

# Encuestador y Notificador: aplicaciones móviles para la captura de datos en campo sobre propiedades georeferenciadas

## *Encuestador and Notificador: mobile applications for field data capture on georeferenced properties*

Joanna Campbell Amos<sup>1\*</sup>, Aracelys García Armenteros<sup>2</sup>

**Resumen** En este trabajo se presenta una propuesta de solución para desarrollar dos aplicaciones móviles cartográficas, que permitan recoger información asociada a propiedades geolocalizadas a partir de formularios y notificar que la encuesta a dichas propiedades fue realizada. Como parte de la investigación se analizan aplicaciones con funcionalidades similares, y se realiza un estudio de plataformas de desarrollo disponibles para el manejo de mapas sin conexión teniendo en cuenta sus ventajas y desventajas. En el trabajo se muestran elementos del diseño y la implementación de la propuesta de solución basado en requerimientos aportados por los clientes y realizados a partir de una cartografía sin conexión e información inicial de la propiedad proporcionada por un servicio externo. Se presentan las aplicaciones móviles Encuestador y Notificador, con el propósito de recoger la información de las propiedades y certificar que las encuestas a las propiedades se realizaron, respectivamente. La validación del funcionamiento de ambas aplicaciones demuestra que cumplen satisfactoriamente con los requerimientos planteados.

**Palabras Clave:** desarrollo móvil, formularios, levantamiento en campo, mapa sin conexión, Sistema de Información Geográfica

**Abstract** *This paper presents a proposed solution for developing two mapping mobile applications that collect information associated with geolocable properties from forms and notify the completion of a survey. As part of the research, applications with similar functionalities are analyzed, and a study is conducted of available development platforms for managing offline maps, taking into account their advantages and disadvantages. The paper presents elements of the design and implementation of the proposed solution based on requirements provided by clients and developed from offline maps and initial property information provided by an external service. The result is the presentation of the mobile applications Encuestador and Notificador, which are responsible for collecting property information and certifying that the property surveys were completed, respectively. Validation of the operation of both applications demonstrates that they satisfactorily meet the stated requirements.*

**Keywords:** mobile development, forms, field data capture, offline map, Geographic Information System.

**Mathematics Subject Classification:** 68-04, 68N19, 68N30, 90-4, 90B99.

<sup>1</sup>Departamento de Programación y Sistemas de Información, Facultad de Matemática y Computación, Universidad de La Habana, La Habana, Cuba. Email: [joanna@matcom.uh.cu](mailto:joanna@matcom.uh.cu)

<sup>2</sup>Departamento de Programación y Sistemas de Información, Facultad de Matemática y Computación, Universidad de La Habana, La Habana, Cuba. Email: [aracelys@matcom.uh.cu](mailto:aracelys@matcom.uh.cu)

\*Autor para Correspondencia (*Corresponding Author*)

**Citar como:** Campbell Amos, J., García Armenteros, A. (2025). Encuestador y Notificador: aplicaciones móviles para la captura de datos en campo sobre propiedades georeferenciadas. *Ciencias Matemáticas*, 39(2), 23-32. DOI: <https://doi.org/XXXXXXXX>. Recuperado a partir de <https://revistas.uh.cu/rcm/article/view/11545>.

### Introducción

La llegada de la revolución digital, y la aparición más tarde de las impresoras, los escáneres y el procesamiento digital de imágenes, facilitaron la expansión en la producción, el acceso

y la difusión de los mapas. Partiendo de esto, el desarrollo de la industria del software que se produjo a finales del siglo pasado sembró la semilla de la cual brotaron herramientas computacionales dedicadas al manejo de la información geo-

gráfica, los Sistemas de Información Geográfica o SIG (GIS, por sus siglas en inglés, *Geographic Information System*) [2].

Un SIG [2] es un software que integra y relaciona diversos componentes que permiten la organización, almacenamiento, manipulación y análisis de grandes cantidades de datos que están vinculados a una referencia espacial, facilitando la incorporación de aspectos socio-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más eficaz.

El surgimiento y desarrollo del internet y de las tecnologías móviles permitió la globalización de las aplicaciones cartográficas potenciando así la explotación de la información geográfica. Tener acceso a estas herramientas resulta de gran utilidad para resolver diversos tipos de problemáticas. La gran variedad de información que pueden proporcionar y manejar estas herramientas puede ser utilizada para investigaciones científicas, la gestión de los recursos, la gestión de la arqueología, la evaluación del impacto ambiental, la cartografía, por nombrar unos pocos de los tantos campos en los que pueden resultar de gran impacto estas herramientas.

Una de las variantes de aplicaciones cartográficas es la destinadas a la recolección de información de sitios de interés a partir de formularios. El uso fundamental de este tipo de aplicaciones reside en el trabajo de campo, donde el trabajador debe transportarse hacia un lugar que no es su lugar habitual o fuera de su oficina. Esto generalmente se aplica en el ámbito de la ingeniería, arquitectura o en el ámbito de las ciencias sociales. Las aplicaciones cartográficas destinadas a este tipo de trabajo resultan de mucha utilidad ya que facilitan el trabajo y brindan una portabilidad y seguridad de la información para su traslado hasta el lugar de procesamiento. Existen múltiples escenarios donde se pudieran aplicar, desde una empresa de electricidad que necesite manejar información necesaria sobre las torres eléctricas de una determinada ciudad para saber si es necesario dar mantenimiento, una empresa de suministro de agua para focalizar las roturas planificando el arreglo según la gravedad, o en empresas encargadas del saneamiento de la ciudad para reportar la situación de los vertederos y contenedores de basura, así como, para la recaudación de impuestos, fiscalización y cuestionarios de los gobiernos provinciales. Como se puede apreciar, este tipo de herramientas tienen gran importancia e impacto en el desarrollo social y otras esferas por lo que resulta importante su estudio. Este trabajo dedica su atención a la implementación de aplicaciones para dispositivos móviles que permiten guardar información a partir de formularios de lugares georeferenciados, finalizando con el desarrollo de dos aplicaciones de este tipo.

