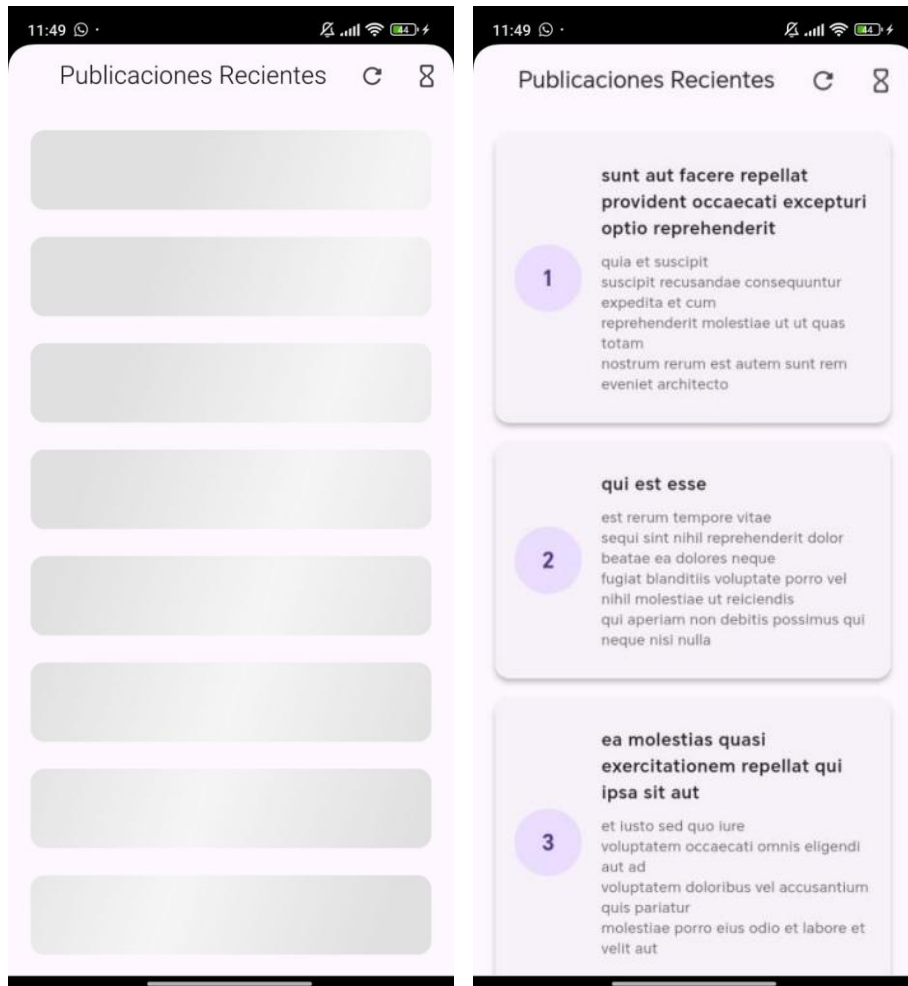
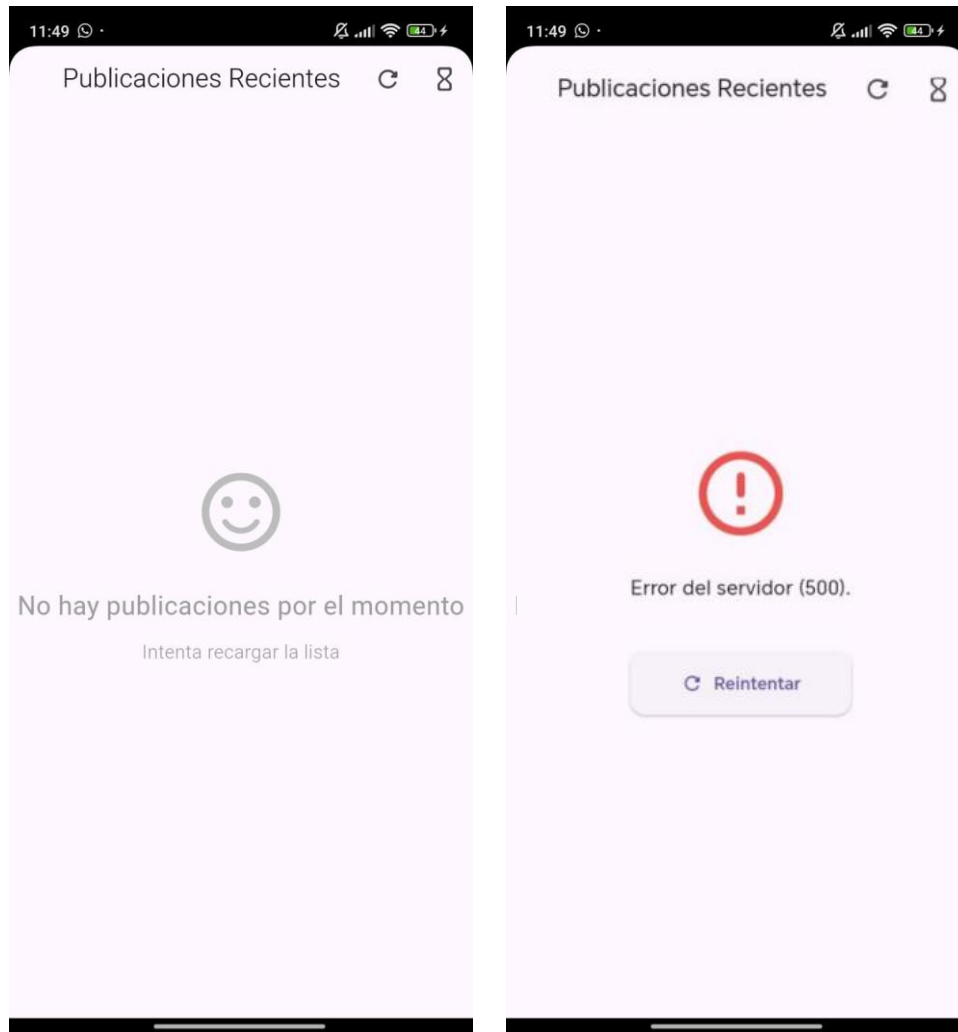


