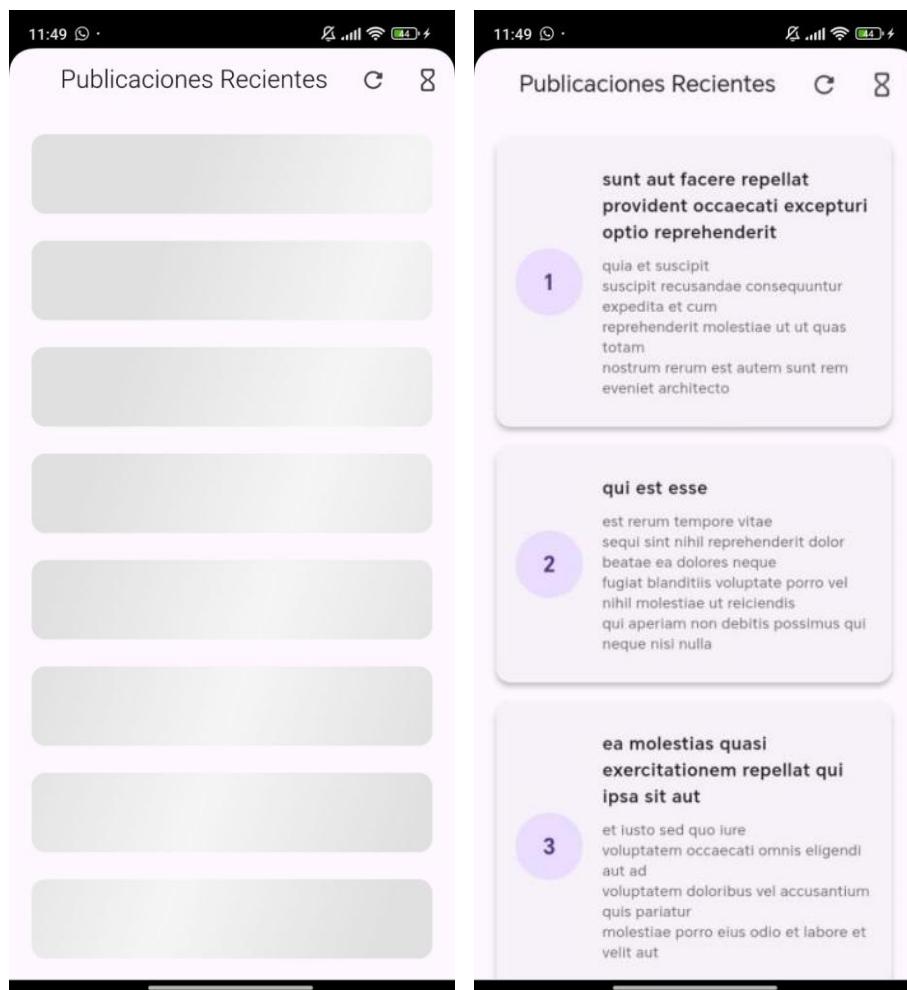
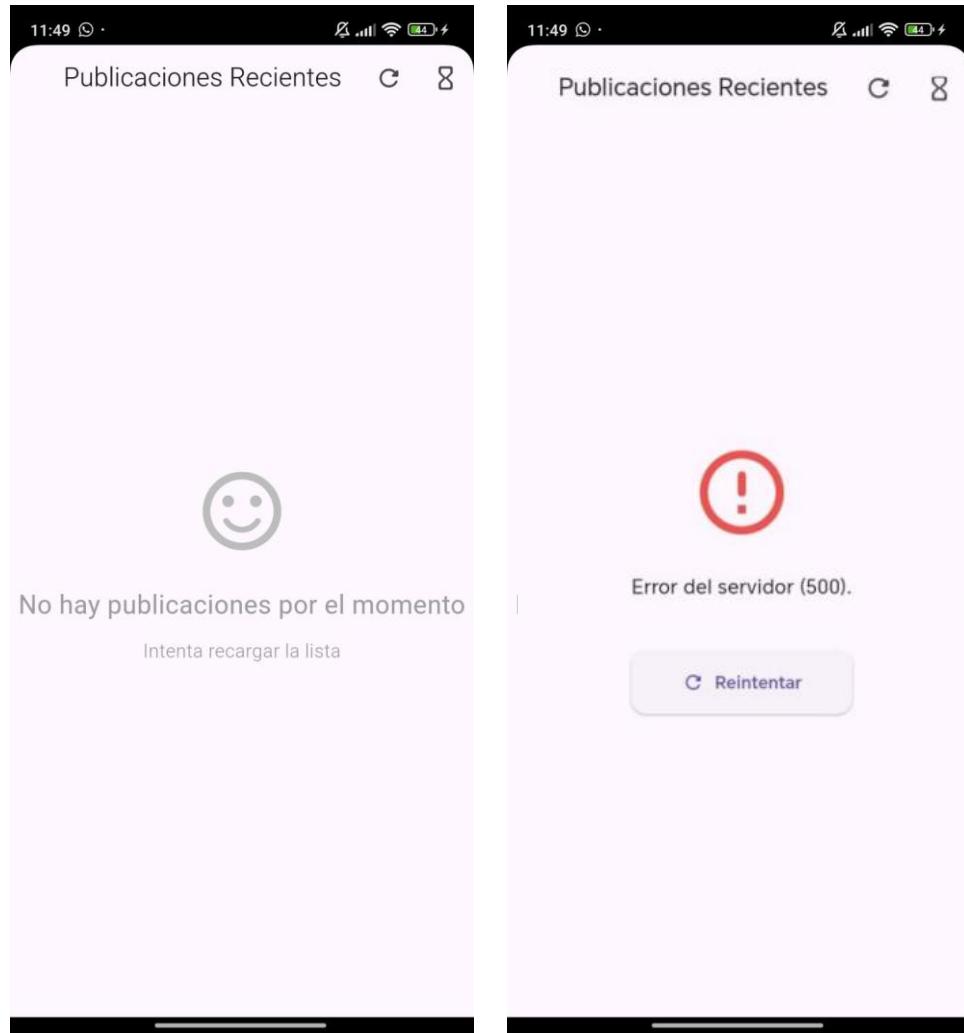


Universidad Internacional del Ecuador

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 {
  5     Future<void> onRequest(
  6       final delaySeconds = Random().nextInt(4) + 1;
  7       await Future.delayed(Duration(seconds: delaySeconds));
  8
  9       if (Random().nextDouble() < 0.20) {
 10         final isServerError = Random().nextBool();
 11         final statusCode = isServerError ? 500 : 401;
 12         final message = isServerError
 13           ? 'Error interno del servidor (simulado)'
 14           : 'No autorizado (simulado)';
 15
 16         handler.reject(
 17           DioException(
 18             requestOptions: options,
 19             response: Response(
 20               data: {'message': message},
 21               statusCode: statusCode,
 22               requestOptions: options,
 23               ), // Response
 24               type: DioExceptionType.badResponse,
 25               ), // DioException
 26             );
 27           return;
 28         }
 29       }
 30     handler.next(options);
 31   }
 32 }
```

¿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.