La recogida de información sobre las propiedades en la alcaldía de Managua, Nicaragua, generalmente se realiza de manera manual, mediante encuestas impresas en papel. Estas encuestas no son más que formularios sobre una propiedad donde se introduce información como datos generales y datos físicos. La importancia de estas encuestas radica en que con ellas se logran obtener los datos necesarios para computar el valor de las propiedades. Este proceso de encuestas se realiza

anualmente y constituye la base para que la alcaldía recaude los impuestos sobre las propiedades<sup>1</sup>.

Además del proceso descrito anteriormente, un inspector debe trasladarse a la propiedad para certificar que realmente la encuesta fue realizada. Es por ello que la alcaldía precisa una solución para automatizar y agilizar los procesos de encuestar las propiedades y notificar que la encuesta se realizó. Por lo tanto, podemos enunciar nuestro problema de investigación como la demora en los procesos encuestar las propiedades y notificar la realización de la encuesta debido a la carencia de una solución automatizada en correspondencia a los datos y términos requeridos por la alcaldía.

Mediante el convenio Cadica<sup>2</sup>-UH que se desarrolla entre la Alcaldía de Managua y la Casa del Software en la Facultad de Matemática y Computación, Universidad de La Habana, se presentó la tarea de desarrollar las aplicaciones móviles denominadas Encuestador y Notificador para su uso en la Alcaldía de Managua. Estas aplicaciones deben de cumplir con los siguientes requerimientos funcionales, no funcionales y de entorno suministrados por los clientes:

1. Deben ejecutarse sobre sistema operativo *Android*<sup>3</sup> a partir de su versión 5.0.
2. Deben soportar un visor de mapas que consuma cartografía sin conexión.
3. No requiere autenticación.
4. Encuestador y Notificador tienen que consumir de un servicio externo la cartografía y las propiedades con la información que hay que mostrar.
5. Encuestador y Notificador tienen que exportar hacia el servicio externo los datos modificados o adicionados.
6. Configuración mínima, interfaz visual sencilla y de fácil uso en las aplicaciones móviles.
7. Encuestador tiene que mostrar un mapa con las propiedades dadas por el servicio externo y cuando se de clic sobre alguna de estas mostrar los formularios con la información asociada a cada una. Se tiene que permitir salvar las modificaciones hechas en los formularios.
8. Notificador tiene que mostrar un mapa con los puntos dados por el servicio externo y cuando se de clic sobre algún punto mostrar el formulario de notificación que permite certificar que la propiedad fue encuestada.

Dado el problema y los requerimientos de los clientes se presenta como objetivo general diseñar e implementar las aplicaciones móviles Encuestador y Notificador para optimizar

<sup>1</sup>Tributo municipal aplicable anualmente sobre bienes inmuebles, calculado con base en datos catastrales (valor del terreno, construcción, uso), según la Ley de Tributación Municipal y reglamentos específicos de la Alcaldía de Managua.

<sup>2</sup><https://www.cadica.com/>

<sup>3</sup>Sistema operativo móvil basado en el núcleo Linux y otros softwares de código abierto.

el proceso de recolección de información sobre propiedades y su verificación física, reduciendo los tiempos operativos y mejorando la eficiencia en la gestión del catastro municipal de la Alcaldía de Managua, Nicaragua.

## 1. Análisis de aplicaciones con funcionalidades similares

En el proceso investigativo se analizaron herramientas existentes para el trabajo con mapas y que permitan visualizar cartografía sin conexión como Google Maps [7], MAPS.ME [8], All-In-One Offline Maps [13] cuyas funcionalidades son similares a los requerimientos planteados por el cliente. Este análisis se realizó con el objetivo de adoptar o mejorar las funcionalidades dependiendo del caso. Sobre el análisis realizado se concluye lo siguiente:

1. Las herramientas mencionadas permiten la creación y el manejo de puntos de interés, en nuestro caso serían propiedades que estarían representadas por tachuelas en el mapa, pero tienen formularios estáticos que no se pueden modificar, cuyos campos no coinciden con los asociados a una propiedad.
2. Los visores de mapas con que cuentan las aplicaciones analizadas necesitan de internet para mostrar la cartografía, inclusive aunque haya que descargar el mapa una sola vez para manejarlo sin conexión. Esto incumple con uno de los requerimientos del cliente.
3. Las herramientas para el manejo de mapas accesibles desde un dispositivo móvil, permiten llegar a usuarios no expertos y producen una mejor explotación de la información geográfica en variadas situaciones.
4. Los trabajos encaminados al manejo de información geográfica desarrollados para dispositivos móviles van en aumento, por varios factores, entre los que están el bajo costo de los terminales en relación con los ordenadores, así como su creciente uso. Esto unido a las posibilidades que abren estos dispositivos por su capacidad portable, suscita el interés de muchos para ampliar y desarrollar soluciones en este tema.
5. Son numerosas las aplicaciones orientadas al trabajo con mapas actualmente, mientras algunas de estas definen un API<sup>4</sup> de forma pública para que terceros puedan utilizar parte de las funcionalidades o incluso implementar sus propias aplicaciones. Entre las populares es notable Google Maps, reconocida por su espectro de funcionalidades. Si se busca un API con un acceso gratuito y con una información actualizada de la cartografía global, es recomendable OpenStreetMap [4] debido a sus funcionalidades y la experiencia de los autores con su uso.

