

Universidad Tecnológica de La Habana

“José Antonio Echeverría”

Facultad de Ingeniería Informática

**Guatiní 3.0: Enciclopedia móvil**

**Informe de Práctica Profesional 2**

**Autor:**

Damian Aldair Garcia Pérez (<mailto:dgarciape@ceis.cujae.edu.cu>)

**Tutor:**

Dr.C Juan Carlos Sepúlveda Peña (<mailto:jcarlos@ceis.cujae.edu.cu>)

La Habana, Cuba

Abril de 2023

**Resumen**

El proyecto Guatiní surge como proyecto extensionista, con el propósito de inculcar, en las nuevas generaciones preferentemente, los conocimientos sobre la avifauna cubana y la importancia del cuidado del medio ambiente.

Tras varios ejercicios de prácticas profesionales y tesis de diploma de estudiantes de la Universidad Tecnológica de La Habana “José Antonio Echeverría”, CUJAE, se han ido desarrollando aplicaciones que cumplieron su objetivo de servir como “multimedias” sobre aves endémicas cubanas.

El objetivo principal es aumentar el alcance del proyecto, eliminado el límite actual de poseer sólo aves endémicas de Cuba y dar la posibilidad de incluir información de seres vivos de todas las especies.

Como resultado del ejercicio se espera la implementación de una aplicación tipo enciclopedia, capaz de asimilar la información de cualquier ser viviente que se desee introducir en la base de datos.

**Palabras clave:** Aplicación móvil, enciclopedia, seres vivos.

***Abstract***

*The* “Guatiní” *project arose as an extension project, with the purpose of instilling in the new generations, above all, knowledge about the cuban birds and the importance of caring for the environment.*

*After several exercises of professional practices and diploma thesis of students of* “Universidad Tecnológica de La Habana José Antonio Echeverría, CUJAE”*, applications have been developed to serve as “multimedia” about cuban endemic birds.*

*The main objective is to increase the scope of the project, removing the current limit of having only endemic birds of Cuba and giving the possibility of including information of all species.*

*As a result of the exercise, it is expected the implementation of an encyclopedia-type application, capable of assimilating the information of any living being that one wishes to enter in the database.*

***Key words:*** *Mobile application, encyclopedia, living beings.*

**Índice**

[Introducción 1](#_Toc107763891)

[Capítulo 1: Fundamentación teórica 5](#_Toc107763892)

[1.1. Aplicaciones tipo enciclopedia en la actualidad 5](#_Toc107763893)

[1.2. Taxonomía biológica 6](#_Toc107763894)

[1.3. Plataformas objetivo 7](#_Toc107763895)

[1.4. Tecnologías para el desarrollo de aplicaciones 8](#_Toc107763896)

[1.5. Tecnología a utilizar en el desarrollo de la aplicación móvil: Flutter 10](#_Toc107763897)

[1.6. Base de datos: SQLite 12](#_Toc107763898)

[Conclusiones parciales 13](#_Toc107763899)

[Capítulo 2: Solución propuesta 14](#_Toc107763900)

[2.1. Técnicas de captura de información 14](#_Toc107763901)

[2.2. Modelado del negocio 18](#_Toc107763902)

[2.3. Captura de requisitos 21](#_Toc107763903)

[2.4. Descripción detallada de los Casos de Uso 25](#_Toc107763904)

[2.5. Diseño de la solución 35](#_Toc107763905)

[2.5.1. Diseño de la base de datos 35](#_Toc107763906)

[2.5.2. Diagrama de clases 39](#_Toc107763907)

[2.6. Descripción de la distribución 39](#_Toc107763908)

[2.6.1. Requisitos mínimos 39](#_Toc107763909)

[2.6.2. Requisitos recomendados 39](#_Toc107763910)

[Conclusiones parciales 43](#_Toc107763911)

[Capítulo 3: Validación de la solución 44](#_Toc107763912)

[3.1. Prueba funcional: Buscar artículo de una especie 44](#_Toc107763913)

[3.2. Prueba funcional: Visualizar artículo 46](#_Toc107763914)

[3.3. Resultados de las pruebas 47](#_Toc107763915)

[Conclusiones parciales 47](#_Toc107763916)

[Conclusiones 48](#_Toc107763917)

[Recomendaciones 49](#_Toc107763918)

[Referencias bibliográficas 50](#_Toc107763919)

[Anexos 52](#_Toc107763920)

**Índice de tablas**

[Tabla 1: Descripción detallada del caso de uso Buscar artículo de especie 25](#_Toc107763868)

[Tabla 2: Descripción detalla del caso de uso Visualizar artículo 28](#_Toc107763869)

[Tabla 3: Descripción detalla del caso de uso Reproducir contenido multimedia 31](#_Toc107763870)

[Tabla 4: Descripción detalla del caso de uso Seleccionar Base de datos 34](#_Toc107763871)

[Tabla 5: Descripciones de las tablas contenidas en la base de datos. 37](#_Toc107763872)

[Tabla 6: Prueba funcional del caso de uso Buscar artículo de una especie 44](#_Toc107763873)

[Tabla 7: Prueba funcional del caso de uso Visualizar artículo 46](#_Toc107763874)

**Índice de figuras**

[Figura 1: Cuota de mercado de los sistemas operativos móviles en el mundo en el período mayo 2021 - mayo 2022 8](#_Toc107763522)

[Figura 2: Diagrama de casos de uso del negocio 18](#_Toc107763523)

[Figura 3: Diagrama de actividades del caso de uso “Obtener información de una especie” 18](#_Toc107763524)

[Figura 4: Diagrama de la subactividad “Aplicar filtros de búsqueda”. 19](#_Toc107763525)

[Figura 5: Diagrama de objetos de negocio 19](#_Toc107763526)

[Figura 6: Diagrama de Casos de Uso del Sistema 21](#_Toc107763527)

[Figura 7: Modelo lógico de la base de datos 36](#_Toc107763528)

[Figura 8: Modelo físico de la base de datos 36](#_Toc107763529)

[Figura 9: Diagrama de clases. Parte 1 40](#_Toc107763530)

[Figura 10: Diagrama de clases. Parte 2 41](#_Toc107763531)

[Figura 11: Diagrama de clases. Parte 3 42](#_Toc107763532)

# **Introducción**

**Antecedentes**

El proyecto Guatiní tiene sus orígenes varios años atrás, cuando nació como proyecto extensionista con el fin de promocionar la cultura medioambiental en los niveles de enseñanza primario, medio, medio superior y universitario principalmente.

En sus inicios el proyecto tenía como objetivo la creación de “multimedias” enfocadas en la divulgación de información únicamente de la avifauna cubana. Posteriormente este contenido fue empaquetado en una aplicación móvil desarrollada para el sistema operativo Android, lo que ayudó grandemente a su fácil distribución. En la aplicación móvil, junto a la información de las aves, también se encontraban juegos que ayudaban a los más pequeños a memorizar la información. También se creó, para ordenador, una herramienta de autor o sistema de gestión de base de datos que se encargaba de manipular la información a mostrar en la aplicación móvil anteriormente mencionada. Estas aplicaciones, desarrolladas por el grupo de Investigación de Aplicaciones Móviles de la Facultad de Ingeniería Informática de la Universidad Tecnológica de La Habana, José Antonio Echeverría (CUJAE), obtuvieron el Premio Provincial de Innovación Tecnológica de la provincia de La Habana en 2020 [1].

**Presentación de la idea de investigación**

Actualmente, el proyecto Guatiní pretende eliminar el límite de contener información sólo de la avifauna cubana y ayudar al fomento del conocimiento en nuestro país de especies de seres vivos en general. Se pretenden realizar tres aplicaciones:

1. Un gestor que se encargue del manejo de todos los datos que estarían contenidos en la base de datos que se utilizará como fuente de información para la aplicación móvil. Se desarrollará de forma tal que pueda correr en diferentes sistemas operativos de escritorio.
2. Una aplicación móvil que será compilada con la base de datos generada anteriormente por el gestor y que se encargará del despliegue de la misma en el dispositivo móvil. El fin de esta aplicación es poder difundir la base de datos de manera sencilla y en un formato que sea manejado en las tiendas de aplicaciones móviles.
3. Una aplicación principal, también orientada a dispositivos móviles, que se encargue de leer toda la información de la base de datos anteriormente desplegada y mostrársela al usuario de manera potable para este.

Este ejercicio de Prácticas Profesionales se centra en el punto 3.

**Situación problemática**

Muchas personas recurren a los motores de búsqueda, como Google, para buscar la información, obteniendo en ocasiones resultados parciales, erróneos o incluso, no acorde a lo que se pretende buscar.

Las enciclopedias más conocidas, tanto las offline como las online, poseen gran volumen de información y no se especializan solo en especies de seres vivos.

La solución actual (las versiones existentes) no están adaptadas para mostrar información de cualquier especie que se desee registrar en la base de datos.

Referente a la población fuera de la comunidad científica, la difusión de la aplicación actualmente es casi nula y no está disponible en la tienda cubana de aplicaciones móviles Apklis.

**Problema**

La aplicación actual solo está diseñada para mostrar contenido de aves.

**Objetivo general**

Implementar una nueva versión de la aplicación perteneciente al proyecto Guatiní para hacerla multiplataforma, atractiva, acorde a los estándares de diseño actuales y capaz de mostrar correctamente información sobre cualquier especie que se necesite introducir en la base de datos que leerá.

**Objetivos específicos**

* Analizar el contexto actual:
  + Analizar aplicaciones tipo enciclopedia similares al objetivo final para identificar características necesarias a implementar en el proceso de desarrollo.
  + Analizar conceptos relacionados con la taxonomía biológica para el posterior diseño de la base de datos a utilizar.
  + Analizar la características y bondades de los sistemas operativos más utilizados en Cuba para conocer a qué sistemas operativos objetivo se debe orientar el desarrollo de las aplicaciones.
  + Analizar diferentes tecnologías para el desarrollo de las aplicaciones para seleccionar una tecnología que cumpla con las necesidades del proyecto.
* Implementar nuevas versiones de las aplicaciones que sean capaces de asimilar todo tipo de especies:
  + Identificar los requisitos de las nuevas versiones de las aplicaciones para mejorar y agilizar el diseño de las aplicaciones.
  + Realizar el diseño de la base de datos que contendrá toda la información a mostrar.
  + Realizar el diseño visual de las aplicaciones.
  + Realizar implementación de las nuevas aplicaciones.
* Validar la solución propuesta:
  + Realizar pruebas para verificar el correcto funcionamiento de las nuevas versiones de las aplicaciones.

**Preguntas de investigación**

* ¿Ya existe una versión anterior o aplicación similar?
* De ser afirmativa la respuesta anterior, ¿en qué innovaría las nuevas versiones que motivarían a las personas a utilizarlas?
* ¿Qué tipo de especies incorporar a la aplicación?
* ¿Qué características tienen estas especies y cómo clasificarlas?
* ¿Para qué sistemas operativos desarrollar las nuevas versiones?
* ¿Qué tecnologías usar en el desarrollo de las nuevas versiones?
* ¿Cómo sería el diseño de las aplicaciones?
* ¿Dónde desplegar las aplicaciones una vez terminado su desarrollo?

**Evaluación**

Las nuevas versiones aportarían al proyecto aplicaciones más atractivas visualmente, acordes con los estándares de diseño actuales para aplicaciones móviles y capaces de mostrar información de cualquier tipo de especie que se introduzca en la base de datos.

Con la tienda cubana de aplicaciones móviles, Apklis, se podría difundir la aplicación y llegar a gran parte de la población que esté interesada en conocer sobre las diferentes especies de seres vivos.

Se lograría fomentar conocimientos sobre el cuidado del medio ambiente y su importancia.

Las nuevas versiones de las aplicaciones se podrían instalar en la gran mayoría de dispositivos móviles que circulan en Cuba actualmente, lo que ayuda a la difusión del proyecto.

**Factibilidad**

La utilización de Flutter como framework de desarrollo de UI actual y propio del propietario del sistema operativo Android hace que sea posible la implementación de últimas funcionalidades en las versiones finales de las aplicaciones móviles y que estas estén optimizadas para una ejecución satisfactoria.

El despliegue en Apklis ayudaría a una mayor difusión y todos tendrían la capacidad de obtener las aplicaciones con relativa facilidad.

El uso de estas aplicaciones ayudaría a tomar conciencia sobre el cuidado del medio ambiente a partir del conocimiento de las especies contenidas en la base de datos.

# **Capítulo 1: Fundamentación teórica**

* 1. **Aplicaciones tipo enciclopedia en la actualidad**

A continuación se hace un análisis de diferentes aplicaciones móviles actuales que actúan como enciclopedias.

**Wikipedia**

Existe una aplicación móvil de esta enciclopedia [Anexo 1], desarrollada por la Fundación Wikimedia. Gracias a esta aplicación es posible consultar contenidos enciclopédicos, definiciones o simplemente leer por ocio, sin necesidad de acudir a la versión web [2].

La aplicación puede ser descargada desde la Apple App Store y la Google Play Store para los sistemas operativos iOS y Android respectivamente.

