



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE INGENIERÍA DE SISTEMAS

**DISEÑO DE UNA APLICACIÓN MÓVIL ANDROID, PARA LA CONSULTA DE LA
DISPONIBILIDAD DE ZONAS DE PARQUEO Y EL REGISTRO DEL INGRESO
VEHICULAR DE VISITANTES AL CAMPUS SUR DE LA UNIVERSIDAD POLITÉCNICA
SALESIANA, APLICANDO PROGRAMACIÓN CON PARALELISMO.**

Trabajo de titulación previo a la obtención del título de
Ingenieros de Sistemas

AUTORES: WASHINGTON OSWALDO FALCON AGUAYO
JEFFERSON GONZALO LASCANO PICO

TUTOR: LINA PATRICIA ZAPATA MOLINA

Quito – Ecuador
2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros Washington Oswaldo Falcon Aguayo con documento de identidad N° 1722495510 y Jefferson Gonzalo Lascano Pico con documento de identidad N° 1724033046, manifestamos que:

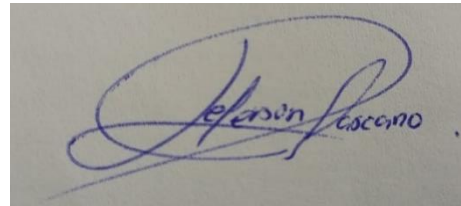
Somos los autores y responsables del presente artículo académico y autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 27 de septiembre del 2022

Atentamente,



Washington Oswaldo Falcon Aguayo
1722495510



Jefferson Gonzalo Lascano Pico
1724033046

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros Washington Oswaldo Falcon Aguayo con documento de identidad N° 1722495510 y Jefferson Gonzalo Lascano Pico con documento de identidad N° 1724033046, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Académico: “Diseño de una aplicación móvil Android, para la consulta de la disponibilidad de zonas de parqueo y el registro del ingreso vehicular de visitantes al Campus Sur de la Universidad Politécnica Salesiana, aplicando programación con paralelismo”, el cual ha sido desarrollado para optar por el título de: Ingenieros de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

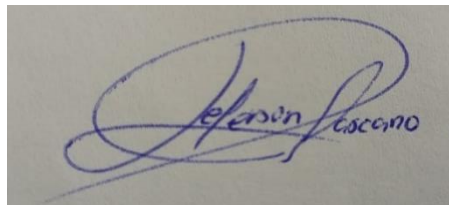
En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 27 de septiembre del 2022

Atentamente,



Washington Oswaldo Falcon Aguayo
1722495510

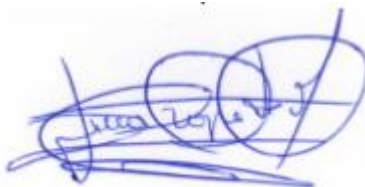


Jefferson Gonzalo Lascano Pico
1724033046

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Lina Patricia Zapata Molina con documento de identificación N° 0501877278 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el artículo académico, con el tema: DISEÑO DE UNA APLICACIÓN MÓVIL ANDROID, PARA LA CONSULTA DE LA DISPONIBILIDAD DE ZONAS DE PARQUEO Y EL REGISTRO DEL INGRESO VEHICULAR DE VISITANTES AL CAMPUS SUR DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, APLICANDO PROGRAMACIÓN CON PARALELISMO, realizado por Washington Oswaldo Falcon Aguayo con documento de identificación N° 1722495510 y Jefferson Gonzalo Lascano Pico, y N° 1724033046, obteniendo como resultado final el trabajo de titulación bajo la opción a Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 27 de septiembre del 2022



.....
Ing. Lina Patricia Zapata Molina, PH. D

0501877278

DISEÑO DE UNA APLICACIÓN MÓVIL PARA ANDROID UTILIZANDO PARALELISMO, PARA LA CONSULTA DE LA DISPONIBILIDAD DE ZONAS DE PARQUEO Y EL REGISTRO DEL INGRESO VEHICULAR DE VISITANTES AL CAMPUS SUR DE LA UNIVERSIDAD POLITÉCNICA SALESIANA