Universidad Internacional del Ecuador

Estudiante: Juan Fuentes

Capturas solicitadas:





Código del interceptor solicitado:

```

post_model.dart  main.dart  post_provider.dart  post_screen.dart  api_se
act213 > lib > services > network_interceptor.dart > NetworkInterceptor > onRequest
4  class NetworkInterceptor extends Interceptor {
6    Future<void> onRequest(
11      final delaySeconds = Random().nextInt(4) + 1;
12      await Future.delayed(Duration(seconds: delaySeconds));
13
14      if (Random().nextDouble() < 0.20) {
15        final isServerError = Random().nextBool();
16        final statusCode = isServerError ? 500 : 401;
17        final message = isServerError
18          ? 'Error interno del servidor (simulado)'
19          : 'No autorizado (simulado)';
20
21        handler.reject(
22          DioException(
23            requestOptions: options,
24            response: Response(
25              data: {'message': message},
26              statusCode: statusCode,
27              requestOptions: options,
28            ), // Response
29            type: DioExceptionType.badResponse,
30          ), // DioException
31        );
32        return;
33      }
34      handler.next(options);
35    }
36  }

```

¿Cómo cambió la percepción de la app al pasar de un círculo de carga a un Shimmer Effect?

En esta práctica, el uso del Shimmer Effect en vez de un CircularProgressIndicator mejora la perspectiva o percepción en cuanto al rendimiento, generando que la espera se sienta un tanto mas corta.

Manejar los estados de Loading, Success, Empty y Error proporciona una experiencia mas adecuada con mensajes claros, iconos y un botón de "Reintentar" garantizando una buena experiencia con redes inestables.