Como ventajas de que tiene esta aplicación se encuentra la posibilidad de consultar los contenidos de Wikipedia en cualquier momento y lugar gracias al uso del dispositivo móvil, utilizar la aplicación no tiene costo adicional alguno y se pueden guardar varios artículos para una posterior lectura sin conexión.

La principal desventaja de esta aplicación es que para obtener la información es necesario una conexión a los servidores de Wikipedia vía Internet.

**Kiwix**

Kiwix realmente es un proyecto de software libre que da la posibilidad de acceso a sitios web basados en MediaWiki[[1]](#footnote-1) sin conexión a Internet [Anexo 2]. Es más famoso por ser el lector offline de Wikipedia soportado oficialmente por la Fundación Wikimedia [3].

Gracias a que está destinado a un uso *offline*, esta aplicación trabaja con bases de datos con extensión ZIM, donde se encuentra toda la información. Estas bases de datos no se distribuyen junto a la aplicación por lo se suma la tarea de conseguir los archivos de bases de datos, al menos una vez.

**EcuMóvil**

EcuMóvil surge como la idea de crear una versión de la EcuRed Portátil para dispositivos móviles. Como antecedente, ya existía la propia EcuRed Portátil que se desplegaba para ordenador y que el proyecto Kiwix contaba con una versión para dispositivos móviles. Tras 3 años en que la aplicación se caracterizaba por un aspecto poco profesional [Anexo 3], EcuMóvil implementa todo el contenido en una nueva versión basada en el proyecto Kiwix, lo que permitió implementar imágenes y mejorar la política de actualizaciones [4].

**Guatiní (versión existente)**

Como parte del proyecto extensionista Guatiní, se desarrollaron varias versiones de la aplicación móvil por parte del departamento del grupo de Investigación de Aplicaciones Móviles de la Facultad de Ingeniería Informática de la Universidad Tecnológica de La Habana, “José Antonio Echeverría” (CUJAE). Estas aplicaciones fueron muy diversas: unas con un aspecto muy apegado al ámbito de los juegos y otras con un estilo más parecido al resto de aplicaciones de Android, sistema operativo en donde se despliega [1].

La última versión [Anexo 4] logró un comportamiento de mutación en dependencia de los módulos que se le incorporaban: Proyecto Guatiní, Murciélagos cubanos, Artes Plásticas. De esta forma se lograba una “peculiar unión” de los contenidos de estos módulos dentro de la misma aplicación mediante la actualización de esta. Con estas acciones el desarrollo de la aplicación se desviaba del objetivo real del proyecto.

* 1. **Taxonomía biológica**

La taxonomía biológica en la ciencia que clasifica jerárquica y sistemáticamente los grupos de seres vivos. Dentro de estas clasificaciones es necesario destacar las más importantes que servirán para el diseño de la base de datos que contendrá la información.

Dominio: Categoría taxonómica utilizada en los sistemas de clasificación biológica que se encuentra por arriba del reino.

Reino: Representa cada una de las grandes subdivisiones taxonómicas en las que se clasifican los seres vivos respecto a su parentesco evolutivo.

Filo: Categoría taxonómica situada entre el reino y la clase. Al referirse a las plantas suele utilizarse también el término “división”, aunque ambos son equivalentes.

Clase: Categoría taxonómica situada entre el filo y el orden.

Orden: Categoría taxonómica entre la clase y la familia. Antiguamente era sinónimo de familia.

Familia: Unidad sistemática y categoría taxonómica situada entre el orden y el género.

Género: Categoría taxonómica que se ubica entre la familia y la especie. Grupo de organismos que a su vez puede dividirse en varias especies.

Especie: Unidad básica de clasificación biológica. Conjunto de organismos o poblaciones naturales capaces de entrecruzarse y producir descendencia fértil.

* 1. **Plataformas objetivo**

En la Figura 1 se muestra un gráfico extraído de StatCounter que muestra los sistemas operativos móviles más utilizados actualmente en el mundo. Se puede evidenciar claramente que el sistema operativo más utilizado es Android, con alrededor del 71%, seguido de iOS con un 28% aproximadamente. El resto de sistemas operativos móviles no son notorios respecto a los anteriormente mencionados, por lo que adoptar iOS y Android como plataformas objetivo para el desarrollo de la aplicación garantiza una gran cuota de mercado en el proceso de despliegue.

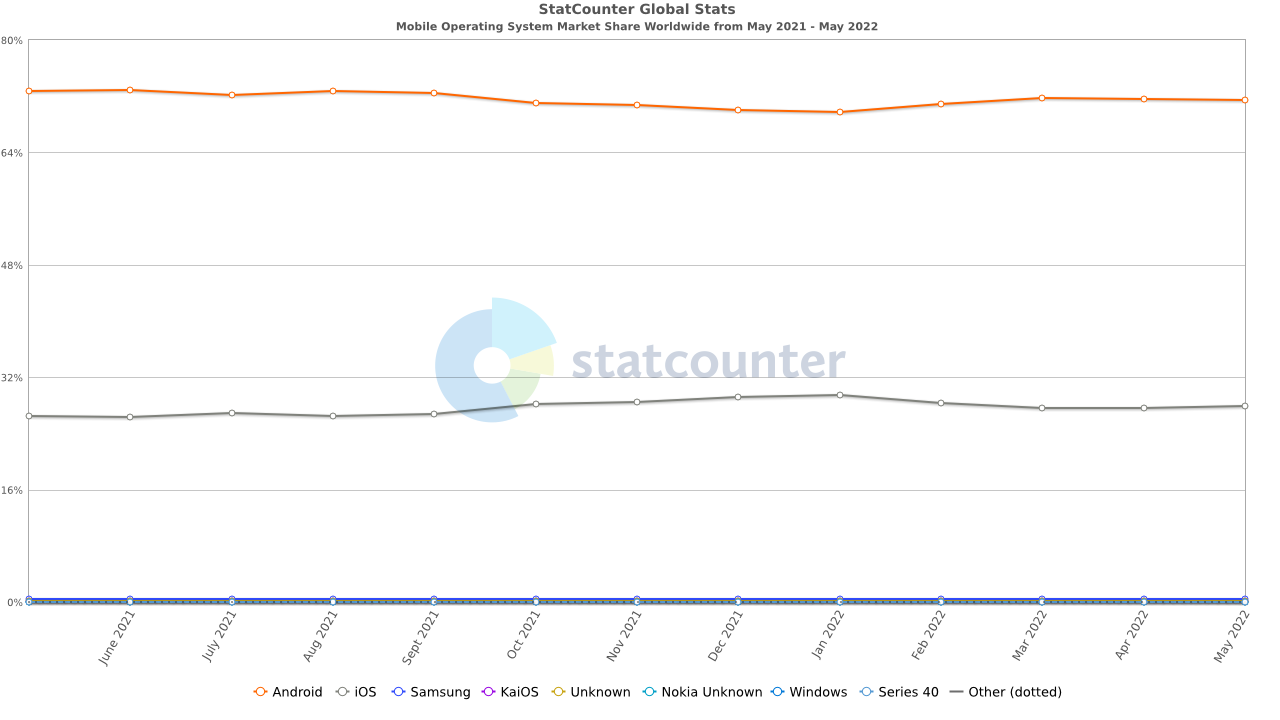


Figura 1: Cuota de mercado de los sistemas operativos móviles en el mundo en el período mayo 2021 - mayo 2022

* 1. **Tecnologías para el desarrollo de aplicaciones**

Para el desarrollo de aplicaciones móviles existen varios lenguajes de programación que se podrían adoptar.

Para el sistema operativo Android se puede programar aplicaciones en Java donde se utilizarían librerías especiales para correr en Android Runtime (ART), sucesor de Dalvik; ambas son máquinas virtuales de Java modificadas, adaptadas para dispositivos móviles. También existe Kotlin, lenguaje desarrollado por JetBrains, que es interoperable con Java al utilizar la Máquina Virtual de Java (JVM) y es más conciso al poder generar aplicaciones similares con un 40% menos de líneas de código que con Java [5].

Para iOS y iPadOS se puede desarrollar en Swift, lenguaje presentado en 2014 y desarrollado por Apple específicamente para el desarrollo en iOS, iPadOS y macOS. También se puede utilizar Objective-C, pero no se recomienda [6].

Si bien programar de forma nativa las aplicaciones tiene el gran beneficio de poder implementar características específicas para cada sistema operativo y en ocasiones necesarias para la solución de *bugs*, el uso de marcos de trabajo o *frameworks* es totalmente recomendable en el momento del desarrollo de dichas aplicaciones gracias a la velocidad que se alcanza en el proceso de escritura del código de las aplicaciones, además de proveer en muchas ocasiones la posibilidad de crear aplicaciones multiplataformas simultáneamente [7].

**Ionic**

Ionic es un framework que permite desarrollar aplicaciones con tecnología web como HTML5, CSS y JS para desarrollar y ejecutar las aplicaciones móviles, lo que requiere el envoltorio o *wrapper* Cordova para acceder a los controladores de la plataforma nativa. No ofrece un rendimiento similar al de las aplicaciones nativas. En cuanto al desarrollo de aplicaciones, no utiliza componentes nativos pero ofrece una apariencia similar a la de las aplicaciones nativas. Esto se hace utilizando tecnologías web. La ventaja de usar Ionic es que la prueba de las aplicaciones es un proceso rápido. Esto agiliza todo el proceso de desarrollo de aplicaciones [8].

**React Native**

React Native es un *framework* que utiliza JavaScript, combina los beneficios de este lenguaje y ReactJS. La gran ventaja es que permite al desarrollador escribir las partes de código en Swift, Objective-C o Java siempre que sea necesario. Si se desea manejar operaciones pesadas en la aplicación, existen acceso a módulos y bibliotecas nativas en aplicaciones basadas en React Native [9].

Proporciona a las aplicaciones un rendimiento similar al de las nativas. Esto se debe a que toma elementos del código que son específicos de las APIs (interfaz de programación de aplicaciones, del inglés: *Application Programming Interface*) de React Native.

Estos módulos nativos se programan utilizando los lenguajes nativos, lo que facilita el acceso de los desarrolladores. La desventaja de esto es que no se puede reutilizar para otras plataformas. Estos módulos nativos sólo ofrecen un alto rendimiento. Los módulos de React Native trabajan con los controladores de UI (interfaz de usuario, del inglés: *User Interface*) nativos, lo que ofrece una mejor experiencia de usuario similar a la de las aplicaciones nativas. También hace uso de las librerías ReactJS y de los elementos de interfaz de usuario, lo que consolida el desarrollo de la interfaz de usuario.

**Xamarin**

Xamarin es un *framework* que combina lo mejor de C# y .Net para desarrollar aplicaciones para dispositivos móviles y Mac. Las aplicaciones desarrolladas con Xamarin funcionan de forma similar a las aplicaciones nativas.

Se puede desarrollar la interfaz de usuario utilizando Xamarin.Android/iOS o Xamarin.Forms. El primer método consume mucho tiempo, pero ofrece un aspecto similar al de las aplicaciones nativas en lo que respecta a la UX (experiencia de usuario, del inglés: *User eXperience*). Cuando se utiliza Xamarin.Forms, el proceso de desarrollo es más fácil y consume menos tiempo, pero, este desarrollo no garantiza el aspecto y la sensación que es similar a las aplicaciones nativas [10].

**Flutter**

Cuando se busca construir aplicaciones de alta calidad utilizando *frameworks*, muchos desarrolladores seleccionan Flutter, el cual utiliza Dart como lenguaje de programación. Dart es un lenguaje procedural que ofrece varios beneficios y está influido por C#, JavaScript y Java. Gracias a Dart, los desarrolladores utilizan Flutter para desarrollar aplicaciones con cierta complejidad de una manera relativamente sencilla. Ofrece una velocidad increíble, se puede lograr una UI totalmente nativa y no se utiliza JavaScript para salvar la interacción con los componentes nativos del dispositivo [11].

* 1. **Tecnología a utilizar en el desarrollo de la aplicación móvil: Flutter**

Dado que la empresa detrás del *framework* Flutter y la que desarrolla actualmente Android es la misma, Google, se garantiza siempre una adopción rápida de las últimas características añadidas para el sistema operativo en sus últimas versiones listas para su implementación, por lo que se optó por utilizar este *framework* junto al lenguaje de programación Dart.

* Herramientas de desarrollo:
  + Flutter: Kit de desarrollo de código abierto desarrollado por Google y enfocado al desarrollo rápido de aplicaciones móviles multiplataformas [12].
  + Gradle: Sistema de automatización de construcción de código abierto que se encarga de ensamblado de las aplicaciones para su posterior depuración o lanzamiento [13].
  + Android SDK: Kit de desarrollo necesario para desarrollar aplicaciones en la plataforma Android. Contiene el depurador de código, las librerías, los simuladores de dispositivos, la documentación, ejemplos de código y tutoriales [14].
  + Java Development Kit (JDK): Kit de desarrollo para programar en lenguaje Java. Necesario también para la programación para Android [15].
  + Java Runtime Environment (JRE): Entorno de ejecución para aplicación escritas en Java. Necesario también para la programación para Android [15].
* Lenguaje de programación:
  + Dart: Lenguaje de programación creado por Google y utilizado para el *framework* Flutter.
* Fundamentación de la selección:

Flutter es un kit de desarrollo de código abierto, desarrollado por Google. Sus contribuciones se basan en la propia compañía, desarrolladora del sistema operativo Android, y la comunidad, por lo que tiende a ser un SDK (kit de desarrollo de software, del inglés: *Software Development Kit*) bastante actualizado acorde a las últimas características implementadas por el sistema Android o requeridas por los desarrolladores [12].