DESIGN OF A MOBILE APPLICATION FOR ANDROID USING PARALLELISM, FOR THE CONSULTATION OF THE AVAILABILITY OF PARKING ZONES AND THE REGISTRATION OF VEHICULAR ENTRY OF VISITORS TO THE SOUTH CAMPUS OF THE SALESIAN UNIVERSITY OF POLYTECHNICS

Washington Oswaldo Falcon Aguayo¹, Jefferson Gonzalo Lascano Pico², Lina Patricia Zapata Molina³

Resumen

El presente artículo académico consiste en el diseño de una aplicación móvil mediante el marco de referencia Flutter usando lenguaje de código abierto Dart, un estilo de arquitectura de transferencia de estado representacional API REST que permite a las aplicaciones acceder a un recurso de nuestra aplicación o servicio a través de una petición HTTP para la manipulación de las solicitudes: GET devuelve la consulta de espacios disponibles de parqueaderos y POST el registro de los visitantes al campus de la Universidad Politécnica Salesiana Campus Sur en formato JSON. Se emplea el método de paralelismo para que las tareas se ejecuten simultáneamente ya que con eso se espera reducir el tiempo de respuesta a 0.10 minutos.

Palabras Clave: Android, paralelismo, aplicación, react native, lenguaje Dart, Isolate.

Abstract

This academic article consists of the design of a mobile application using the Flutter reference framework using the open-source language Dart, an API REST representational state transfer architecture style that allows the application to access a resource of our application or service. through an HTTP request for the handling of requests: GET returns the query of available parking spaces and POST the registration of visitors to the campus of the Salesian Polytechnic University South Campus in JSON format. The parallelism method is used so that the tasks are executed simultaneously, since this is expected to reduce the response time to 0.10 minutes.

Keywords: Android, parallelism, application, react native, Dart language, Isolate.

¹ Estudiante de Ingeniería de Sistemas – Universidad Politécnica Salesiana, Egresado – UPS – sede Quito. Autor para correspondencia: jlascanop@est.ups.edu.ec

² Estudiante de Ingeniería de Sistemas – Universidad Politécnica Salesiana, Egresado – UPS – sede Quito. Autor para correspondencia: jlascanop@est.ups.edu.ec

1. Introducción

En la actualidad la programación en paralelo ha tenido un crecimiento favorable en el entorno de las aplicaciones móviles para su eficiente funcionamiento cumpliendo con procesos que tienen múltiples tareas de funcionamiento, asimismo desde el punto de vista tecnológico podemos decir que los dispositivos Android y IOS han tomado mucha importancia al momento de efectuar acciones o tareas respecto a el ámbito de tener información amplia y precisa en tiempos reales [1].

Se puede fomentar la programación en paralelo ya que dispone de lenguajes clásicos y funciones específicas que ayudan a la ejecución correcta de procesos. El lenguaje paralelo se utiliza para fomentar soluciones más rápidas y efectivas utilizando todos los recursos que puede disponer una maquina y si no es suficiente adaptando más recursos a la misma para su correcto funcionamiento. Como términos precisos los algoritmos en paralelo se lo utilizan para disminuir tiempos tomando en cuenta la distribución que deben obtener cada una de sus tareas de procesamiento [2].

La mayoría de las instituciones tanto públicas como privadas han buscado el optimizar las cogestiones de sus áreas de trabajo según C. Edward ha resaltado que el porcentaje de tráfico de vehículos es del 30% y todo es ocasionado por que no encuentran lugares disponibles de estacionamiento que estén vacíos y sean seguros [3].

Tomando en cuenta las palabras de Javier Gil Arenales agente para el estudio de instituciones Smarter Cities en la actualidad la tecnología ayuda mucho al abastecimiento de información como a descongestionar espacios muy concurridos brindando información a los usuarios de forma breve y precisa y con esto fomentando el desarrollo en ciudades e instituciones [4].

Palma, Garrido, Sánchez y Quesada, resalta que un algoritmo en paralelo es concurrente estructurado o

diseñado para que funcione con un sistema multiprocesador. Además, que los dispositivos tecnológicos tienen varios núcleos, es decir que el sistema operativo que manejan los móviles tiene una ejecución secuencial casi igual a los procesadores que tienen un solo núcleo en sus estructuras operativas [5].