<sup>4</sup>Application Programming Interface (por sus siglas en inglés): es un conjunto de rutinas, funciones y procedimientos que ofrece una biblioteca para ser utilizada por otro software

A partir de los aspectos expuestos anteriormente se concluye que es necesario el diseño e implementación de las aplicaciones que cumplan con los requerimientos expuestos en este documento, porque en general estas aplicaciones analizadas no permiten trabajar sobre mapas sin conexión y asociarle a cada punto de interés en el mapa su formulario correspondiente acorde a la información necesaria sobre las propiedades.

## 2. Diseño e implementación de la propuesta de solución

En esta sección se presentan los resultados de la investigación realizada sobre las distintas plataformas para el desarrollo de aplicaciones móviles. Se muestran además, elementos del diseño de una arquitectura extensible, haciendo uso de los principios de desarrollo SOLID<sup>5</sup> [11] y empleando buenas prácticas de programación, así como el modelo de datos definido. Brindar mecanismos de acceso a los sensores del dispositivo móvil para una recopilación de datos más completa haciendo uso de recursos de los dispositivos móviles como el GPS o la cámara, entre otros.

### 2.1 Entorno de desarrollo

Se realizó un análisis comparativo sobre tecnologías que se pudieran emplear para dar solución al problema planteado. Las plataformas analizadas para desarrollo móvil fueron: *React Native* [1], *Flutter* [6], *Xamarin* [12]. Del análisis se pudo identificar como las tecnologías más prometedoras a *React Native* y *Flutter*. Se excluyó a *Xamarin* por su reducida comunidad, su bajo rendimiento en alta demanda y su gran tamaño de aplicación en comparación con el resto de las aquí analizadas.

Para el caso de *React Native*, se cuenta con una amplia comunidad, gran número de paquetes y tiene un buen rendimiento, pero posee un motor gráfico aún deficiente y carece de fluidez para las animaciones más simples. Mientras *Flutter*, a pesar de poseer una comunidad aún pequeña, es cooperativa y está creciendo en una magnitud superior a la de *React Native*. *Flutter* incluye herramientas que abordan los aspectos deficientes de *React Native* respecto al apartado gráfico, con una excelente documentación y un lenguaje de programación poderoso.

Luego de todo lo expuesto, sumado a la experiencia previa que se posee en el uso de *Flutter* para el desarrollo multiplataforma, fue seleccionado para implementar las aplicaciones Encuestador y Notificador.

Otro aspecto a tener en cuenta en el proceso de investigación, una vez conocido la plataforma de desarrollo, fue encontrar un paquete que funcione sobre *Flutter* y soporte el uso de mapas. Los paquetes analizados fueron: *google maps flutter* [5] y *flutter map* [9], quedando este último como el seleccionado porque permite el trabajo con mapas sin conexión, y que es un requerimiento de las aplicaciones a desarrollar.

<sup>5</sup>SOLID: Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion

## 2.2 Modelo de datos

Es un requerimiento que estas aplicaciones se sirvan de un servicio externo encargado de proporcionar todos los datos que van a ser utilizados en Encuestador y Notificador. Este servicio externo obtiene su información de la base de datos original del proyecto que se encuentra en *SQLServer*. Por cuestiones de seguridad, las aplicaciones no pueden acceder directamente a esta base de datos, por tanto se dispone de un servicio externo para mediar el flujo de datos. Además, dados los requerimientos del cliente y las condiciones de trabajo serían necesarias bases de datos embebidas, pues las aplicaciones no van a estar conectadas a ningún servidor ya que van a ser utilizadas en lugares donde no hay conectividad. *SQLite* es la tecnología por excelencia empleada en *Flutter* para la persistencia de los datos, por su característica portable y multiplataforma, pero de ser empleada en la solución propuesta serían necesarias migraciones para llevar la información a la base de datos original del proyecto.

Teniendo en cuenta lo descrito anteriormente, y dada la simplicidad de los datos, se determinó que era más factible trabajar con archivos JSON (*JavaScript Object Notation*, por sus siglas en inglés), un formato ligero de intercambio de datos, de fácil lectura y escritura para los usuarios, y además fácil de analizar y generar mediante un herramienta de software [3].

Estos archivos cuentan con una estructura bien definida, para su uso se diseñó una jerarquía de clases que consta en el caso de Encuestador de seis entidades principales relacionadas entre sí, estas son: *MarkerEntity*, *PropertyEntity*, *OwnerEntity*, *ImprovementEntity*, *BuildingEntity* e *ImprovementBuildingEntity*. Están también las clases *NomenclatorEntity* e *InitialsEntity*. En el caso de Notificador se tienen dos entidades fundamentales: *MarkerEntity* y *PropertyEntity*. Además, en aras de incluir la funcionalidad de obtener un camino de costo mínimo para brindarle al encuestador una propuesta de cuál es el mejor recorrido para visitar las propiedades asignadas, se tienen en ambas aplicaciones las clases *IDistanceEntity* y *EuclideanDistance*. En la figura 1 se muestra el diagrama de clases para la aplicación Encuestador.