Este *framework* es capaz de crear aplicaciones de códigos diferentes (por ejemplo: Swift para iOS y Kotlin para Android) desde un mismo código fuente escrito en Dart, lenguaje de programación propio de Flutter.

Permite el desarrollo rápido con implementación de “*hot reload*” y “*hot restart*”, que son funciones depuración que permite ver al desarrollador los cambios realizados en el código en tiempo real y recargar la aplicación en el dispositivo virtual o físico sin necesidad de recompilar toda la aplicación, proceso que normalmente demora.

Está enfocado en la experiencia del usuario. Es capaz de crear aplicaciones que sean “*responsives*”, o sea, que se adapten a cualquier tamaño de pantalla. También posee un rendimiento prácticamente igual al nativo, brindando un comportamiento nativo según la plataforma y unas velocidades de carga rápidas.

* Entorno de Desarrollo Integrado a utilizar (IDE, del inglés *Integrated Development Environment*):

Aunque es de obligación tener instalado Android Studio para el desarrollo de aplicaciones Android, dado que este es el software que gestiona nativamente todo lo relacionado con el desarrollo, se seleccionó Visual Studio Code:

Visual Studio Code es un IDE desarrollado por Microsoft. Soporta depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código y refactorización de código. Es muy personalizable por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Posee capacidad de adición de extensiones por lo que es posible añadirle características. Su desempeño es bastante bueno, consumiendo menos memoria RAM que otros IDE, como Android Studio o similares, basados en IntelliJ Platform. Es gratuito y de código abierto.

* 1. **Base de datos: SQLite**

Para almacenar información y administrarla es muy común utilizar una base de datos. Desde hace bastante tiempo, las bases de datos se manejan del lado del servidor y los dispositivos móviles se comunican con ellos mediante la red para obtener la información. Sin embargo, en ocasiones en necesario que la aplicación sea menos dependiente de la conectividad de la red.

Para ello es necesario almacenar la base de datos en el propio almacenamiento del propio dispositivo, por lo que la base de datos debe ser ligera, rápida, segura y que consuma poca memoria y energía.

La utilización de bases de datos SQLite es muy popular en el ámbito de los desarrolladores móviles ya cumple con la ligereza y la rapidez que se quiere, además de ser una base de datos relacional, con todos los beneficios que conlleva.

## **Conclusiones parciales**

Según lo analizado en la fundamentación teórica:

* iOS y Android son los sistemas operativos móviles más usados mundialmente, por lo que los esfuerzos deben ir es esa dirección.
* Flutter es el framework idóneo para el desarrollo rápido de las aplicaciones y lograr el factor multiplataforma junto con el rendimiento y el atractivo.
* Las bases de datos SQLite son una buena alternativa para lograr un buen almacenamiento y administración de los datos de la aplicación.

# **Capítulo 2: Solución propuesta**

En este capítulo se exponen los detalles de la solución propuesta desarrollada para la resolución del problema anteriormente planteado, como el proceso de captura de requisitos y la descripción de sistema. Se utilizan diversos artefactos UML que ayudan a explicar los diseños implementados en el desarrollo de la aplicación.

* 1. **Técnicas de captura de información**

Para un eficiente desarrollo de software es necesario aplicar técnicas de captura de información, lo que ayuda a identificar requisitos y a ahorrar tiempo que puede ser mejor utilizado.

**Técnicas de encuestas**

Entrevista

Se logró hacer una entrevista al Dr. Alejandro Barro Cañamero, profesor de la Facultad de Biología de la Universidad de La Habana [Anexo 5]. La confección de la entrevista tomó aproximadamente unos 15 minutos y aplicación alrededor de otros 15 minutos; cabe aclarar que estos sucesos no sucedieron el mismo día. Esta entrevista tenía la finalidad de obtener información hacer acerca datos generales acerca de los seres vivos y cómo debería organizarse la información relacionada con estos. Se obtuvo información acerca de la clasificación de los seres vivos y sus niveles de conservación. Dicha información ayuda a un correcto diseño de la base de datos que se utilizará para almacenar toda la información que mostrará la aplicación, donde se pueden percibir posibles tablas como Dominio, Reino, Filo, Clase, Orden, Familia, Género, Especie, Conservación, Multimedia y Localización. También se lograron determinar algunas de las reglas de negocio que se reflejarán más adelante.

Cuestionario

Para la confección de los formularios de utilizó la herramienta web Formularios de Google. La confección del cuestionario tomo alrededor de 20 minutos; actualmente es posible consultar resultados ya que sigue disponible para responder en el enlace siguiente: https://docs.google.com/forms/d/e/1FAIpQLScpEH5UgBl4ExuMe-quuEzsdcK9y\_vRq0GUS2FanKS5DWFrBQ/viewform?usp=sf\_link. La aplicación de este cuestionario tenía como finalidad conocer el uso de los dispositivos móviles por parte del cubano de hoy y conocer el nivel de difusión del Proyect Guatiní. Gracias a la fácil difusión de esta encuesta en estilo digital, hasta el día 11 de mayo de 2022 se alcanzó llegar a unas 148 personas [Anexo 6]. Gracias a la aplicación de este formulario se detectaron los siguientes puntos:

* Gran parte de la población usa el dispositivo móvil con mucha frecuencia [Anexo 7].
* Gran cantidad de personas vincula su dispositivo móvil en muchas tareas cotidianas [Anexo 8].
* La gran mayoría de la población prefiere aplicaciones simples y muchos de ellos con la posibilidad de obtener bastante información [Anexo 9].
* La mayoría de la población prefiere buscar información de manera digital [Anexo 10].
* Se necesita más difusión del Proyecto Guatiní [Anexo 11].

**Técnicas centradas en documentos**

Estándares

En Cuba coexisten dos grandes sistemas operativos móviles como son Android y iOS, por los que se decidió realizar el desarrollo de la aplicación móvil en el *framework* Flutter. Toda la documentación necesaria para la instalación del *framework*, herramientas necesarias para el desarrollo de aplicaciones y el cómo desarrollarlas se encuentra digitalmente en la página web oficial [16]. Se detectó que este *framework* da la posibilidad de que el desarrollador de encargue programar en un solo lenguaje de programación, Dart, y automáticamente se genera el código nativo en Kotlin para Android y en Swift para iOS.

Dart, como lenguaje de programación a utilizar para el desarrollo en el *framework* Flutter, también tiene su documentación de forma digital en su página web oficial [17]. Se detectó que Dart es un lenguaje que pretende ser una herramienta sencilla para proyectos grandes y ofrecer una mejor seguridad.

Dado que tanto el framework Flutter como Dart son propiedad de Google, esta empresa impulsa su línea de diseño Material Design, presentada en 2014 y con documentación de forma digital en su página web oficial [18]. Se detectó que este código de diseño, también llamado así por Google, promueve un diseño simple, elegante e intuitivo para facilitar su uso, íconos minimalistas y fácilmente adaptables, utilización de colores mayormente planos (aunque se aceptan degradados simples), tipografías simples y fáciles de visualizar, diseño tipo tarjetas en que se agrupan contenidos similares, el uso de barras de herramientas para acciones habituales, el empleo de botones bien diferenciados del fondo y fáciles de distinguir y el empleo de botones flotantes de acción para las acciones más simbólicas o importantes de la pantalla que se muestre en ese momento.

Arqueología del sistema

Para la realización de esta técnica, se abrió el archivo APK de la versión anterior de la aplicación, o sea, el instalador de la aplicación en dispositivos Android, con la herramienta de compresión/descompresión 7-zip [Anexo 12]. Esta técnica de aplicó para conocer qué formatos de archivo fueron utilizados en el desarrollo de la aplicación y detectar otros aspectos que surjan durante la ejecución de la técnica. Tras el análisis del contenido de la aplicación se detectaron los siguientes puntos:

* Utilización de Kotlin como lenguaje nativo [Anexo 13].
* Utilización de Flutter como framework de desarrollo [Anexo 14].
* Utilización de algunos paquetes específicos que se podrían utilizar para la realización de la nueva versión de la aplicación [Anexo 15].
* Utilización de un archivo DB para la base de datos [Anexo 16].
* El archivo DB se puede trabajar en SQLite [Anexo 17].
* Utilización de archivos MP3 para audios [Anexo 18].
* Utilización de archivos PNG para lograr transparencias si es necesario [Anexo 19].
* Utilización de un archivo SVG (imagen vectorial) para evitar la pérdida de calidad en momentos de zoom [Anexo 20].
* Utilización de archivos MP4 para videos [Anexo 21].
* Compilación de la aplicación realizada para todas las plataformas disponibles en Android [Anexo 22].

Reutilización

Tras el análisis de varios trabajos de diploma relacionados con las versiones anteriores del proyecto [19-21], en la búsqueda de aspectos que se puedan reutilizar para la futura versión de la aplicación, se detectaron los siguientes puntos:

* Uso del modelo relacional para el diseño de la base de datos.
* Búsqueda de artículos a partir de varios criterios de búsqueda: texto, códigos QR, ubicación manual en el mapa y ubicación en tiempo real.
* Implementación de juegos didácticos.
* Sistema operativo Android como objetivo para el despliegue de la aplicación.
* Visualizado de publicidad en cada pantalla de la aplicación.

**Técnicas de creatividad**

Técnica de analogía

Se realizó esta técnica con las aplicaciones tipo enciclopedia más conocidas actualmente: Kiwix [Anexo 2] y Wikipedia [Anexo 1] con la finalidad de detectar aspectos esenciales en dos las aplicaciones tipo enciclopedias más utilizadas un Cuba actualmente. Tras realizar un análisis se detectaron los siguientes puntos:

* Botón o barra para realizar búsquedas [Anexo 23].
* Botón para buscar palabras específicas dentro del artículo que se lee [Anexo 24].
* Botón para guardar marcadores o favoritos [Anexo 25].
* Tabla con resumen de las características más importantes [Anexo 26].
  1. **Modelado del negocio**