La problemática abordada por este artículo está en que en la actualidad existen pocas referencias de aplicaciones utilizando la programación con paralelismo para reconocimiento de espacios de parqueaderos públicos o privados por lo cual se estructuro una aplicación Android y IOS que permita evaluar un sistema de reconocimiento de espacios disponibles de parqueaderos logrando terminar las falencias en los parqueaderos de los usuarios de la Universidad Politécnica Salesiana Campus Sur reduciendo el tiempo de búsqueda de manera física por todo el campus, por lo cual el sistema le indicara en que sección (A,B,C) se encuentra disponible un parqueadero.

1.1. Trabajos relacionados

Para el siguiente artículo científico se indagaron varios libros e informes con temas similares al planteado, para conseguir información solida con temas relacionados a la investigación que se está laborando y desde esa perspectiva poder analizar de manera más clara los enfoques que puede tomar la elaboración de la investigación tomando en cuenta una guía de los estudios realizados y así ver las implementaciones de estacionamientos desarrollados por la tecnología a nivel nacional e internacional, desde estudios realizados con anterioridad de tesis, artículos, proyectos y material de estudio de estacionamientos inteligentes [6].

En su trabajo de titulación “Análisis del diseño y desarrollo para el funcionamiento de un sistema Web Online para Ciudad de Guayaquil Parqueo Seguro” Jonathan Grimaldo desarrollo un sistema web online para dar a los usuarios la información de parqueaderos públicos disponibles en el sector de Guayaquil tomado como nombre “Parqueo Seguro”, dando solución tecnológica o informática

a las personas que transitan por dicho sector al momento de buscar un lugar de estacionamiento para colocar sus vehículos de manera segura de manera online y así convirtiendo este sector en un aporte de desarrollo tecnológico para la ciudad de Guayaquil [7].

En Madrid en el año 2013 se realizó una investigación de “Sistemas Inteligentes de Estacionamiento” quien fue el responsable David Herrador quien desarrollo una aplicación inteligente utilizando las herramientas, Visual Basic y Sensores de Ultrasonidos para su aporte de titulación en donde como conclusión se demostró que es posible minimizar el tiempo de búsqueda de estacionamiento en parqueos públicos o privado [8].

Márquez Rosero Marcelo David de la Universidad de las Fuerzas Armadas del Ecuador (ESPE) el cual desarrollo un proyecto de un prototipo de sistema inteligente, que es una de las tantas aplicaciones que se puede implementar en el país para su desarrollo. ZigBee resulto una solución perfecta para recolectar los datos que se generan al monitoreo las plazas de parqueo, ya que al ser una tecnología única para su uso en redes de sensores inalámbricos tiene una tasa de transmisión baja, suficiente para la trama que maneja el estándar. Logrando objetivos claros podríamos resaltar que la programación en paralelo utilizada en aplicaciones móviles nos ayudase a efectivizar la fluidez de tráfico para los usuarios y para los espacios de estacionamientos analizando un sistema de control inteligente [9].

A partir del año 2019 se han venido implementando aplicaciones móviles inteligentes para la administración de Estacionamientos Municipales de Quito lo cual beneficio a varios usuarios para encontrar parqueaderos existentes a nivel provincial. Además, los usuarios cuentan con información de horarios de operación y tarifas por hora fracción. El nuevo sistema de gestión de los estacionamientos municipales permite una mayor administración, a través del control de registro contable, esto a permitido el mayor control de los

vehículos municipales. Quito ha tenido múltiples cambios en su manera de manejar los sistemas a nivel provincial lo cual ayuda al desarrollo y a las actualizaciones tecnológicas en el país y en la provincia sacando beneficios de aplicaciones móviles para el manejo efectivo de automóviles en instituciones públicas y privadas a nivel local [10]. Los estudios ya antes mencionados de proyectos inteligentes e investigaciones se utilizaron protocolos (IEEE 802.11, TCP/IP, ZigBee), los cuales son utilizados para las conexiones y eficacia de los aplicativos o sistemas online según el entorno donde fueron implementados.

2. Materiales y métodos

La presente averiguación sugiere el diseño de 2 aplicaciones móvil:

El aplicativo 1 para uso de alumnos y personal profesores de la universidad politécnica salesiana visualizará los espacios de parqueaderos accesibles y no accesibles de los sectores A, B, y C.