## 2.3 Arquitectura

Se investigaron varias arquitecturas que logran la separación de obligaciones, dividiendo el software en capas. Valorando la capacidad de extender la solución así como las facilidades para la realización de pruebas y teniendo en cuenta la plataforma de desarrollo, se optó por una Arquitectura N-Capas [10]. Dadas las características del proyecto se considera apropiado el uso del patrón de arquitectura de dos capas, las cuales son descritas a continuación:

- **Presentation:** En esta capa se encuentra toda la interfaz del sistema. Sus principales tareas son: obtener información del usuario, enviar la información del usuario a la capa de servicios e infraestructura para su procesamiento y presentar información al usuario. Está compuesta principalmente por *Widgets*, que contiene además el *widget* asociado al visor de mapas del paquete *flutter*

*map*. Aquí se encuentran vistas como: *home page*, que es el *entrypoint*<sup>6</sup> de cada aplicación; seguida de esta se tiene la vista *configuration page*, donde se cargan todos los archivos de configuración; luego está *map page* que despliega el mapa y todas las funcionalidades del visor de mapas; después viene *marker page* que se encarga de mostrar el formulario asociado a cada propiedad. En el caso de Notificador es un formulario que contiene *property tab*, Encuestador está formado por varias pestañas para organizar el código y cuidar el aspecto visual de la aplicación, estos son *property tab*, *building tab*, *improvement tab* y *owner tab*; por último en el caso de Encuestador se tiene la vista *help page* donde se encuentra el Manual de Ayuda, accesible desde *marker page*.

- **Service Infrastructure:** En esta capa se encuentran las entidades de uso general en la aplicación, contiene además toda la lógica de negocio, con los servicios que atienden a cada caso de uso y el acceso a los sensores del terminal. Esta capa es totalmente independiente de la capa de presentación. Se encuentran todas las entidades para el mapeo de los JSONs, en el caso de Encuestador se incluyen entidades como: *marker entity*, *property entity*, *building entity*, *owner entity*, *improvement entity* e *improvement building entity*, todas estas necesarias para trabajar con los formularios de las propiedades; se encuentran además las entidades *nomenclator entity* e *initials entity*, para el Manual de Ayuda. En el caso de Notificador solo se tiene *marker entity* y *property entity* para el trabajo con los formularios. Se incluyó además una funcionalidad para determinar el orden en que el usuario debía visitar las propiedades tal que la distancia recorrida fuera mínima; para esto se implementó un criterio de ordenación basado en la Distancia Euclideana entre dos coordenadas. Se diseñó de modo tal que pudieran ser utilizados otros criterios de ordenación, con solo incluir una clase que implemente el método *getDistance* de la clase abstracta *IDistanceEntity*. Se implementan en esta capa además servicios como: el acceso a los archivos del teléfono para obtener el mapa, los archivos de configuración y las imágenes, mediante los métodos *mapPicker()*, *jsonPicker()* e *imgFromGallery()*, ubicados todos en el script *pickers.dart*, se incluye el método *imgFromCamera()* para acceder a la cámara del terminal y tomar fotos; el método *loadJson()* para leer y decodificar los archivos de configuración de las propiedades y los nomencladores que luego van a ser mapeados en *mapMarkerEntities()* y *nomenclatorEntities()*, estos métodos se encuentran en los scripts *load json.dart* y *json mapper.dart* respectivamente; en *load configuration.dart* está *loadData()*, método que decodifica los archivos de configuración del mapa y los mapea para su posterior

<sup>6</sup>Punto de inicio del flujo de ejecución de un programa, donde se accede al resto del sistema