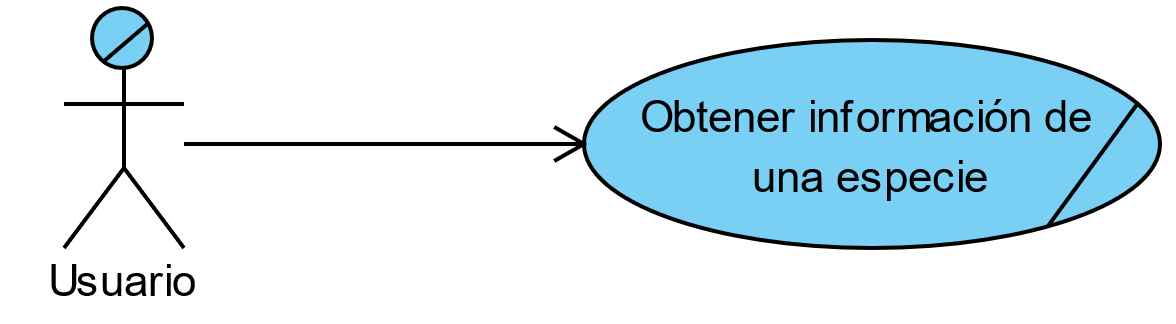


Figura 2: Diagrama de casos de uso del negocio

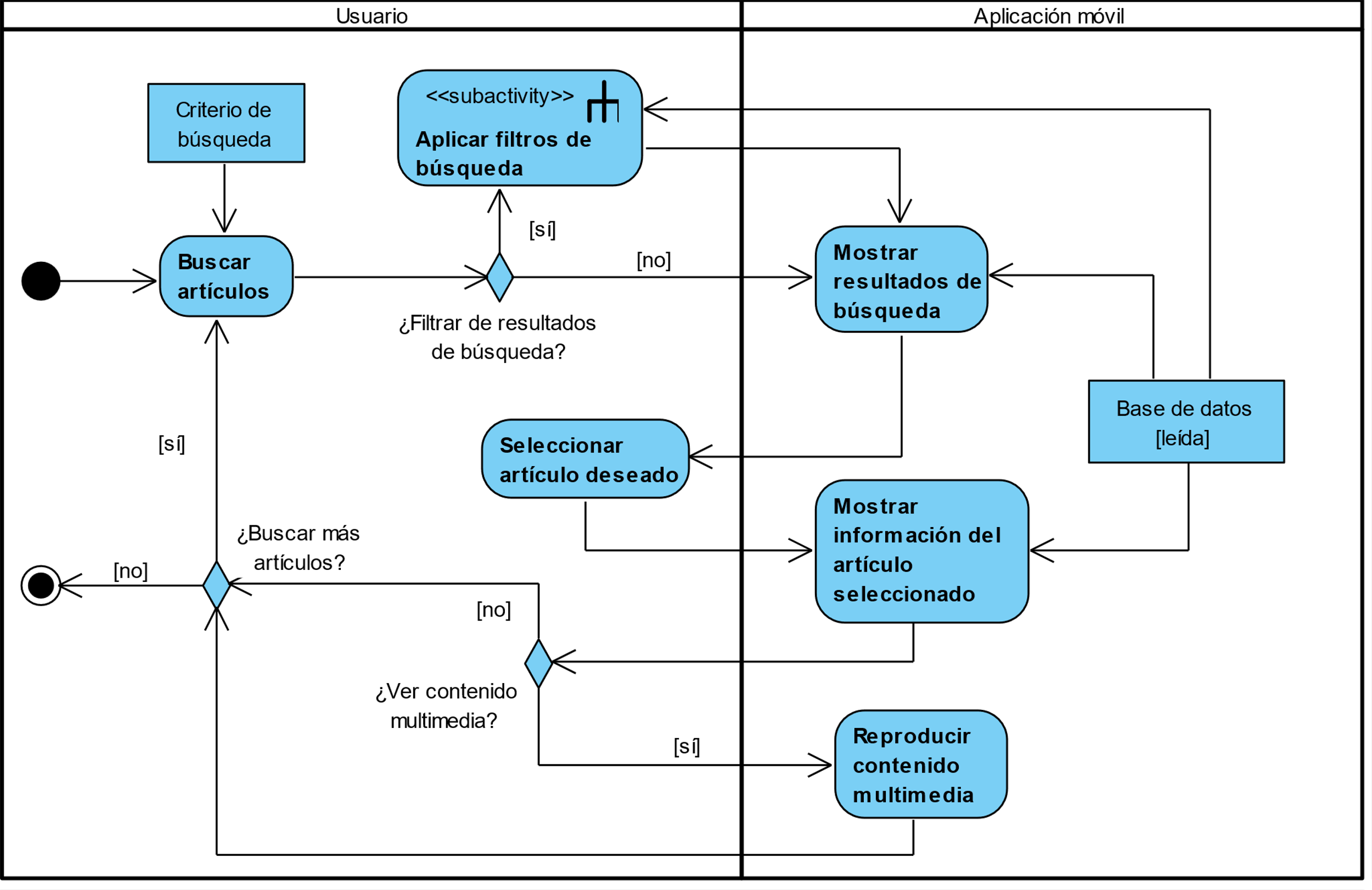


Figura 3: Diagrama de actividades del caso de uso “Obtener información de una especie”

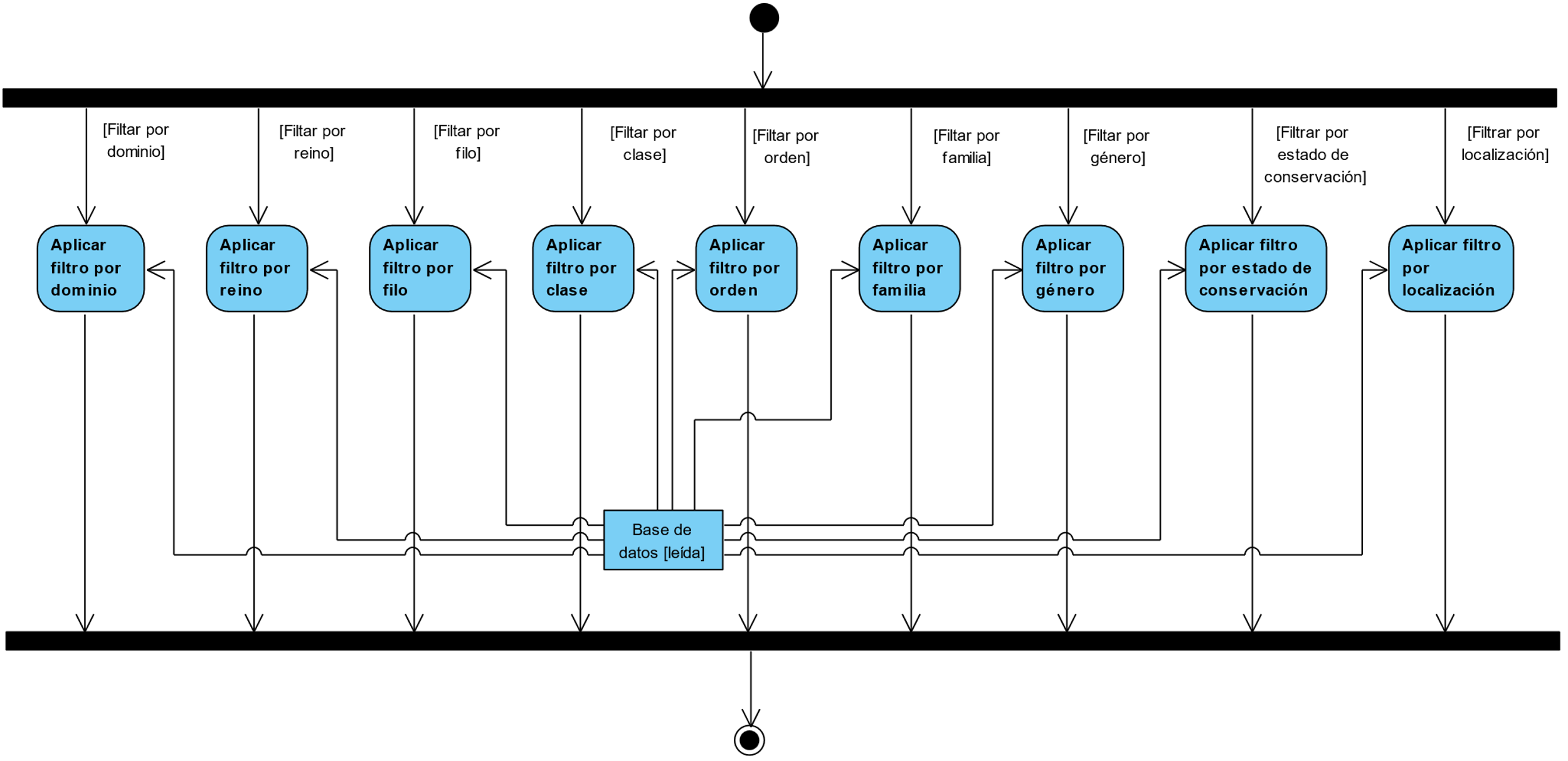


Figura 4: Diagrama de la subactividad “Aplicar filtros de búsqueda”.

El objeto “Criterio de búsqueda” hace alusión a una variable de entrada para la actividad “Buscar artículos”. Este criterio es una cadena de caracteres: puede ser una palabra, una frase o una ubicación geográfica.

El objeto “Base de datos” hace alusión a una carpeta que contiene los diferentes archivos necesarios para que la aplicación pueda mostrar la información deseada. Entre estos archivos se encuentran imágenes, videos, audios, mapas y un archivo DB, el cual será la base de datos relacional hecha con el Sistema de Gestión de Bases de Datos SQLite.

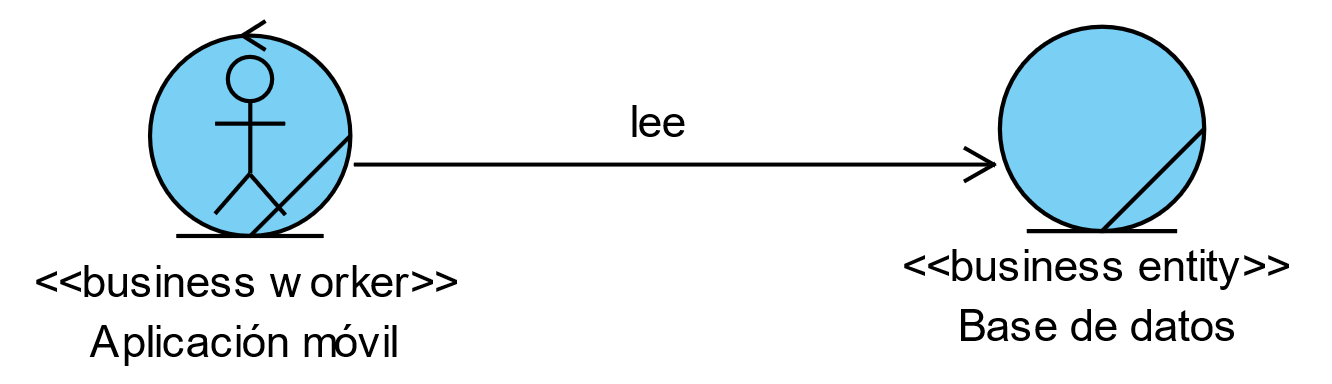


Figura 5: Diagrama de objetos de negocio

**Reglas del negocio**

Se han identificado las siguientes reglas de negocio:

Con patrón valor:

* Dominio = “Prokaryota” o “Eukarya”
* Reino = “Plantae” o “Animalia” o …
* Filo = “Acanthocephala” o “Acoelomorpha” o “Annelida” o “Arthropoda” o …
* Clase = “Mammalia” o “Insecta” o “Bacilli” o …
* Orden = “Trogoniformes” o “Carnivora” o “Siluriformes” …
* Familia = “Felidae” o “Trogonidae” o “Canidae” …
* Género = “Canis” o “Felis” o “Mus” …
* Especie = “Canis lupus” o “Felis silvestris” o “Pterois antenatta” o …
* Estado de conservación = “Preocupación menor” o “Casi amenazada” o “Vulnerable” o “En peligro” o “En peligro crítico” o “Extinta en estado silvestre” o “Extinta”
* Tipo de multimedia = “Imagen” o “Sonido” o “Video”
* Formato de imagen = “JPG” o “JPEG” o “PNG” o “SVG” o “BMP”
* Formato de video = “MP4” o “AVI” o “MPG” o “WMV”
* Formato de audio = “MP3” o “M4A” o “WMA” o “WAV” o “FLAC”

Con patrón Flujo de Control:

* Se necesita conocer el criterio de búsqueda para mostrar los resultados de búsqueda.

Con patrón Estímulo/Respuesta

* Si se aplica un filtro en alguna categoría taxonómica o aspecto, solo se muestran los artículos que cumplan con el filtro aplicado, sino se pueden mostrar todos los artículos que concuerden con el criterio de búsqueda.

Con patrón Repetición

Mientras que el usuario esté leyendo un artículo y desee seguir realizando búsquedas puede volver a abrir la pantalla de búsqueda.