El aplicativo 2 de uso único del personal de guardianía de la universidad politécnica salesiana el cual va a tener un formulario de registro de los visitantes, el cual llevará la Información personal del visitante, hora de ingreso y datos del transporte. Los componentes de software utilizados se detallan a continuación:

1. Lenguaje de programación Flutter
2. Gestor de Base de Datos MySQL Aplicativo Control Parqueadero Aplicativo Control Vista
3. Servicios Rest

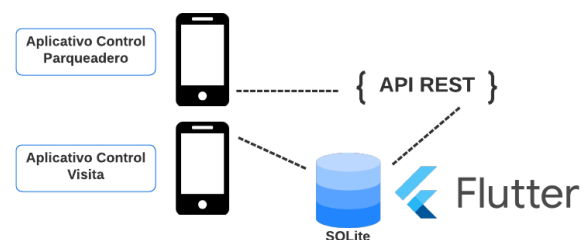


Figura 1. Proceso del funcionamiento de la aplicación móvil con el uso de servicios Rest.

2.1 Herramientas y Arquitecturas de la aplicación móvil

Flutter

El aplicativo se desarrolla en Flutter (DART), ya que es multiplataforma y es un SDK de Google creado para diseñar interfaces nativas iOS y Android. Flutter tiene un modelo de objeto unificado y consistente que se lo conoce como Widget. Los Widget son los elementos básicos de la interfaz de usuario de una aplicación Flutter. Cada Widget es una declaración inmutable de parte de la interfaz de usuario, como se puede observar en la Figura 1 [11].

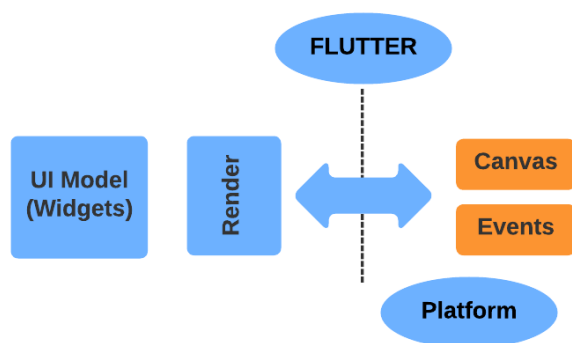


Figura 2. Arquitectura Flutter.

Isolate

Se empleó el paralelismo por datos por medio de la herramienta Isolate, cuya estructura se basa en cada Isolate tiene su propio montón de memoria, lo que garantiza que ninguno de los estados de un Isolate sea accesible desde cualquier otro Isolate. Debido a que no hay memoria compartida, no tiene que preocuparse por mutexes o bloqueos. Al usar Isolate, se puede realizar múltiples tareas independientes a la vez, usando núcleos de procesador adicionales si están disponibles [12].

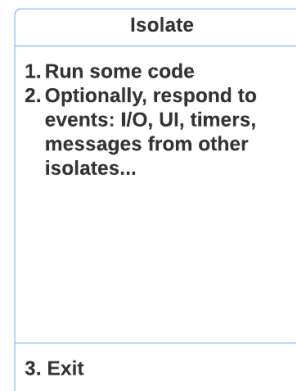


Figura 3. Ciclo de vida Isolate.

Los Isolate son como subprocesos o procesos, sin embargo, cada Isolate tiene su propia memoria y un solo subproceso que hace un periodo de eventos, como se puede mirar en la Figura 2. En la Figura 3 se detalla el proceso del código resultante Isolate.

```

Proceso Parqueadero
Lista global de parqueadero
Inicia el objeto de lista global
Build:
  construye primer bloque de parqueadero
  construye fila de parqueadero
Finaliza
Recorremos la lista global y cambiamos el estado
  Estado verde = disponible
  Estado rojo = ocupado

Petición http
  true: lista parqueaderos actualizada
  false: error al cambiar el estado de la lista global
Finaliza petición http
Invoca metodo de isolate
  Retorna lista global de parqueaderos vacios Estado verde = disponible
Finaliza metodo isolate
FinProceso
  
```

Figura 4. Código resultante Isolate.

A continuación, se presenta el pseudocódigo el cual proporciona el tiempo de respuesta a usar en cada una de las arquitecturas propuestas; paralelo y secuencial.


```

Proceso Medicion de Tiempo
Inicio
entero i,n
variable suma
n=1000000000
i=1
suma=0
Para i=1 hasta n hacer
    suma=suma+1
Fin_para

Escribir(Suma)
Fin
FinProceso

```

Figura 5. Método medición de tiempo.