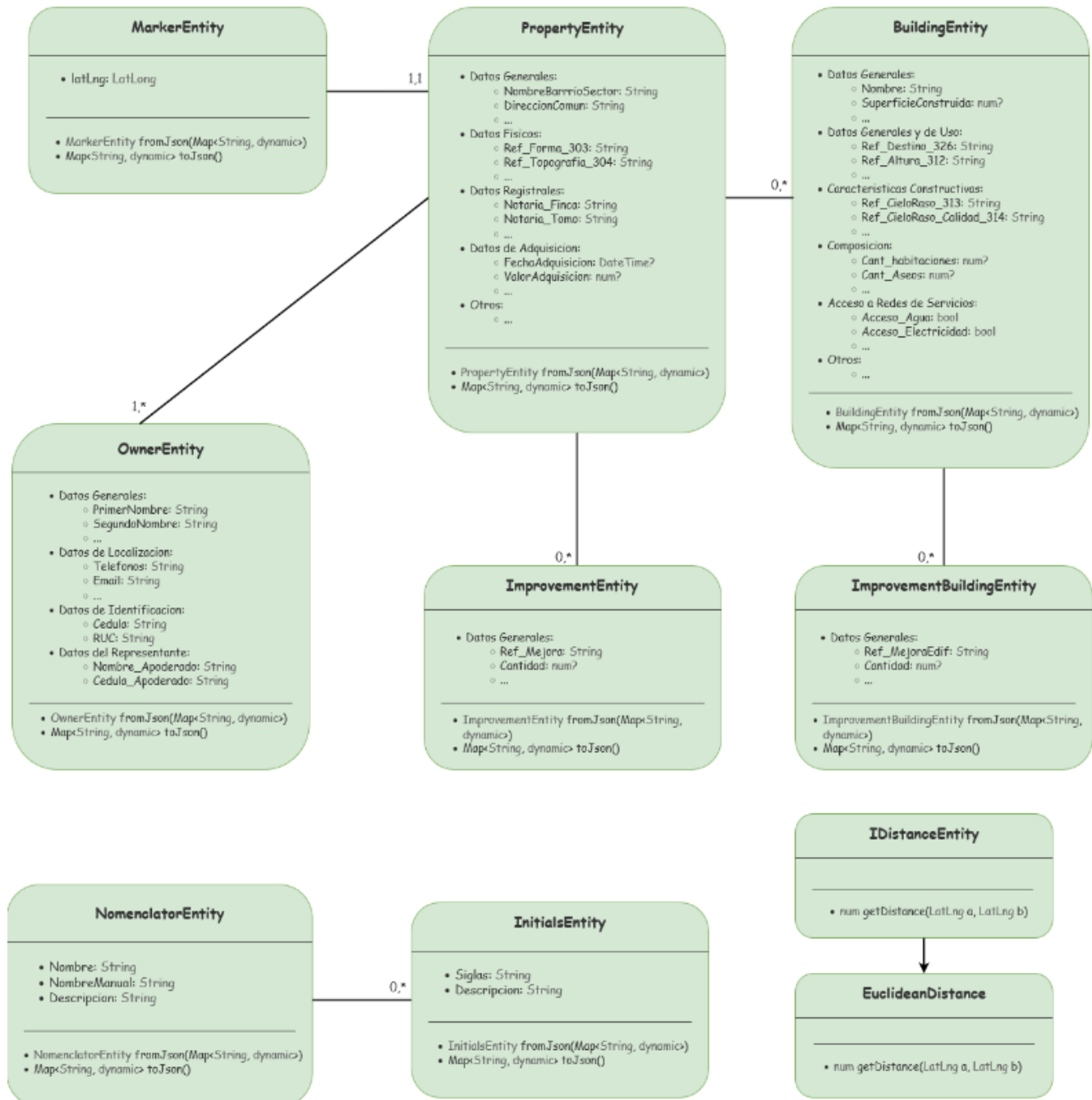


Figura 1. Diagrama de Clases de Encuestador  
 Figure 1. Encuestador Class Diagram

utilización; tenemos a *saveData()* encargado de generar y guardar en el terminal un nuevo JSON con las actualizaciones realizadas sobre las propiedades; tenemos el script *validates.dart* con métodos de validación manual utilizados para validar algunos campos requeridos en los formularios. Tenemos a *loadLocation()* para obtener la posición GPS<sup>7</sup> del dispositivo; y *getCode()*

en el caso de Notificador para escanear códigos QR<sup>8</sup>. Están implementados bajo el directorio */shorted path/* los métodos *getCompareEntities()* para obtener todos los criterios de ordenación desarrollados y *get path* que

<sup>7</sup>Sistema de Posicionamiento Global, Global Positioning System por sus siglas en inglés. Sistema que permite conocer la posición de un objeto o de una persona gracias a la recepción de señales emitidas por una red de satélites  
<sup>8</sup>Código de Respuesta Rápida, Quick Response code por sus siglas en inglés. Es un método para almacenar información por medio de contrastes de color y bidimensional

devuelve las propiedades ordenadas a partir del criterio de ordenación seleccionado utilizando BFS<sup>9</sup> para su solución.

Una ampliación de los detalles de la arquitectura propuesta se puede encontrar en [3].

### 3. Validación de las funcionalidades

Para demostrar el adecuado funcionamiento de las aplicaciones desarrolladas se realizan un conjunto de pruebas funcionales sobre un teléfono inteligente con las siguientes características:

- Procesador Octa-Core Max 2.0GHz.
- Memoria RAM<sup>10</sup> de 4 GB
- Sistema Operativo *Android* versión 8.1.

Se iniciaron las pruebas de funcionalidad partiendo de una cartografía en formato XYZ.png<sup>11</sup> proveniente del servicio externo, un visor de mapas centrado en Managua, Nicaragua e información de dos propiedades dadas en formato JSON provenientes del servicio externo.

#### 3.1 Mostrar en el mapa las propiedades provenientes del JSON en la aplicación Encuestador

A partir del JSON proveniente del servicio externo se visualizan en el mapa las propiedades mediante unos marcadores. En la figura 2 se muestra un fragmento de código del JSON con dos propiedades y sus respectivas coordenadas para poderlas ubicar en el mapa. En la figura 3, se muestra el visor con la cartografía de Managua en la aplicación Encuestador, donde se pueden observar los dos marcadores correspondientes a cada propiedad que existe en el JSON.

```
[
  {
    "Lat": 12.139191638116024,
    "Long": -86.23839286398542,
    "Propiedad": { ...
  },
  {
    "Lat": 12.139069044937118,
    "Long": -86.23640442459774,
    "Propiedad": { ...
  }
]
```

Figura 2. JSON con dos propiedades  
Figure 2. JSON with two properties

<sup>9</sup>Breadth First Search, por sus siglas en inglés. Es un algoritmo de búsqueda no informada utilizado para recorrer o buscar elementos en un grafo

<sup>10</sup>Memoria de acceso aleatorio, Random Access Memory por sus siglas en inglés

<sup>11</sup>Mapa de referencia en el que superponer datos de capas y visualizar información geográfica