* 1. **Captura de requisitos**

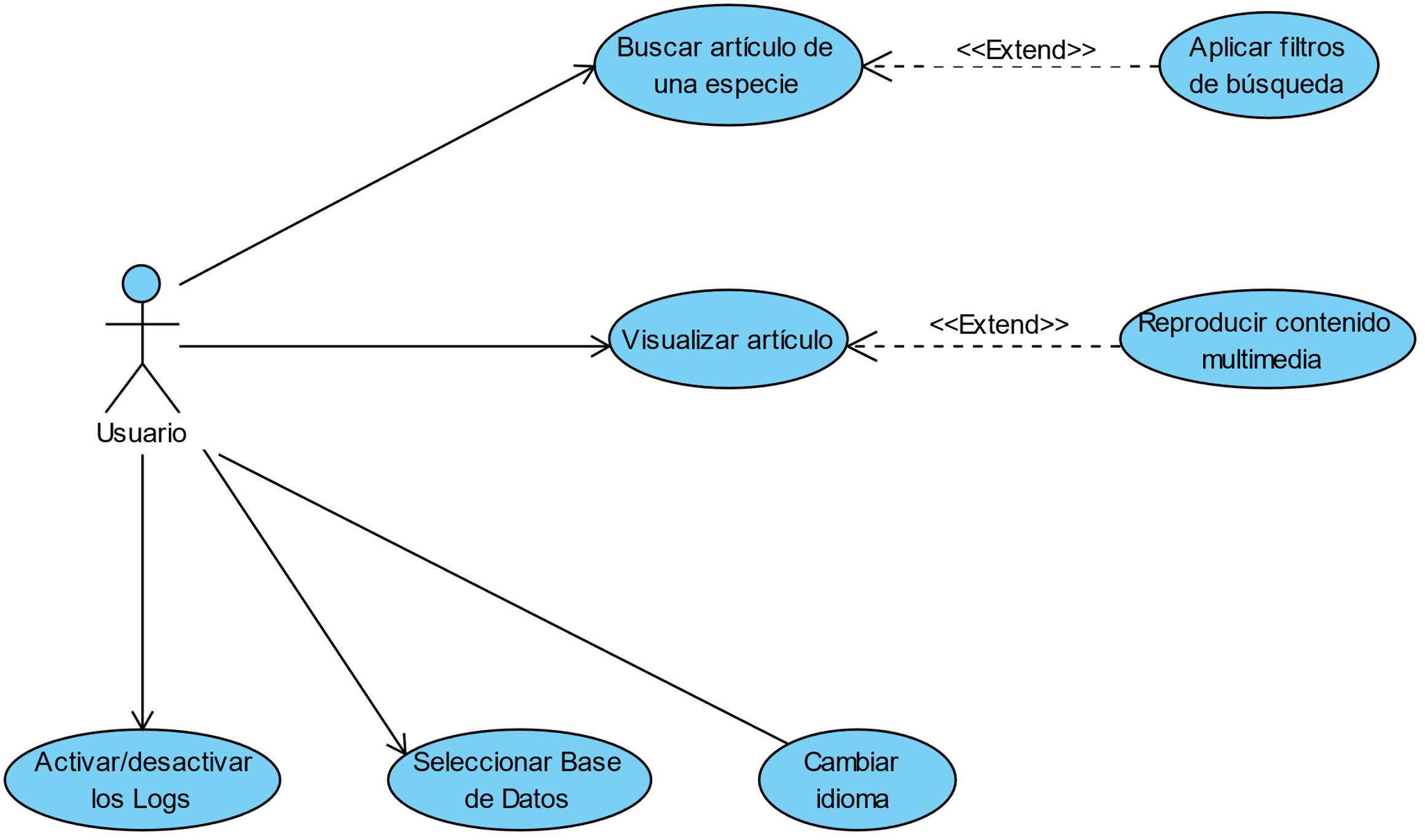


Figura 6: Diagrama de Casos de Uso del Sistema

**Requisitos funcionales**

1. Cuando se realiza una búsqueda el sistema debe mostrar sugerencias de resultados según el criterio de búsqueda en cada instante que se introduzca o de borre un carácter.
2. El sistema debe proveer al usuario la capacidad de aplicar filtros a su búsqueda.
3. El sistema debe discriminar los resultados a mostrar según los filtros aplicados.
4. El sistema debe leer la base de datos para mostrar la información contenida en ella.
5. El sistema debe mostrar en el artículo todo el contenido multimedia disponible en la base de datos correspondiente a dicho artículo.
6. El sistema debe proveer al usuario la capacidad de activar y desactivar los Logs.
7. El sistema debe proveer al usuario la capacidad de seleccionar la base de datos con la que desee trabajar.
8. El sistema deberá proveer al usuario la capacidad de seleccionar el idioma.

**Requisitos de calidad**

Requisitos de apariencia o interfaz externa

* El sistema debe ser legible (tener términos no rebuscados y que los tamaños de los elementos visuales y de la fuente sea acorde con los del dispositivo en que se ejecute).
* El sistema debe ser simple de usar y tener el comportamiento estándar y sencillo de una aplicación Android estándar.
* El sistema debe ser interactivo gracias a la incorporación de imágenes, audios, videos y mapas.
* El sistema debe tener una apariencia neutra y no enfocarse a nichos muy específicos, con estilo tipo Wikipedia o Kiwix, para garantizar llegar al mismo público.

Requisitos de usabilidad

* El sistema debe proveer al usuario la capacidad de cambiar el idioma de la aplicación.
* El sistema debe proveer al usuario la capacidad de documentarse acerca de la aplicación con explicaciones iniciales, una sección de Ayuda y el apartado Acerca de.

Requisitos de rendimiento

* El sistema debe tener alta velocidad de procesamiento, con un tiempo de respuesta menor a 0.5 s, aunque esto siempre depende de la capacidad de procesamiento del dispositivo móvil en que utilice la aplicación.

Requisitos de seguridad

* El sistema debe no guardar información sensible del usuario.
* El sistema debe estar siempre disponible dado que su contenido es *offline*.

Requisitos políticos y culturales

* El sistema debe poseer contenido política y culturalmente correcto en la base de datos para evitar conflictos religiosos, éticos, por costumbres o prejuicios.

Requisitos legales

* El sistema debe respetar las licencias de los contenidos multimedia empleados (Copyright, Creative Commons, Free).

Requisitos de confiabilidad

* El sistema debe recuperarse de fallos que se puedan dar en la lectura de la base de datos, el cierre inesperado de la app o el apagado inesperado del dispositivo.

Requisitos de software

* El sistema debe correr sobre Android 5.0 Lollipop o superior.
* El sistema debe detectar y leer la base de datos externa que posee la información.

Requisitos de hardware

* El sistema debe correr con un mínimo de 512 MB de RAM.

**Restricciones**

* Para el diseño de la aplicación se seguirá el estándar de diseño Material Design, creado por Google.
* Las librerías externas a utilizar en el desarrollo de la aplicación deben proceder del repositorio oficial de librerías de Flutter [www.pub.dev](http://www.pub.dev).

**Problemas frecuentes**

Control y atención a eventos del usuario

* Que la aplicación esté en proceso de obtención de información desde la base datos y no sea capaz de informar al usuario que está en ese proceso (cargando).

Control de navegación del usuario

* Que el usuario no reconozca dónde se encuentra dentro de la aplicación.
* Que el usuario no pueda volver a un punto anterior de la aplicación.

Implementación de acceso a datos

* En caso de poseer varias bases de datos, que el usuario no pueda seleccionar la base de datos que desee utilizar.
* Que exista gran volumen de datos en la base de datos y la aplicación se demore en devolver la información.

Registro de eventos del sistema (logs)

* Que no sea pueda conocer los detalles de algún error inesperado que se pueda dar en la aplicación, porque no hay una generación de trazas.
* Que exista un sistema de trazas y que este no se pueda desactivar.

Tratamiento de excepciones

* Que falte algún tipo de información en el archivo DB.
* Que falte algún elemento multimedia de los referenciados en el archivo DB.
  1. **Descripción detallada de los Casos de Uso**

A continuación de hace una descripción detallada de los casos de uso implementados en la aplicación según lo acordado en el Plan Individual de Tareas del Estudiante, documento conocido como Anexo A. Acorde con el documento, los casos de uso a implementar son los de reproducción de elementos multimedia y visualización de la información de las especies. Adicionalmente a esto de implementó durante el proceso de desarrollo, la selección de bases de datos y la búsqueda simple de especies.

Tabla 1: Descripción detallada del caso de uso Buscar artículo de especie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de uso | Buscar artículo de una especie | | | |
| Actores | Usuario | | | |
| Descripción | El caso de uso inicia cuando el usuario accede a la pantalla de búsqueda. Mientras que el usuario introduce cada carácter se van mostrando en tiempo real los resultados según el criterio de búsqueda en ese momento en la caja de texto. El caso de uso termina cuando el usuario sale de la pantalla de búsqueda, ya sea porque accedió a algún artículo o se dirigió a la pantalla principal. | | | |
| Requisitos funcionales | RF1: Cuando se realiza una búsqueda el sistema debe mostrar sugerencias de resultados según el criterio de búsqueda en cada instante que se introduzca o de borre un carácter.  RF4: El sistema debe leer la base de datos para mostrar la información contenida en ella. | | | |
| Precondiciones | Base de datos correctamente seleccionada. | | | |
| Casos de uso asociados | | | | |
| Nombre del CU | Tipo de asociación | | Momento en que se asocian | |
| Aplicar filtros de búsqueda (no implementado aún) | extend | | Cuando se muestra la pantalla de búsqueda y el usuario desea restringir los resultados de su búsqueda, este seleccionar los criterios del filtrado que desee aplicar. | |
| Excepciones:   * El proceso se búsqueda se interrumpe por la eliminación o movimiento de la base de datos o elemento de la base de datos por parte de un agente externo en segundo plano, como otra aplicación o por acciones del usuario. | | | | |
| Requisitos de calidad | Apariencia o interfaz externa   * El sistema debe ser simple de usar y tener el comportamiento estándar y sencillo de una pantalla de búsqueda estándar, con su cojón de texto, se sección de filtros u su caja de resultados.   Requisitos de usabilidad   * El sistema debe proveer al usuario la capacidad de documentarse acerca de la pantalla, situando *tooltips* en cada botón.   Requisitos de rendimiento   * El sistema debe tener alta velocidad de procesamiento en el proceso de búsqueda, con un tiempo de respuesta menor a 0.5 s, aunque esto siempre depende de la capacidad de procesamiento del dispositivo móvil en que utilice la aplicación.   Requisitos de seguridad   * El sistema debe no guardar información sensible del usuario, más allá que las últimas búsquedas realizadas. * El sistema debe estar siempre disponible dado que su contenido para búsquedas es *offline*.   Requisitos políticos y culturales   * El sistema debe poseer, para las miniaturas a mostrar, imágenes política y culturalmente correctas para evitar conflictos religiosos, éticos, por costumbres o prejuicios.   Requisitos legales   * El sistema debe respetar las licencias de las imágenes utilizadas como miniaturas en los resultados de búsqueda (Copyright, Creative Commons, Free).   Requisitos de software   * El sistema debe detectar y leer la base de datos externa que posee la información. | | | |
| Postcondiciones | Petición a la base de datos creada. | | | |
| Prototipos de interfaz de usuario: | | | | |
| Criterios de prueba | | | | |
| Valores de entrada | Tipos de datos | Obligatorio | | Respuesta del sistema cuando el campo es nulo o se introducen valores inválidos |
| Criterio de búsqueda | Varchar(255) | No | | En caso de que el campo de búsqueda esté vacío, el sistema muestra los últimos criterios de búsqueda utilizados. |

Tabla 2: Descripción detalla del caso de uso Visualizar artículo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de uso | Visualizar artículo | | | |
| Actores | Usuario | | | |
| Descripción | El caso de uso inicia cuando el usuario selecciona un artículo de los disponibles. En la pantalla del artículo se muestra de una forma potable para el usuario la información extraída de la base de datos con la petición creada anteriormente en la búsqueda. El caso de uso termina cuando el usuario se regresa a la pantalla principal. | | | |
| Requisitos funcionales | RF4: El sistema debe leer la base de datos para mostrar la información contenida en ella.  RF 5: El sistema debe mostrar en el artículo todo el contenido multimedia disponible en la base de datos correspondiente a dicho artículo. | | | |
| Precondiciones | Base de datos correctamente seleccionada. | | | |
| Casos de uso asociados | | | | |
| Nombre del CU | Tipo de asociación | | Momento en que se asocian | |
| Reproducir contenido multimedia | extend | | Cuando se muestra la pantalla de detalles de la especie y el usuario desea ver o reproducir un elemento multimedia de los disponible para esa especie. | |
| Excepciones:  - | | | | |
| Requisitos de calidad | Apariencia o interfaz externa   * El sistema debe ser legible (tener términos no rebuscados y que los tamaños de los elementos visuales y de la fuente sea acorde con los del dispositivo en que se ejecute). * El sistema debe ser simple de usar y tener el comportamiento estándar y sencillo de una aplicación Android estándar. * El sistema debe ser interactivo gracias a la incorporación de imágenes, audios y videos. * El sistema debe tener una apariencia neutra y no enfocarse a nichos muy específicos, con estilo tipo Wikipedia o Kiwix, para garantizar llegar al mismo público.   Rendimiento   * El sistema debe tener alta velocidad de procesamiento, con un tiempo de respuesta menor a 0.5 s, aunque esto siempre depende de la capacidad de procesamiento del dispositivo móvil en que utilice la aplicación.   Seguridad   * El sistema debe estar siempre disponible dado que su contenido es *offline*.   Requisitos políticos y culturales   * El sistema debe poseer contenido política y culturalmente correcto en la base de datos para evitar conflictos religiosos, éticos, por costumbres o prejuicios.   Requisitos legales   * El sistema debe respetar las licencias de los contenidos multimedia empleados (Copyright, Creative Commons, Free). | | | |
| Postcondiciones | - | | | |
| Prototipos de interfaz de usuario: | | | | |
| Criterios de prueba | | | | |
| Valores de entrada | Tipos de datos | Obligatorio | | Respuesta del sistema cuando el campo es nulo o se introducen valores inválidos |
| - | - | - | | - |

Tabla 3: Descripción detalla del caso de uso Reproducir contenido multimedia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de uso | Reproducir contenido multimedia | | | |
| Actores | Usuario | | | |
| Descripción | El caso de uso inicia cuando en la pantalla del artículo el usuario decide reproducir uno de los contenidos multimedia disponible para ese artículo. El caso de uso acaba cuando el usuario termina la reproducción del contenido multimedia. | | | |
| Requisitos funcionales | RF4: El sistema debe leer la base de datos para mostrar la información contenida en ella.  RF 5: El sistema debe mostrar en el artículo todo el contenido multimedia disponible en la base de datos correspondiente a dicho artículo. | | | |
| Precondiciones | Base de datos correctamente seleccionada. | | | |
| Casos de uso asociados | | | | |
| Nombre del CU | Tipo de asociación | | | Momento en que se asocian |
| - | - | | | - |
| Excepciones:  - | | | | |
| Requisitos de calidad | Apariencia o interfaz externa   * El sistema debe ser legible (tener términos no rebuscados y que los tamaños de los elementos visuales y de la fuente sea acorde con los del dispositivo en que se ejecute). * El sistema debe ser simple de usar y tener el comportamiento estándar y sencillo de una aplicación Android estándar.   Rendimiento   * El sistema debe tener alta velocidad de procesamiento, con un tiempo de respuesta menor a 0.5 s, aunque esto siempre depende de la capacidad de procesamiento del dispositivo móvil en que utilice la aplicación.   Seguridad   * El sistema debe estar siempre disponible dado que su contenido es *offline*.   Requisitos políticos y culturales   * El sistema debe poseer contenido política y culturalmente correcto en la base de datos para evitar conflictos religiosos, éticos, por costumbres o prejuicios.   Requisitos legales   * El sistema debe respetar las licencias de los contenidos multimedia empleados (Copyright, Creative Commons, Free). | | | |
| Postcondiciones | - | | | |
| Prototipos de interfaz de usuario: | | | | |
| Criterios de prueba | | | | |
| Valores de entrada | Tipos de datos | Obligatorio | Respuesta del sistema cuando el campo es nulo o se introducen valores inválidos | |
| - | - | - | - | |