2.2 Arquitectura Secuencial

En el desarrollo del del aplicativo móvil usando la arquitectura secuencial como se observa en la Figura 5; buscara los espacios vacíos de los parqueaderos por secciones A, B y C.

Paso 1

- Para saber si los espacios permanecen accesibles u ocupados, va a hacer una consulta en los datos de la base de datos parqueaderos UPS e indicará como consecuencia la información de los parqueaderos vacíos.

Paso 2

- Se va a tener los datos almacenados en nuestro servidor para desplegar la información.

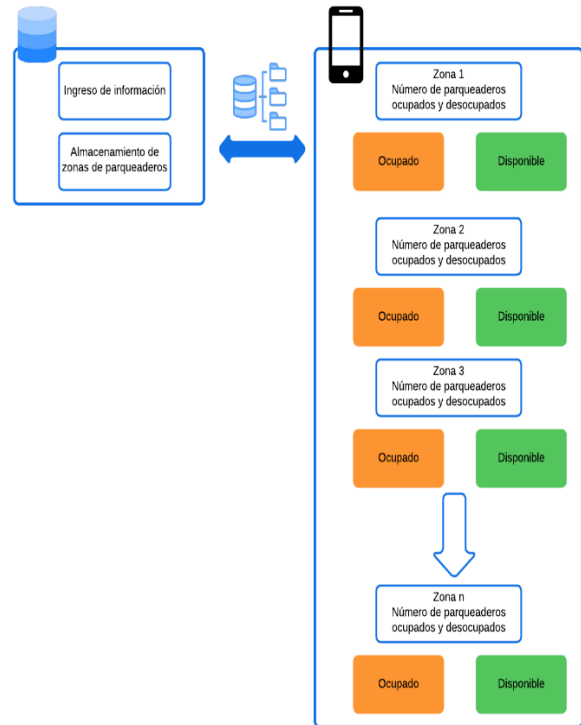


Figura 6. Funcionamiento de la Arquitectura en Secuencial en la búsqueda de espacios vacíos.

2.3 Arquitectura en Paralelo

La arquitectura en Paralelo se apoya en que las normas del aplicativo móvil se ejecutan paralelamente, por lo que desea mencionar que terminan cada una de las normas simultáneamente. En el aplicativo móvil la arquitectura en paralelo como se observa en la Figura 6, buscara paralelamente los espacios vacíos de los parqueaderos con la utilización del Isolate como se observa en la Figura 2, de todos los sectores A, B y C.

Con el método Isolate primero se recupera la lista de los espacios vacíos, luego se almacena en la cache para posteriormente desplegar los espacios vacíos, una vez realizado este proceso se procede a cerrar la vista para dejar de consumir recursos.

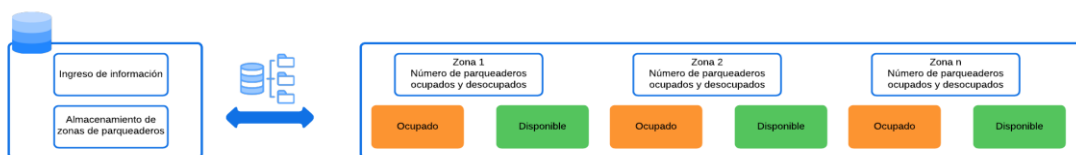


Figura 7. Funcionamiento de la Arquitectura en Paralelo en la búsqueda de espacios vacío.

3. Resultados y discusión

Para eso se usó 3 dispositivos móviles: XIAOMI REDMI NOTE 9, XIAOMI REDMI 9, SAMSUNG GALAXY A10S cuyas propiedades se describen en la **Error! Reference source not found.**

Tabla 1 Dispositivos móviles empleados.

