

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	PLATAFORMAS EMERGENTES (E)				
TÍTULO DE LA PRÁCTICA:	Implementación de componentes nativos				
NÚMERO DE PRÁCTICA:	09	AÑO LECTIVO:	2025 – A	NRO. SEMESTRE:	10
FECHA DE PRESENTACIÓN	09/07/2025	HORA DE PRESENTACIÓN	23:00		
INTEGRANTE (s): <ul style="list-style-type: none"> - Chirinos Concha, Luis Guillermo - Turpo Torres, Gustavo Jonathan 				NOTA:	
DOCENTE(s): <ul style="list-style-type: none"> - Ernesto Mauro Suárez Lopez 					

SOLUCIÓN Y RESULTADOS
<p>I. RUTA SCV GITHUB/GITLAB</p> <p>https://github.com/luisguillermo900/laboratorio9PE.git</p> <p>II. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</p> <p>En la implementación de una aplicación en Flutter (DART), en muchas situaciones es necesario implementar funcionalidades en código nativo Kotlin/Swift. Entonces, investigar las siguientes tecnologías:</p> <ol style="list-style-type: none"> a. <u>MethodChannel</u> b. <u>EventChannel</u> c. <u>BasicMessageChannel</u> d. Pigeon e. FFI <p>Luego selecciona una de ellas e implementa un componente que realice las operaciones básicas de suma, resta, multiplicación y división.</p>

SOLUCIÓN:

- Tecnologías-tabla comparativa

Tecnología	¿Qué es?	Es ideal para	¿Comunicación bidireccional?	¿Requiere configuración compleja?	Uso común con Kotlin
MethodChannel [1]	Canal que permite invocar funciones nativas desde Flutter.	Llamadas nativas puntuales (como suma).	Sí	Media	Muy común
EventChannel [2]	Canal para enviar streams desde nativo hacia Flutter.	Recibir datos continuos (GPS, sensores).	Solo nativo → Flutter	Media	Pocas veces usado para operaciones simples
BasicMessageChannel [3]	Canal genérico para mensajes entre Flutter y nativo (usando Strings).	Comunicación simple tipo mensaje de texto.	Sí	Algo más técnico	Menos usado
Pigeon [4]	Herramienta que genera código automáticamente para la comunicación.	Proyectos grandes con múltiples funciones.	Sí	Alta	Escalable, pero más complejo
FFI (Foreign Function Interface) [5]	Permite llamar código nativo en C/C++ directamente desde Dart.	Código nativo de bajo nivel o bibliotecas C	No aplica a Kotlin	Alta	No sirve para Kotlin (solo C/C++)

Entonces, lo que voy a utilizar es **method channel** como voy a implementar suma, resta, multiplicación y división en kotlin y quiero, luego, llamar desde flutter, es la opción más directa.

La implementación del Methodchannel se hace desde ambos lados: desde Flutter y desde Kotlin (Android nativo), porque es un canal de comunicación entre ambos mundos.

Tecnología	Lenguaje	Comunicación
Flutter	Dart	- Crea el Method channel.- Envía solicitudes a Kotlin (como "sumar", "restar", etc.) y luego se espera una respuesta.
Android	Kotlin	- Se escucha las llamadas del canal.- Ejecuta la función nativa correspondiente (ej. sumar() en Kotlin), luego retorna el resultado a Flutter.

III. SOLUCIÓN DEL CUESTIONARIO

- Si quisieras desarrollar un plugin para acceder a sensores del teléfono ¿Qué tecnología de la sección anterior emplearías? Justificar la respuesta

Si se desea desarrollar un plugin para acceder a los sensores del teléfono la tecnología más adecuada a utilizar es Platform Channels, ya que permite la comunicación entre el código escrito en Flutter y el código nativo del sistema operativo, como Kotlin para Android o Swift para iOS. Esta tecnología es necesaria porque Flutter, por sí solo, no puede interactuar directamente con APIs específicas del dispositivo, como los sensores de movimiento, proximidad o luz. A través de Platform Channels, se pueden enviar mensajes desde Flutter hacia el sistema nativo para solicitar información del sensor y devolver la respuesta. Esta solución está recomendada oficialmente por Flutter para acceder a funcionalidades nativas no disponibles directamente en el framework.

IV. CONCLUSIONES

Para acceder a los sensores del teléfono desde Flutter, la mejor opción es utilizar Platform Channels, ya que permiten conectar el código Flutter con funciones nativas del dispositivo de forma sencilla y eficiente.

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 4</p>

REFERENCIAS Y BIBLIOGRAFÍA

- [1] https://docs.flutter.dev/translate.google/platform-integration/platform-channels?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- [2] <https://api.flutter.dev/flutter/services/EventChannel-class.html>
- [3] <https://api.flutter.dev/flutter/services/BasicMessageChannel-class.html>
- [4] <https://pub.dev/packages/pigeon>
- [5] [https://codelabs.developers.google.com/codelabs/flutter-ffigen?hl=es-419#:~:text=Dart's%20FFI%20\(foreign%20function%20interface,t%20covered%20in%20this%20codelab.](https://codelabs.developers.google.com/codelabs/flutter-ffigen?hl=es-419#:~:text=Dart's%20FFI%20(foreign%20function%20interface,t%20covered%20in%20this%20codelab.)
- [6] <https://docs.flutter.dev/platform-integration/platform-channels>