Tabla 4: Descripción detalla del caso de uso Seleccionar Base de datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de uso | Seleccionar Base de datos | | | |
| Actores | Usuario | | | |
| Descripción | El caso de uso inicia cuando el usuario entra a la pantalla de selección de bases de datos disponibles. El caso de uso termina cuando el usuario regresa a la pantalla anterior o selecciona una base de datos disponible. | | | |
| Requisitos funcionales | RF 7: El sistema debe proveer al usuario la capacidad de seleccionar la base de datos con la que desee trabajar. | | | |
| Precondiciones | - | | | |
| Casos de uso asociados | | | | |
| Nombre del CU | Tipo de asociación | | | Momento en que se asocian |
| - | - | | | - |
| Excepciones:  - | | | | |
| Requisitos de calidad | Apariencia o interfaz externa   * El sistema debe ser legible (tener términos no rebuscados y que los tamaños de los elementos visuales y de la fuente sea acorde con los del dispositivo en que se ejecute). * El sistema debe ser simple de usar y tener el comportamiento estándar y sencillo de una aplicación Android estándar.   Rendimiento   * El sistema debe tener alta velocidad de procesamiento, con un tiempo de respuesta menor a 0.5 s, aunque esto siempre depende de la capacidad de procesamiento del dispositivo móvil en que utilice la aplicación.   Seguridad   * El sistema debe estar siempre disponible dado que su contenido es *offline*. | | | |
| Postcondiciones | - | | | |
| Prototipos de interfaz de usuario: | | | | |
| Criterios de prueba | | | | |
| Valores de entrada | Tipos de datos | Obligatorio | Respuesta del sistema cuando el campo es nulo o se introducen valores inválidos | |
| - | - | - | - | |

* 1. **Diseño de la solución**

Para la solución de la investigación se ha de crear una aplicación móvil que leerá toda la información que mostrará desde una base de datos relacional.

* + 1. **Diseño de la base de datos**

Como parte de la solución propuesta se ha de utilizar una base de datos SQLite para poder almacenar de forma sencilla el archivo de base de datos en el almacenamiento del dispositivo móvil y facilitar se acceso.

**Modelo lógico de la base de datos**

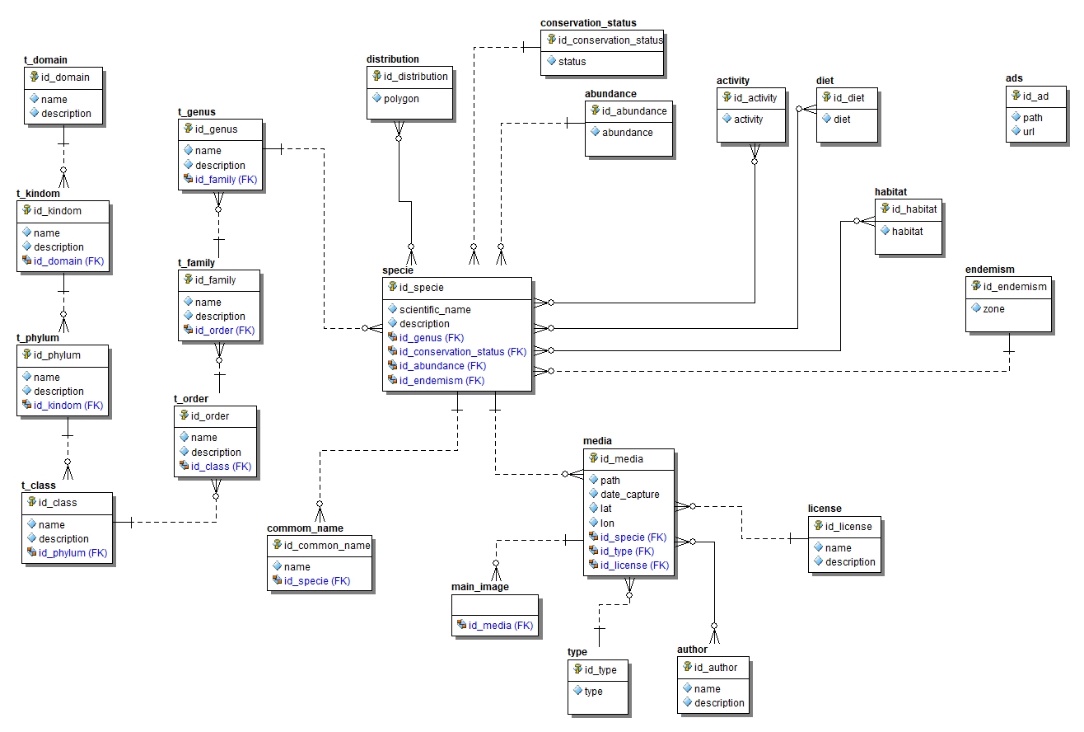


Figura 7: Modelo lógico de la base de datos

**Modelo físico de la base de datos**

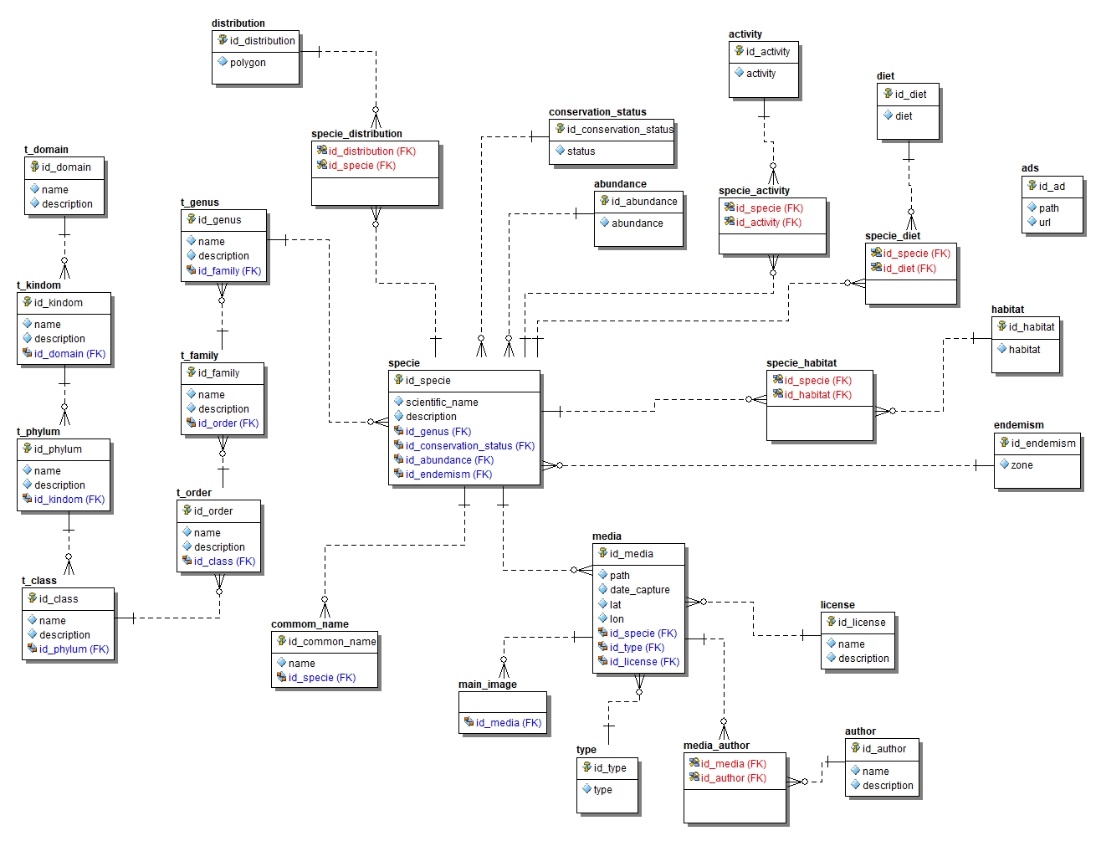


Figura 8: Modelo físico de la base de datos

**Descripción de las tablas**

A continuación se hace una breve descripción de cada tabla del modelo físico anterior.

Tabla 5: Descripciones de las tablas contenidas en la base de datos.

|  |  |
| --- | --- |
| **Nombre de la tabla** | **Descripción** |
| t\_domain | Tabla que representa el dominio como rango de la taxonomía biológica. |
| t\_kindom | Tabla que representa el reino como rango de la taxonomía biológica. |
| t\_phylum | Tabla que representa el filo como rango de la taxonomía biológica. |
| t\_class | Tabla que representa la clase como rango de la taxonomía biológica. |
| t\_order | Tabla que representa el orden como rango de la taxonomía biológica. |
| t\_family | Tabla que representa la familia como rango de la taxonomía biológica. |
| t\_genus | Tabla que representa el género como rango de la taxonomía biológica. |
| specie | Tabla que representa a la especie. |
| common\_name | Tabla que representa los nombres comunes que puede adoptar una especie. |
| distribution | Tabla que representa las zonas en la que se distribuye la especie. |
| specie\_distribution | Tabla que hace de enlace entre las tablas specie y distribution. |
| conservation\_status | Tabla que representa el estado de conservación de una especie. |
| abundance | Tabla que representa la abundancia de una especie. |
| activity | Tabla que representa las actividades de una especie. |
| specie\_activity | Tabla que hace de enlace entre las tablas specie y activity. |
| diet | Tabla que representa las dietas de una especie. |
| specie\_diet | Tabla que hace de enlace entre las tablas specie y diet. |
| habitat | Tabla que representa los hábitats de una especie. |
| specie\_habitat | Tabla que hace de enlace entre las tablas specie y author. |
| endemism | Tabla que representa el lugar del cual una especie es endémica. |
| media | Tabla que lista toda la multimedia relacionada con la especie. |
| main\_image | Tabla que lista si una imagen de una especie es “imagen principal”, o sea, si es imagen de portada para el artículo. |
| type | Tabla que contiene los tipos de multimedia. |
| author | Tabla que contiene los autores de las multimedias. |
| media\_author | Tabla que hace de enlace entre las tablas media y author. |
| license | Tabla que contiene los diferentes tipos de licencias que puede tener un elemento multimedia. |
| ads | Tabla que contiene los anuncios, imágenes y enlacen a Internet. |

* + 1. **Diagrama de clases**

A continuación, en las Figuras 9, 10 y 11, se presenta un diagrama de clases generado automáticamente a partir del código en Dart por la librería dcdg, y por la herramienta PlantUML. La siguiente figura adjuntada está en formato de imagen vectorial, por lo que el acercamiento excesivo no afectará la calidad de visualización.

* 1. **Descripción de la distribución**

Para esta distribución de la aplicación no es necesario equipos potentes en capacidad de procesamiento, con las características mínimas exigidas por Google para la utilización del sistema operativo Android son suficientes, pero es bueno recalcar los requisitos mínimos y recomendados para esta aplicación.

* + 1. **Requisitos mínimos**
* Android 5.0 Lollipop o superior.
* 512 MB de RAM.
* 150 MB de espacio libre en el almacenamiento interno.
  + 1. **Requisitos recomendados**
* Android 5.0 Lollipop o superior.
* 1 GB de RAM.
* 200 MB de espacio libre en el almacenamiento interno.