Gama teléfono	Alta	Media	Baja
Modelo	XIAOMI REDMI NOTE 9	XIAOMI REDMI 9	SAMSUNG GALAXY A10S
Android	12 MIUI 11	10 MIUI 11	9.0 Pie
Procesador	MediaTek Helio G85	Helio G80	MediaTek Helio P22
Memoria RAM	4GB	2GB	2GB
Almacenamiento	128GB	64GB	32GB

Prueba 1: Tiempo de respuesta

Para cada dispositivo móvil, el tiempo de respuesta de ejecución se midió en milisegundos (milisegundos). Esto tiene en cuenta 12 de 50 muestras en relación con el tiempo que tarda el método Isolate y muestra el tiempo necesario para dejar espacio disponible. Obtenga los valores que se muestran en la Tabla 2 para cada estacionamiento en el Campus Sur de UPS, como se muestra en la Figura 4. Esta es una prueba realizada con el dispositivo y muestra el mejor promedio de todos los tiempos tomados en cada región. Promedio de los resultados de la ecuación $media(x) = \frac{\sum_{i=1}^N X_i}{N}$.

Según el comportamiento gráfico presentado en la **Error! Reference source not found.**, tenemos la posibilidad de mirar que la época de contestación tanto en paralelo y en secuencial, tienden hacer parecido una vez que es usada la aplicación móvil en un equipo de gama baja llegando a tener una diferencial 0,038 ms.

Tabla 2 Resultado en segundos de tiempos de ejecución por dispositivo.

Gama	Paralelismo		Paralelismo		Paralelismo	
	Tiempo de Respuesta (ms)	Tiempo de Respuesta 2(ms)	Tiempo de Respuesta 3(ms)	Tiempo de Respuesta 4(ms)	Media (ms)	
Alta	0,09	0,12	0,10	0,09	0,10	
Media	0,19	0,23	0,18	0,21	0,20	
Baja	0,41	0,45	0,38	0,38	0,40	
Gama	Secuencial		Secuencial		Secuencial	
	Tiempo de Respuesta (ms)	Tiempo de Respuesta 2(ms)	Tiempo de Respuesta 3(ms)	Tiempo de Respuesta 4(ms)	Media (ms)	
Alta	0,12	0,13	0,11	0,11	0,1175	
Media	0,25	0,27	0,25	0,24	0,2525	
Baja	0,42	0,46	0,44	0,43	0,438	

Según el comportamiento gráfico presentado en la Tabla 2 tenemos la posibilidad de mirar que la época de contestación tanto en paralelo y en secuencial, tienden hacer parecido una vez que es usada la aplicación móvil en un equipo de gama baja llegando a tener una diferencial 0,038 ms.

Prueba 2: Consumo de recursos

El consumo de memoria de cada dispositivo móvil se midió en MB (megabits), considerando únicamente el uso de aplicaciones. Esto se debe a que se usó Geekbench 3 porque otras aplicaciones se ejecutan en la aplicación móvil en segundo plano. Como se muestra en **Error! Reference source not found.** el tiempo de respuesta del procedimiento y el porcentaje de

memoria consumida por cada dispositivo móvil de donde se extraen los datos.

El comportamiento gráfico que se muestra en la Tabla 3 muestra que, al utilizar aplicaciones móviles con dispositivos de gama alta, el consumo de memoria tiende a ser similar tanto para el método paralelo como secuencial, alcanzando una diferencia del 10% de incremento. Esto significa que cuanto mayor sea el recurso, mayor será el consumo de memoria en el método paralelo.

Tabla 3 Consumo de Memoria

Gama teléfono	Alta	Media	Baja
Secuencial	1,87 MB	1,65 MB	1,23 MB
Porcentaje de consumo de memoria por minuto	0,0011%	0,0010%	0,0010%
Paralelo	2,21 MB	1,73 MB	1,41 MB
Porcentaje de consumo de memoria por minuto	0,0020%	0,0011%	0,0010%

4. Conclusiones

El uso de aplicaciones móviles cumple esa función. Al mismo tiempo, pudimos comprobar su utilidad en cuanto a la provisión de estacionamientos vacíos en paralelo, por lo que pudimos responder rápidamente en cuanto a brindar información en tiempo real.

Todo el código de Dart se ejecuta por separado. Cada aislado de dardo tiene un solo hilo de ejecución y no comparte objetos mutables con otros aislados. Los cálculos asincrónicos suelen ser el resultado de cálculos fuera del código Dart real. Esto incluye cálculos que no se completan de inmediato, o si no desea bloquear su código de dardos mientras espera el resultado. El código asincrónico ayuda a que la aplicación continúe respondiendo a las solicitudes HTTP, pero necesita

actualizar el estado de la vista cuando se completa la solicitud HTTP.

