [primaryType.getLightColor(), Color.white], startPoint: .topLeading,
      endPoint: .bottomTrailing      )    )    .cornerRadius(20)    .shadow(color: primaryType.getColor().opacity(0.3), radius: 8, x: 0, y: 4
}
```

2 Trechos Importantes

3

```
import SwiftUI

struct PokedexView: View {
  @State private var searchText = ""
  @State private var selectedType: ElementType? = nil

  let columns = [
    GridItem(.flexible(), spacing: 16),
    GridItem(.flexible(), spacing: 16)
  ]

  var filteredPokemons: [Pokemon] {
    pokemons.filter { pokemon in
      let matchesSearch = searchText.isEmpty ||
pokemon.name.localizedCaseInsensitiveContains(searchText)
      let matchesType = selectedType == nil || pokemon.types.contains(selectedType!)
      return matchesSearch && matchesType
    }


. ...
```

2 Trechos Importantes

4

```
VStack {
        Text("NOME").font(.caption).foregroundStyle(.secondary)
        Text(pokemon.name.capitalized).font(.largeTitle).bold()
      }.padding(8)
      VStack {
        Text("ID").font(.caption).foregroundStyle(.secondary)
        Text("\(pokemon.id)").font(.largeTitle).fontWeight(.semibold)
      }.padding(8)
      HStack {
        Text("TIPOS").font(.caption).foregroundStyle(.secondary)
        ForEach(pokemon.types) { type in
          ZStack {
            RoundedRectangle(cornerSize: CGSize(width: 20, height:
20)).frame(width:
/*@START_MENU_TOKEN@*/100/*@END_MENU_TOKEN@*/,height: 50)
            Text(type.rawValue).colorInvert()
          }
        }
      }
    }
```

4 Reflexão sobre os desafios

1	Entender a estrutura do SwiftUI e como dividir o app em arquivos separados.
2	Criar uma lista com os pokémons e trabalhar com arrays e enums.
3	Exibir estatísticas com base nos dados.

5 Soluções para os desafios

1	Dividimos os arquivos com base na função de cada um (ex: Pokemon.swift, PokedexView.swift, StatisticsView.swift) para colocar cada parte do código (structs, views, funções e estilos).
2	Usar o componente For Each e o identificador id: \.id para renderizar corretamente para exibir a lista de Pokemons.
3	Atualizamos o Enum, com uma função que retorna a cor de acordo com o caso do enum, usamos String(format: "%03d", pokemon.id) para formatar a URL das imagens, criamos estados para armazenar o pokemon selecionado, uma função que retorna uma some View com quadrados de larguras diferentes para os valores, função para retornar cor clara, para mostrar no gradiente, criamos um estado para salvar o texto de busca para realizar filtragem