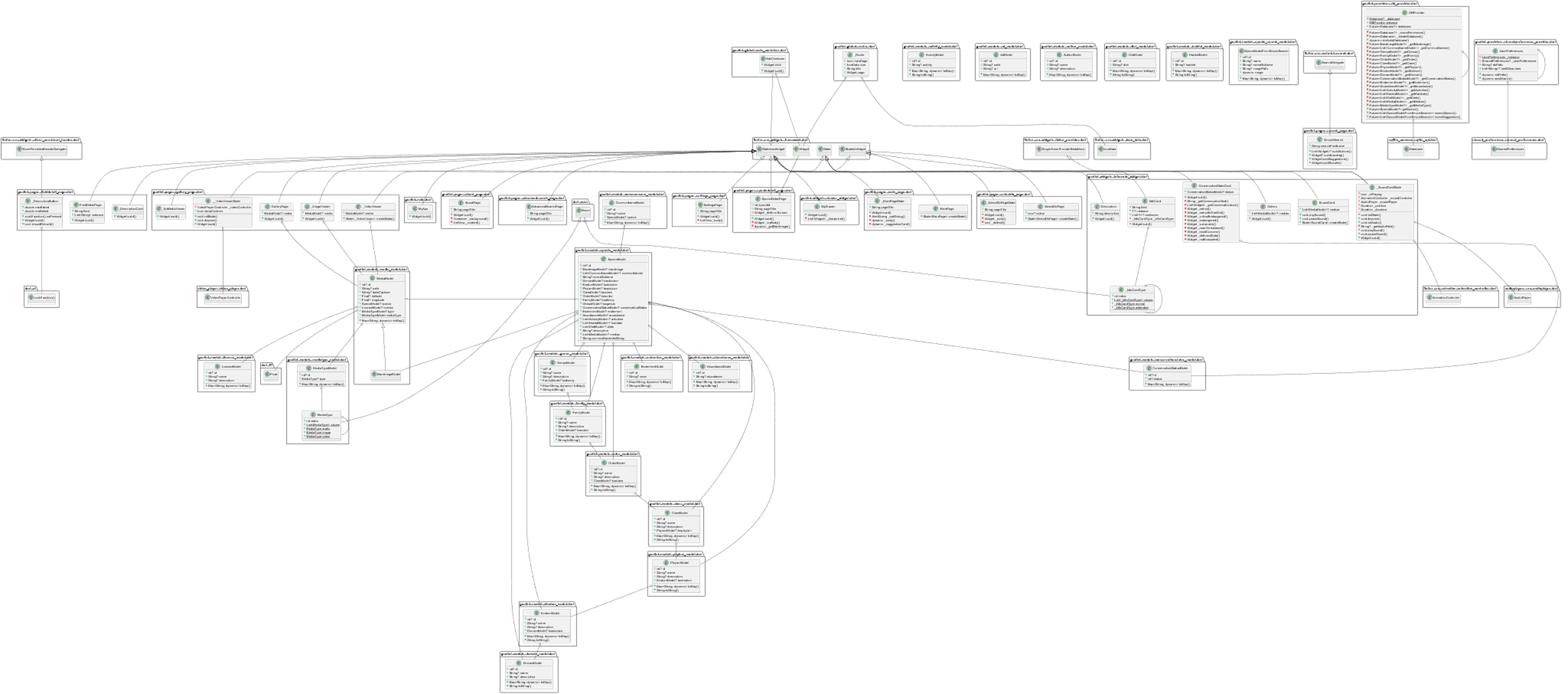


Figura 9: Diagrama de clases. Parte 1

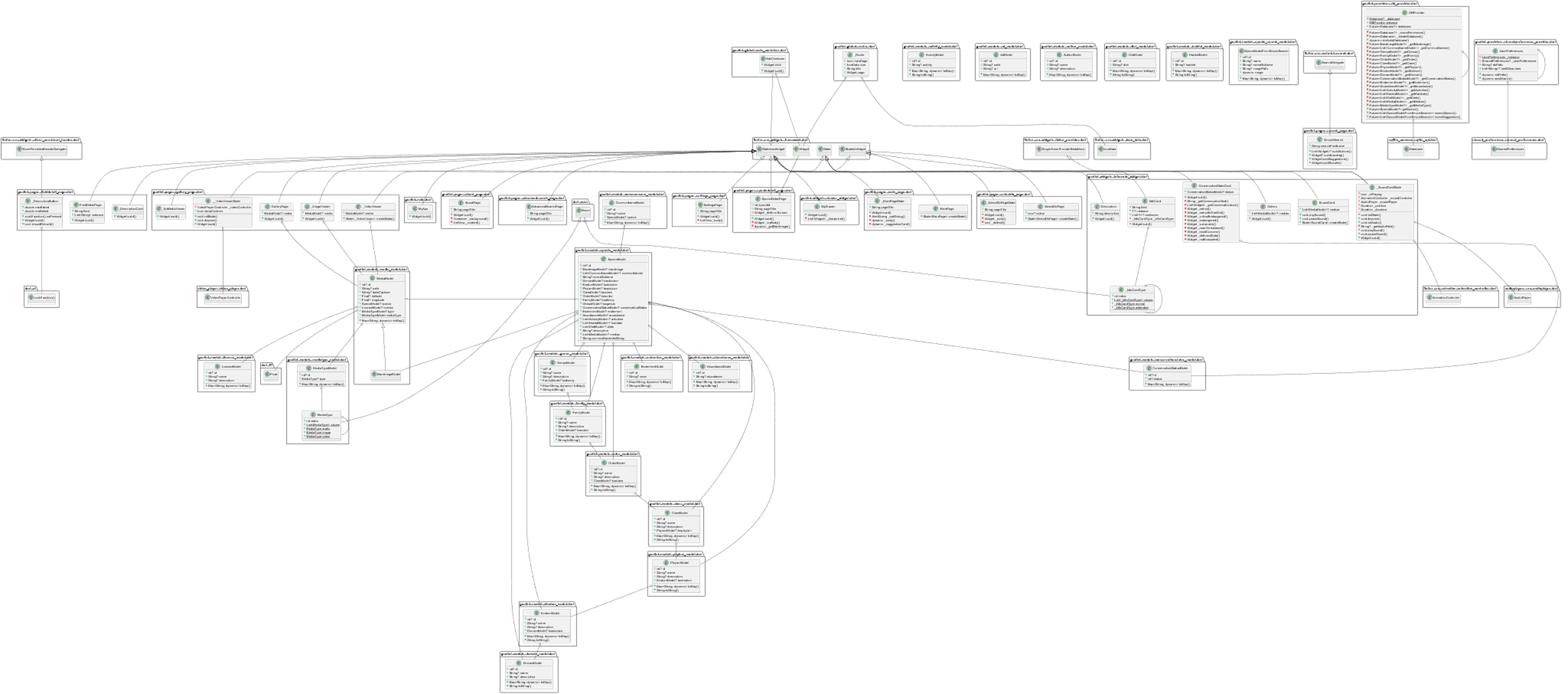


Figura 10: Diagrama de clases. Parte 2

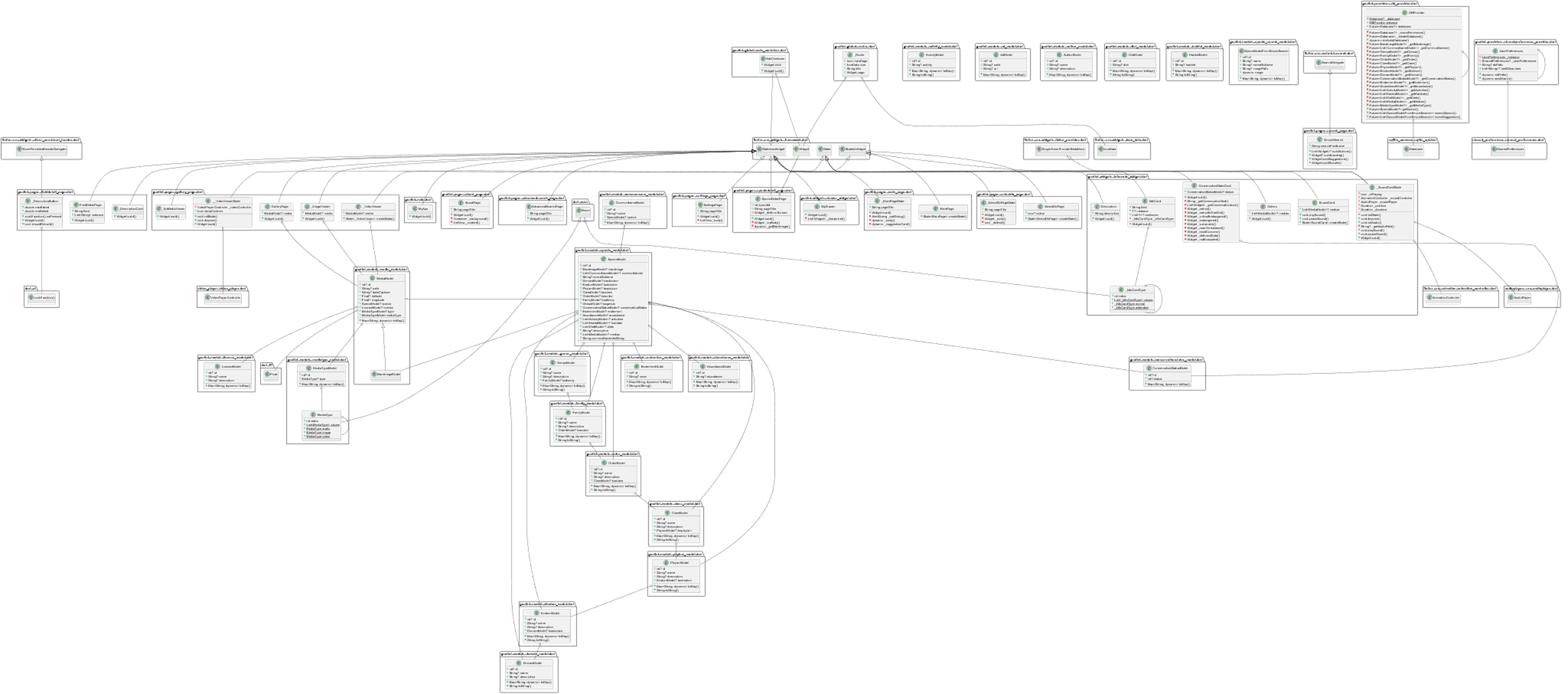


Figura 11: Diagrama de clases. Parte 3

## **Conclusiones parciales**

Según lo analizado en la implementación de la solución propuesta:

* El uso de una base de datos SQLite es ideal para el almacenamiento de la información en el dispositivo móvil.
* La aplicación tiene pocos requisitos de software y hardware para su correcto funcionamiento, por lo que se prevé que pueda ser utilizado en la gran mayoría de dispositivos.

# **Capítulo 3: Validación de la solución**

Para la validación de la solución propuesta se diseñaron pruebas funcionales que permiten validar la eficacia de la aplicación mediante la ejecución y prueba de los requisitos funcionales esenciales.

En el caso siguiente se realizarán pruebas funcionales a los casos de uso de mayor importancia para la aplicación: Buscar artículo de una especie y Visualizar artículo.

* 1. **Prueba funcional: Buscar artículo de una especie**

Tabla 6: Prueba funcional del caso de uso Buscar artículo de una especie

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de prueba** | | **Prueba funcional del caso de uso Buscar artículo de una especie** | |
| **Caso de uso** | | Buscar artículo de una especie | |
| **Funcionalidad** | | Mostrar al usuario los resultados de búsqueda según el criterio de búsqueda introducido. | |
| **Objetivo** | | Probar que se ejecuten correctamente las consultas a la base de datos y que se muestren correctamente los resultados. | |
| **Descripción** | | En este proceso el usuario entra a una pantalla especializada para la búsqueda. El sistema va mostrando con cada carácter introducido el resultado según el criterio ya introducido en el cajón de texto destinado para la búsqueda. | |
| **Criterios de éxito del sistema** | | Se muestra correctamente los resultados al usuario según el criterio de búsqueda introducido. | |
| **Criterios de fallo del sistema** | | Se intenta ejecutar una búsqueda sin criterio de búsqueda introducido. | |
| **Precondiciones** | | Que en la base de datos se encuentre registrado al menos un artículo y se debe hacer al menos una búsqueda. | |
| **Rol** | | Usuario | |
| **Necesidades** | | - | |
| **Programador** | | Damian Aldair Garcia Pérez | |
| **Probador** | | Damian Aldair Garcia Pérez | |
| **Fecha** | | 2022-07-01 | |
| **Condición de ejecución 1:** Verificar que solo se muestre un historial de búsqueda al no haber criterio de búsqueda. | | | |
| **Entradas de prueba** | | Criterio de búsqueda: (vacío) | |
| **Puntos de control** | | Si se pulsa en botón Atrás, se cierra la pantalla y se vuelva a la pantalla inicial. | |
| **Resultados esperados** | | * Si se han realizado búsquedas anteriormente se muestran los últimos 5 criterios de búsqueda utilizados. | |
| **Flujo del caso de prueba** | **#** | **Usuario de sistema** | **Sistema** |
| **1** | Pulsa el botón de búsqueda en la pantalla principal. | Abre la pantalla de búsqueda. |
| **2** | Dejar el cuadro de texto para el criterio de búsqueda en blanco. | Se muestran los últimos 5 criterios de búsqueda utilizados. |
| **Condición de ejecución 2:** Verificar que se muestren correctamente los resultados de búsqueda | | | |
| **Entradas de prueba** | | Criterio de búsqueda: “tocor”. | |
| **Puntos de control** | | Si se pulsa en botón Atrás, se cierra la pantalla y se vuelva a la pantalla inicial. | |
| **Resultados esperados** | | * Se muestran todos los artículos de especies que concuerden con el criterio de búsqueda. | |
| **Flujo del caso de prueba** | **#** | **Usuario de sistema** | **Sistema** |
| **1** | Pulsa el botón de búsqueda en la pantalla principal. | Abre la pantalla de búsqueda. |
| **2** | Escribir en el cuadro de texto para el criterio de búsqueda “tocor”. | Se muestra los artículos relacionados con ese criterio de búsqueda. |