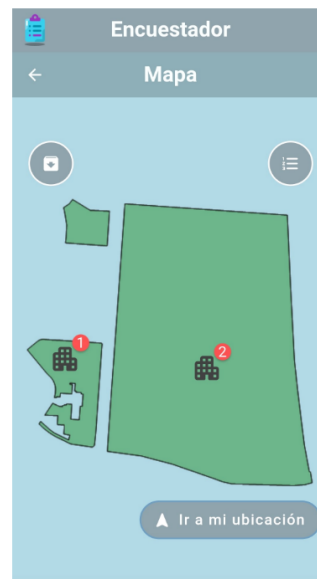


Figura 3. Cartografía de Managua con dos propiedades ubicadas

Figure 3. Cartography of Managua with two properties located

#### 3.2 Mostrar la información asociada a una propiedad en el formulario

Cuando se hace clic sobre algún marcador en el visor se muestra el formulario asociado a la propiedad con la información obtenida del JSON. En la figura 4 se muestra un fragmento del JSON con algunos campos de una de las propiedades. En las figuras 5 y 6, en la aplicación Encuestador, se muestra el formulario con los valores de algunos de los campos asociados a la propiedad mostrada en el JSON anterior. El formulario de una propiedad es anidado ya que además de tener campos simples también tiene listas de otros elementos como edificios, mejoras y propietarios.

```
"Id": 2022046,
"IdVisible": false,
"CodigoSisca": "1003_US11004001009_2021",
"CodigoSiscaVisible": false,
"NombreBarrioSector": "VILLA CUBA LIBRE",
"NombreBarrioSectorVisible": true,
"Direccion Coman": "TERMINAL DE BUSES 164, 2C. SUR, C. ESTE, M.D",
"Direccion_comanVisible": true,
"NombrePopularRazonComercial": null,
"NombrePopularRazonComercialVisible": true,
"Ref_Origen_Avaluo": "10",
"Ref_Origen_AvaluoVisible": true,
"Ref_ConceptoAvaluo": null,
"Ref_ConceptoAvaluoVisible": true,
"Ref_Usopredominante": "001",
"Ref_UsopredominanteVisible": true,
"Ref_Forma_303": "RIG",
"Ref_Forma_303Visible": true,
"Ref_Topografia_304": "13C",
"Ref_Topografia_304Visible": true,
"Ref_Nivel_306": "MIV",
"Ref_Nivel_306Visible": true,
" Frente_calle": 42,
" Frente_calleVisible": true,
" Longitud_fachada": 42,
" Longitud_fachadaVisible": true,
" Longitud_fondo": 166,
```

Figura 4. JSON con algunos datos generales y físicos  
Figure 4. JSON with general and physical data.

The screenshot shows the 'Encuestador' app interface for a 'Propiedad' form. The title bar is 'Encuestador' and the page title is 'Propiedad'. Below the title bar is a navigation bar with icons for home, list, search, and user profile. The main content area is titled 'Datos Generales' and contains several input fields: 'Nombre de Barrio\*' with the value 'VILLA CUBA LIBRE', 'Dirección Común\*' with 'TERMINAL DE BUSES 164, 2C.', 'Nombre Popular\*' with 'null', 'Origen Avaluo\*' with '10', 'Concepto ...' with 'null', and 'Uso Predominante\*' with '001'. At the bottom are 'Cancelar' and 'Guardar' buttons.

Figura 5. Formulario con campos generales  
Figure 5. Form with general fields

The screenshot shows the 'Encuestador' app interface for a 'Propiedad' form, specifically the 'Datos Físicos' section. The title bar is 'Encuestador' and the page title is 'Propiedad'. Below the title bar is a navigation bar with icons for home, list, search, and user profile. The main content area is titled 'Datos Físicos' and contains several input fields: 'Forma del ...' with 'REG', 'Nivel del ...' with 'NIV', 'Topografía\*' with 'INC', 'Frente Calle\*' with '# 42', 'Long. Fachada\*' with '# 42', 'Long. Fondo\*' with '# 166', 'Superf. ...' with '# null', and 'Superf. Terreno\*' with '# null'. At the bottom are 'Cancelar' and 'Guardar' buttons.

Figura 6. Formulario con campos físicos  
Figure 6. Form with physical fields

### 3.3 Mostrar estrategias de visita

El proyecto incluye una funcionalidad con el propósito de facilitar el proceso de visita de los usuarios a las propiedades. En la aplicación Encuestador, partiendo de algún criterio de ordenación se le proporciona al usuario un orden de visita a las propiedades de modo que la distancia recorrida sea mínima. El botón superior derecho que se aprecia en la figura 3, muestra una lista con todos los criterios de ordenación disponibles en la aplicación, ver figura 7. Una vez seleccionado alguno los números que aparecen en las propiedades ubicadas en

el mapa se ordenarán, mostrando el orden en que el usuario debe visitar las propiedades. El orden inicial que aparece en las propiedades no es el óptimo, sino el orden en que fueron proporcionados los datos de las propiedades.

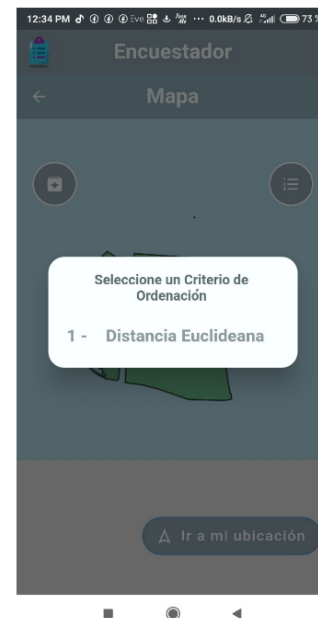


Figura 7. Lista de las estrategias de visita disponibles  
Figure 7. List of available visiting strategies

