INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: ACESSO À REDE – BAIXANDO IMAGENS

Prática 09

Obs.: esta prática é continuação da prática 08; use controle de versões.

Parte 1: Carregando imagens na lista de cidade e nas previsões climáticas

Passo 1: Baixe uma imagem que será usada como *placeholder* enquanto as imagens acima estão carregando.

Salve a imagem em `res/drawable/loading.png` no projeto Android. Exemplo:

Passo 2: Adicione o suporte as bibliotecas Picasso e Coil ao **build.gradle.kts** do app.

```
...
dependencies {
    ...
    implementation(libs.coil.compose) // coil
    implementation(libs.picasso)      // Picasso
    ...
}
```

Essa bibliotecas são responsáveis por carregar e exibir imagens da Internet.

Passo 3: Modifique o `CityItem` em `ListPage` como abaixo:

```
@Composable
fun CityItem( ... ) {
    Row( ... ) {
        // Substitui o Icon(...)
        AsyncImage(
            model = city.weather?.imgUrl,
            modifier = Modifier.size(75.dp),
            error = painterResource(id = R.drawable.loading),
            contentDescription = "Imagem"
        )
        ...
    }
}
```

O componente `AsyncImage` (que substitui o `Icon`) carrega uma imagem da rede de forma assíncrona a partir de uma URL. Em caso de erro na carga, será exibida a imagem local configurada no Passo 1.

Passo 4: Modifique a `HomePage` para usar o `AsyncImage`:

```
@Composable
fun HomePage( ... ) {
    Column {
        Row {
            AsyncImage( // Substitui o Icon
                model = viewModel.city?.weather?.imgUrl,
                modifier = Modifier.size(100.dp),
                error = painterResource(id = R.drawable.loading),
                contentDescription = "Imagem"
            )
            ...
        }
        ...
    }
}
```

Passo 5: Modifique o `ForecastItem` usado na `HomePage`:

```
@Composable
fun ForecastItem( ... ) {
    ...

    Row( ... ) {
        AsyncImage( // Substitui o Icon
            model = forecast.imgUrl,
            modifier = Modifier.size(40.dp),
            error = painterResource(id = R.drawable.loading),
            contentDescription = "Imagem"
        )

        ...
    }
}
```

Passo 6: Rode e teste a aplicação. Faças a correções se necessárias.

Se as imagens estiverem carregando corretamente, faça um novo *commit*.

Parte 2: Carregando imagens das condições climáticas no mapa.

Passo 1: Adicione o método abaixo ao `WeatherService`:

```
fun getBitmap(imgUrl: String, onResponse: (Bitmap?) -> Unit) {
    Picasso.get().load(imgUrl).into(object : com.squareup.picasso.Target {
        override fun onBitmapLoaded(bitmap: Bitmap?,
                                    from: Picasso.LoadedFrom?) {
            onResponse.invoke(bitmap)
        }
        override fun onPrepareLoad(placeHolderDrawable: Drawable?) {}
        override fun onBitmapFailed(e: Exception?, errorDrawable: Drawable?) {
            Log.w("WeatherApp WARNING", "" + e?.message)
            e?.printStackTrace()
        }
    })
}
```

Esse método usa a biblioteca Picasso para carregar a imagem em um objeto do tipo `Bitmap`, que é retornado via *lambda*.

Passo 2: Adicione o método abaixo ao `Repository`:

```
fun loadBitmap(city: City) {
    weatherService.getBitmap(city.weather!!.imgUrl) { bitmap ->
        city.weather!!.bitmap = bitmap
        listener.onCityUpdated(city)
    }
}
```

Passo 3: Modifique a chamada ao `GoogleMap()` em `MapPage`:

```
GoogleMap ( ... ) {
    viewModel.cities.forEach { city ->
        if (city.location != null) {
            var marker = BitmapDescriptorFactory.defaultMarker()

            if (city.weather == null) {
                repo.loadWeather(city)
            } else if (city.weather!!.bitmap == null) {
                repo.loadBitmap(city)
            } else {
                marker = BitmapDescriptorFactory
                          .fromBitmap(city.weather!!.bitmap!!.scale(200, 200))
            }

            Marker(
                state = MarkerState(position = city.location!!),
                icon = marker,
                title = city.name,
                snippet = city.weather?.desc?:"carregando..."
            )
        }
    }
}
```

Nesse código, verificamos se as condições climáticas da cidade foram carregadas. Caso não tenham, lançamos o carregamento via `Repository`. Se já estiverem carregadas, verificamos se o *bitmap* (imagem) associado às condições climáticas foi carregado, e carregamos caso contrário. O *bitmap* é usado como marcador da cidade no mapa.

Passo 4: Rode e teste a aplicação. Faças a correções se necessárias.

Se as imagens estiverem carregando corretamente, faça um novo *commit*.