La función `CircularProgressIndicator()` contiene código que crea e inicia un aislamiento de trabajo en segundo plano y devuelve un resultado. Antes de que se genere el método aislado, se crea un `ReceivedPort` que permite que el trabajador aislado envíe y envíe mensajes al trabajador principal. Luego proceda a llamar al método `Isolate.spawn()`. Este método crea un aislamiento de trabajo en segundo plano. Aquí, el primer parámetro del aislamiento es la función principal y el segundo parámetro es el puerto de escucha utilizado para enviar el mensaje al trabajador principal.

Flutter actualmente no tiene una solución de paralelismo de memoria compartida, pero admite `Isolate`, que se puede ejecutar en paralelo como subprocesos separados. La función de aislamiento se comunica enviando y recibiendo mensajes asincrónicos.

Referencias

- [1] U. L. Institute, «The Dimensions of Parking,» 2009.
- [2] L. Franco Cordero, «LA MOVILIDAD SOSTENIBLE EN CAMPUS UNIVERSITARIOS: UNA COMPARACIÓN DE LAS MEJORES PRÁCTICAS EN ESTADOS UNIDOS Y EUROPA. APLICABILIDAD EN UNIVERSIDADES VENEZOLANAS / Sustainable Mobility in University Campus: A Comparison Between Best Practices in United S,» n° 23-40, 2014.
- [3] A. Bull y I. Thomson, «La congestión del tránsito urbano: causas y consecuencias económicas y sociales,» 2022. [En línea]. Available: <https://repositorio.cepal.org/handle/11362/10804>.
- [4] M. Navarro, «Smart cities,» 2022. [En línea]. Available: <https://revistabyte.es/tema-de-portada-byte-ti/smart-cities/>.
- [5] L. Vázquez, A. Valdez, G. Campos, R. Campos y R. Hernández, «Programación paralela en una técnica de optimización de tiempos de producción,» ecorfan.org, 2022. [En línea]. Available: <https://www.ecorfan.org/handbooks/Ciencias%20de%20la%20Ingenieria%20y%20Tecnologia%20T-VII/ARTICULO%207.pdf>.
- [6] D. S. Sinaluisa Buñay , «PLATAFORMA DE ESTACIONAMIENTO INTELIGENTE CON SISTEMA DE INFORMACIÓN EN TIEMPO REAL USANDO APLICACIÓN MÓVIL PARA SHOPPING CENTER DE QUEVEDO,» 2016. [En línea]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/20571/1/TRABAJO%20DE%20TITULACION%20DAVID%20SINALUISA.pdf>.
- [7] J. G. Tasigchana Villegas, «ESTUDIO, ANÁLISIS, DISEÑO Y PUESTA EN FUNCIONAMIENTO DE UN SISTEMA WEB ONLINE PARA INFORMAR LA DISPONIBILIDAD DE PARQUEOS PÚBLICOS SEL SECTOR COMERCIAL Y BANCARIO DE LA CIUDAD DE GUAYAQUIL. "PARQUEO SEGURO",» 2014. [En línea]. Available: <http://repositorio.ucsg.edu.ec/bitstream/3317/1241/1/T-UCSG-PRE-ING-CIS-66.pdf>.
- [8] A. Bull y I. Thomson, «La congestión del tránsito urbano: causas y consecuencias económicas y sociales,» 2022. [En línea]. Available: <https://repositorio.cepal.org/handle/11362/10804>.
- [9] R. Márquez y D. Marcelo, «Desarrollo de un prototipo de parqueadero inteligente empleando redes de sensores inalámbricos,» *Universidad de las Fuerzas Armadas*, 2015.
- [10] Quito Informa, «Aplicación móvil para el Sistema de Estacionamientos Municipales,» 2019. [En línea]. Available: <http://www.quitoinforma.gob.ec/2019/08/27/aplicacion-movil-para-el-sistema-de-estacionamientos-municipales/>.
- [11] IONOS Digital Guide, «Flutter: introducción al framework multiplataforma,» 2022. [En línea].

Available:

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-flutter/>.

- [12] Flutter, «Concurrency in Dart,» 2022. [En línea]. Available: <https://dart.dev/guides/language/concurrency>.