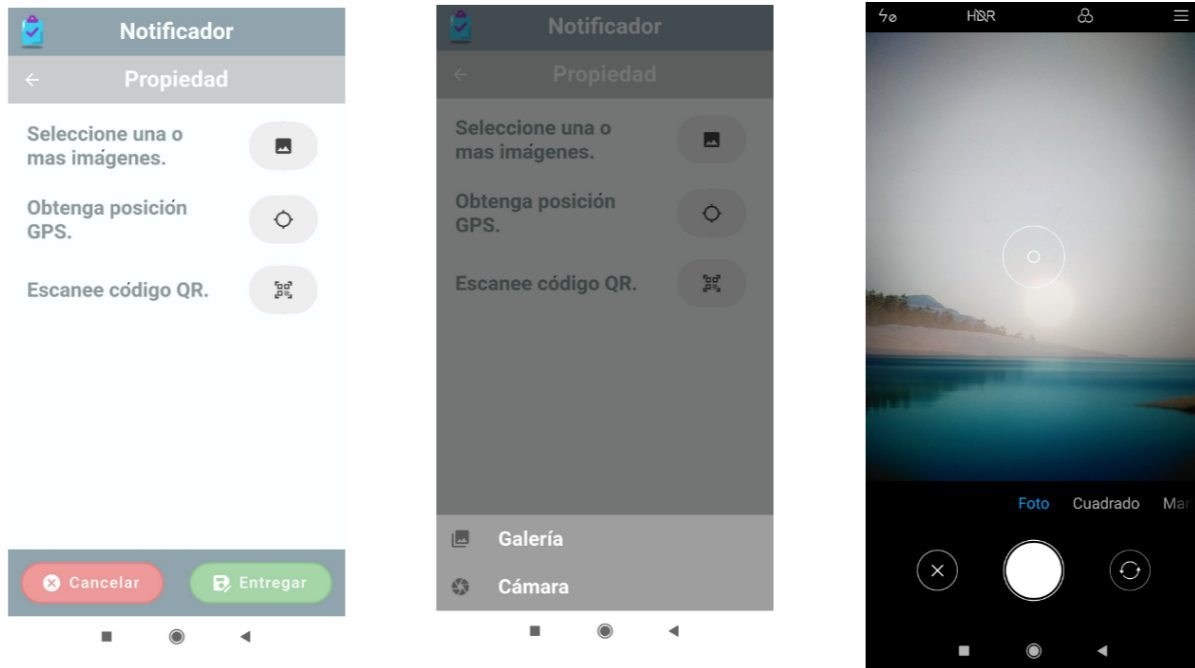
### 3.4 Mostrar las funcionalidades de notificación

En la aplicación Notificador, las funcionalidades obtener la posición del GPS, escanear un código QR y tomar o seleccionar imágenes para certificar que una propiedad fue encuestada funcionan correctamente. En la figura 8 se muestra el formulario de certificación de una propiedad, donde los campos Imágenes, Posición GPS y Código QR están vacíos. Además se aprecian las dos formas de seleccionar imágenes que ofrece la aplicación, si el usuario decide tomar una foto, este tiene acceso a la cámara del teléfono.

En la figura 9 se aprecia la posición del GPS del dispositivo de prueba, ubicado en La Habana, demostrando así el funcionamiento de este requisito de notificación. La figura muestra cómo se accede a la cámara del terminal para escanear, o bien centrando algún código QR en el cuadrado que se observa en el medio de la figura 9 o seleccionando alguna imagen que contenga un código QR, de cualquier forma el código puede ser escaneado correctamente.

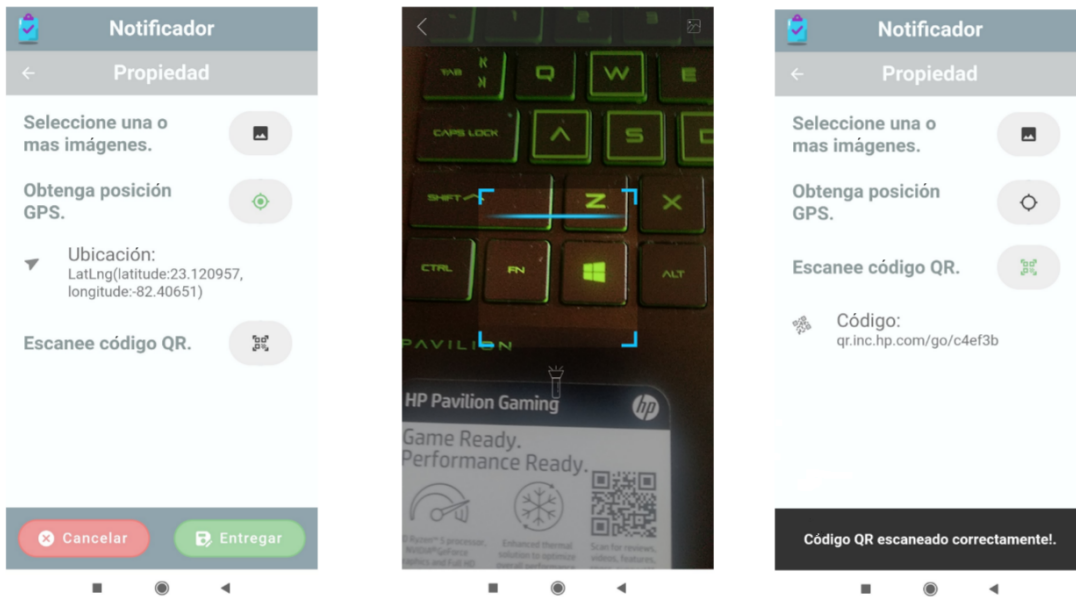
## 4. Conclusiones

El presente trabajo cumple su objetivo principal de implementar dos aplicaciones móviles funcionales: Encuestador, encargada de recopilar información de propiedades, y Notificador, destinada a certificar la realización de las encuestas. Ambas aplicaciones satisfacen integralmente los requerimientos establecidos por los clientes.



**Figura 8.** A la izquierda formulario de notificación con todos sus campos vacíos. En el centro tipos de selección de imágenes. A la derecha cámara del dispositivo.

**Figure 8.** On the left, the notification form with all fields empty. In the center, image selection types. On the right, the device's camera



**Figura 9.** A la izquierda campo Posición GPS con las coordenadas de La Habana. En el centro cámara del dispositivo con el escáner disponible. A la derecha campo Código QR con el código escaneado por la cámara.

**Figure 9.** On the left, the GPS position field with Havana's coordinates. In the center, the device's camera with the scanner available. On the right, the QR code field with the code scanned by the camera

El análisis de aplicaciones existentes con funcionalidades similares evidenció la necesidad de diseñar e implementar desde cero las aplicaciones, asegurando una solución adaptada a los requerimientos específicos del catastro municipal de Managua.

La elección de *Flutter* como plataforma de desarrollo y *flutter map* para la gestión de mapas sin conexión demostró su capacidad para cumplir con los criterios de portabilidad, rendimiento y seguridad exigidos por el cliente.

Las pruebas funcionales confirmaron que ambas aplicacio-

nes optimizan la recolección de información georreferenciada mediante formularios, garantizan la verificación física de propiedades mediante notificaciones y mejoran la eficiencia en la gestión catastral.

La solución no solo resuelve un problema tangible para la Alcaldía de Managua, sino que establece un patrón tecnológico escalable para futuros desarrollos con la adopción de *Flutter* como plataforma de desarrollo, arquitectura N-Capas para escalabilidad y extensibilidad de futuras funcionalidades y *flutter map* para el trabajo con mapas. Este patrón, no es una solución monolítica e inmutable; su escalabilidad reside en una arquitectura modular que requiere de ajustes y reconfiguraciones específicas para ser aplicado en otros escenarios, como diferentes municipios o tipos de recolección de datos. La complejidad inherente no radica en alterar el núcleo del sistema, sino en adaptar sus componentes parametrizables (como los formularios de datos, los flujos de trabajo y las reglas de validación) para alinearse con los nuevos requisitos operativos, legales o técnicos. Así, el esfuerzo de implementación se reduce significativamente en comparación con un desarrollo desde cero, validando el uso de herramientas modernas y *frameworks* ágiles que facilitan esta personalización.

### Relevancia del estudio

Las aplicaciones móviles cartográficas descritas, basadas en formularios, son herramientas esenciales para la recolección de información georreferenciada en trabajo de campo, fuera del entorno de oficina. Su utilidad radica en facilitar la labor en sectores como ingeniería, arquitectura y ciencias sociales, ofreciendo portabilidad y seguridad en el manejo de datos hasta su procesamiento. Estas aplicaciones presentan múltiples casos de uso: gestión de infraestructura eléctrica (mantenimiento de torres), exploración petrolera (estudios de expansión), gestión hídrica (detección y priorización de roturas), saneamiento urbano (monitoreo de vertederos) o fiscalización gubernamental (recaudación, encuestas). La relevancia principal de la investigación subyacente reside en el desarrollo de las aplicaciones móviles Encuestador y Notificador que resuelven un problema real mediante tecnologías modernas. Este resultado sienta una base y establece un patrón para el futuro desarrollo de soluciones destinadas al levantamiento de información en campo.

### Suplementos

Este artículo no contiene información suplementaria.

### Agradecimientos

Los autores expresan su agradecimiento a los miembros del colectivo Casa del Software en la Facultad de Matemática y Computación, Universidad de La Habana por su invaluable apoyo técnico y colaboración durante la realización de este trabajo. Su contribución fue muy importante para el éxito de esta investigación.

### Conflictos de interés

Se declara que no existen conflictos de interés.  
Los autores declaran que no hubo subvenciones involucradas en este trabajo.

### Contribución de autoría

**Conceptualización** J.C.A., A.G.A.

**Análisis formal** J.C.A., A.G.A.

**Investigación** J.C.A., A.G.A.

**Metodología** J.C.A., A.G.A.

**Administración de proyecto** J.C.A.

**Software** J.C.A.

**Supervisión** J.C.A.

**Validación** A.G.A.

**Visualización** J.C.A., A.G.A.

**Redacción: preparación del borrador original** J.C.A.

**Redacción: revisión y edición** A.G.A.

### Referencias

- [1] Bonnie Eisenman: *Learning React Native*. Primera edición, 2016, ISBN 1491929006.
- [2] Bosque Sendra, J.: *Sistemas de información geográfica*. Ediciones Rialp, S.A, MADrid, segunda edición, 1997.
- [3] Castro, Sheyla Cruz: *Aplicaciones móviles para la recogida y notificación de datos en campo sobre las propiedades de Managua, Nicaragua*. Trabajo de Diploma, Universidad de La Habana, 2022.
- [4] Corporate Members: *OpenStreetMap*. <https://www.openstreetmap.org>, visitado el 2022-11-15.
- [5] Flutter, Google map: *google maps flutter para Flutter*. [https://pub.dev/packages/google\\_maps\\_flutter](https://pub.dev/packages/google_maps_flutter), visitado el 2022-09-28.
- [6] Flutter, Sitio Oficial: *Flutter*. <https://flutter.dev>, visitado el 2022-10-11.
- [7] Google: *Google Maps*. <https://www.google.es/maps>, visitado el 2022-10-01.
- [8] Group, Mail.ru: *Maps.me Official Web Site*. <https://maps.me>, visitado el 2022-09-28.
- [9] map, Flutter flutter: *flutter map para Flutter*. [https://pub.dev/packages/flutter\\_map](https://pub.dev/packages/flutter_map), visitado el 2022-11-18.
- [10] Mark Richards, Neal Ford: *Fundamentals of Software Architecture: An Engineering Approach*. Primera edición, 2020, ISBN 9781492043454.

- [11] Martin, Robert C: *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall Press, primera ed edición, 2017, ISBN 0134494164.
- [12] Microsoft: *Xamarin*. <https://dotnet.microsoft.com/es-es/apps/xamarin>, visitado el 2022-11-24.
- [13] Psyberia: *All-In-One OfflineMaps*. <https://www.offline-maps.net/>, visitado el 2022-11-12.