* 1. **Prueba funcional: Visualizar artículo**

Tabla 7: Prueba funcional del caso de uso Visualizar artículo

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de prueba** | | **Prueba funcional del caso de uso Visualizar artículo** | |
| **Caso de uso** | | Visualizar artículo | |
| **Funcionalidad** | | Mostrar al usuario el contenido del artículo seleccionado. | |
| **Objetivo** | | Probar que se ejecuten correctamente la consulta a la base de datos y que se muestre correctamente el contenido del artículo. | |
| **Descripción** | | En este proceso el usuario selecciona un artículo para ver. El sistema internamente solicita todos los detalles del artículo por el id de este. | |
| **Criterios de éxito del sistema** | | Se muestra correctamente todo el contenido del artículo seleccionado. | |
| **Criterios de fallo del sistema** | | Se intenta mostrar un artículo de un id que no existe. | |
| **Precondiciones** | | Que en la base de datos se encuentre registrado al menos un artículo. | |
| **Rol** | | Usuario | |
| **Necesidades** | | - | |
| **Programador** | | Damian Aldair Garcia Pérez | |
| **Probador** | | Damian Aldair Garcia Pérez | |
| **Fecha** | | 2022-07-01 | |
| **Condición de ejecución 1:** Verificar que se muestre correctamente el artículo seleccionado. | | | |
| **Entradas de prueba** | | Id de artículo: 1 | |
| **Puntos de control** | | Si se pulsa en botón Atrás, se cierra la pantalla y se vuelva a la pantalla inicial. | |
| **Resultados esperados** | | * Se muestra al usuario el artículo con todos los detalles de este. | |
| **Flujo del caso de prueba** | **#** | **Usuario de sistema** | **Sistema** |
| **1** | Pulsa sobre un artículo para ver más detalles de este. | Abre la pantalla de visualización del artículo mostrando todos los detalles de este. |

* 1. **Resultados de las pruebas**

Las pruebas diseñadas devolvieron los resultados esperados y favorables en todos los casos. Esto permite validar el correcto funcionamiento de la aplicación para los casos de uso Buscar artículo de una especie y Visualizar artículo. Por lo tanto se comprueba que el sistema cumple con las funcionalidades de:

* Buscar artículos según el criterio de búsqueda introducido por el usuario.
* Mostrar todos los detalles de un artículo seleccionado.

## **Conclusiones parciales**

Con la validación a la solución propuesta se puede concluir que:

* El diseño y ejecución de las pruebas permitieron verificar el correcto funcionamiento de la aplicación.
* La aplicación es capaz de mostrar correctamente información de cualquier ser vivo.

# **Conclusiones**

Tras el análisis del marco teórico, la implementación de la solución propuesta y su validación, se puede llegar a la conclusión que:

* iOS y Android son los sistemas operativos móviles más usados mundialmente y en Cuba, por lo que los esfuerzos deben ir es esa dirección.
* Flutter es el framework idóneo para el desarrollo rápido de las aplicaciones y lograr el factor multiplataforma junto con el rendimiento y el atractivo.
* Las bases de datos SQLite es la alternativa que se busca para lograr un buen almacenamiento y administración de los datos de la aplicación.
* La identificación de requisitos ayudó en la implementación de la solución.
* Aún se necesita implementar más características para hacer la aplicación más completa.
* Aún se necesita publicar la aplicación en la tienda cubana de aplicaciones Apklis, para una mejor promoción del proyecto Guatiní.
* Aun se necesita implementar el idioma inglés dentro de la aplicación.

# **Recomendaciones**

Para trabajos posteriores se recomienda:

* Habilitar todo el framework de Flutter en un ordenador con el sistema operativo MacOS, para lograr la compilación de la aplicación en formato IPA para su instalación los dispositivos iPhone e IPad.

# **Referencias bibliográficas**

[1] J. C. Sepúlveda Peña *et al.*, "Guatiní: un proyecto para fomentar el conocimiento de la avifauna cubana en las nuevas generaciones," *Ciencia y Agricultura,* vol. 16, no. 1, pp. 17-30, 2019, doi: <https://doi.org/10.19053/01228420.v16.n1.2019.8803>.

[2] F. Wikimedia. "Wikipedia (aplicación)." <https://es.wikipedia.org/wiki/Wikipedia_(aplicaci%C3%B3n>) (accessed jun. 29, 2022).

[3] K. Association. "Kiwix lets you access free knowledge - even offline." <https://www.kiwix.org/en/> (accessed jun. 29, 2022).

[4] EcuRed. "EcuMóvil." <https://www.ecured.cu/EcuM%C3%B3vil> (accessed jun. 29, 2022).

[5] L. Ardito, R. Coppola, G. Malnati, and M. Torchiano, "Effectiveness of Kotlin vs. Java in Android App Development Tasks," *Information and Software Technology,* vol. 127, 2020.

[6] C. González García, J. Pascual Espada, B. C. Pelayo García-Bustelo, and J. M. Cueva Lovelle, "Swift vs. Objective-C: A New Programming Language," *International Journal of Artificial Intelligence and Interactive Multimedia,* vol. 3, no. 3, pp. 74-81, 2015, doi: 10.9781/ijimai.2015.3310.

[7] P. Nawrocki, K. Wrona, M. Marczak, and B. Sniezynski, "A comparison of native and cross-platform frameworks for mobile applications," *Computer,* vol. 54, no. 3, pp. 18-27, 2021.

[8] Ionic. "Cross-Platform Mobile App Development: Ionic Framework." <https://ionicframework.com/> (accessed jun. 30, 2022).

[9] M. Platforms. "React Native - Leran Once, write anywhere." <https://reactnative.dev/> (accessed jun. 30, 2022).

[10] Microsoft. "Xamarin | Open-source mobile app platform for .NET." <https://dotnet.microsoft.com/en-us/apps/xamarin> (accessed jun. 30, 2022).

[11] Flutter. "Flutter - Build apps for any screen." <https://flutter.dev/> (accessed jun. 30, 2022).

[12] M. L. Napoli, *Beginning flutter: a hands on guide to app development*. John Wiley & Sons, 2019.

[13] G. Inc. "Gradle Build Tool." <https://gradle.org/> (accessed jun. 1, 2022).

[14] Google. "Herramientas de línea de comandos | Desarrolladores de Android | Android Developers." <https://developer.android.com/studio/command-line> (accessed jun. 1, 2022).

[15] Oracle. "Home. Java Platform, Standard Edition (Java SE) 8 Release 8." <https://docs.oracle.com/javase/8/> (accessed jun. 1, 2022).

[16] Flutter. "Flutter documentation | Flutter." <https://docs.flutter.dev/> (accessed jun. 30, 2022).

[17] Dart. "Dart documentation | Dart." <https://dart.dev/guides> (accessed jun. 30, 2022).

[18] Google. "Design - Material Design." <https://material.io/design> (accessed jun. 30, 2022).

[19] D. Alcalde Sánchez, "Desarrollo de una Aplicación Didáctica Multiplataforma Reutilizable," Tesis de Diploma para optar por el título de Ingeniería Informática, Facultad de Ingeniería Informática, Universidad Tecnológica de La Habana “José Antonio Echeverría”, La Habana, 2021.

[20] K. Hernández Albor, "Guatiní; multimedia sobre aves endémicas de Cuba," Tesis de Diploma para optar por el título de Ingeniería Informáica, Facultad de Ingeniería Informáica, Universidad Tecnológica de La Habana “Jose Antonio Echeverría”, La Habana, 2021.

[21] C. A. Villanueva Hedesa, "Proyecto Guatiní: Herramienta para la difusión de la información sobre las características de la fauna cubana: enciclopedia.," Trabajo de diploma, Facultad de Ingeniería Informática, Universidad Tecnológica de La Habana “José Antonio Echeverría”, La Habana, 2019.

# **Anexos**



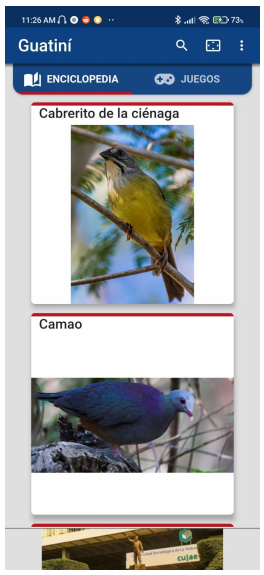
1. Captura de pantalla de la aplicación Wikipedia.



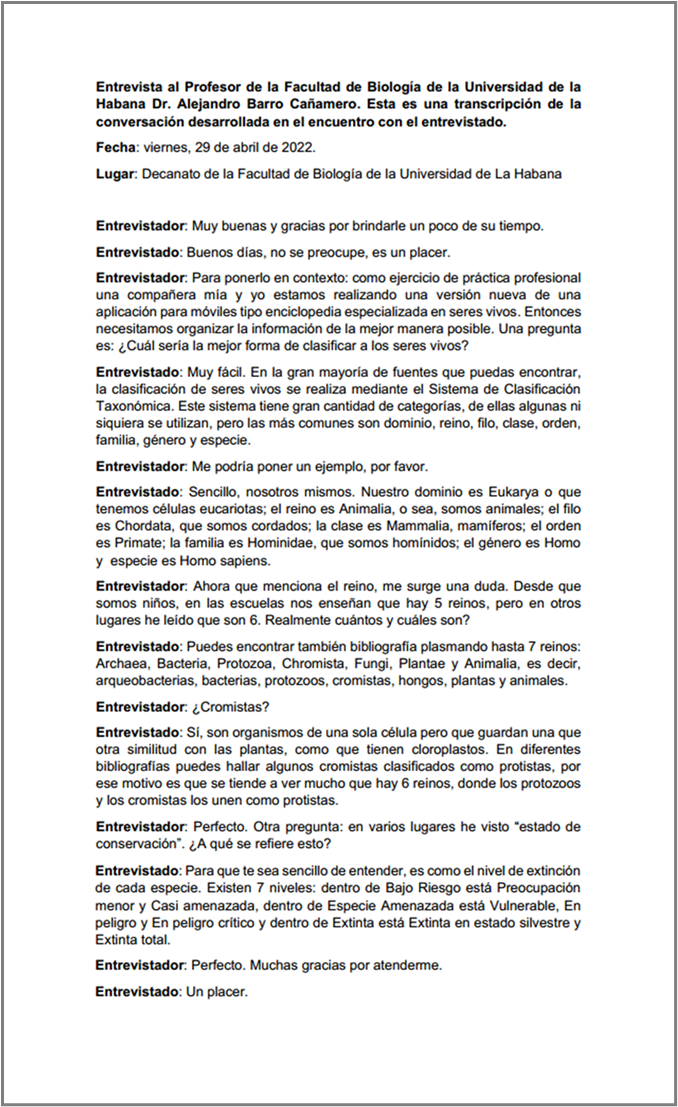
1. Captura de pantalla de la aplicación Kiwix.



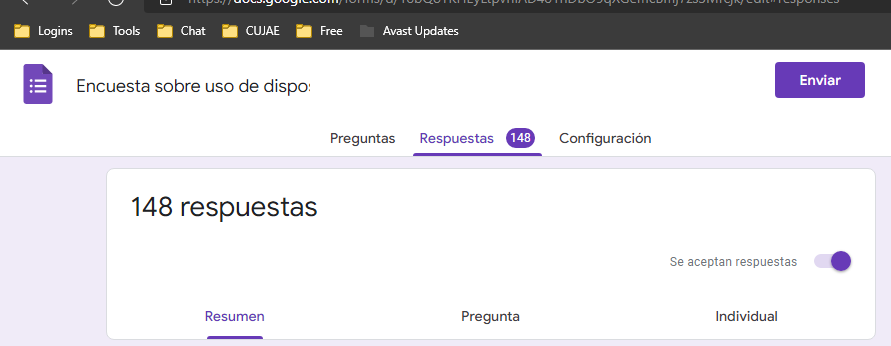
1. Vista de un artículo en EcuMóvil en la versión no basada en el proyecto Kiwix [4].



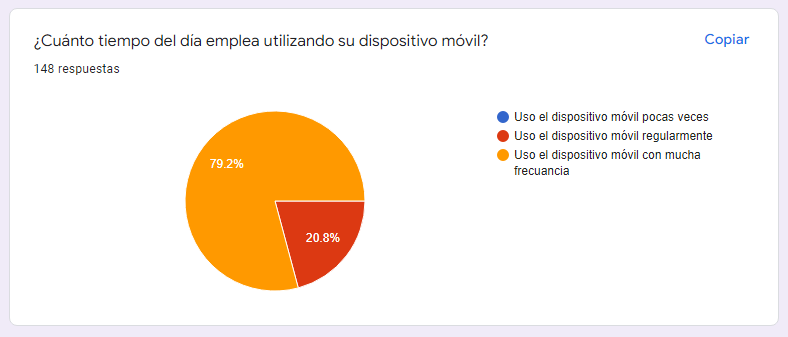
1. Pantalla inicial de la versión existente de Guatiní [19].



1. Captura de entrevista realizada al profesor de la Facultad de Biología de la Universidad de La Habana Dr. Alejandro Barro Cañamero.



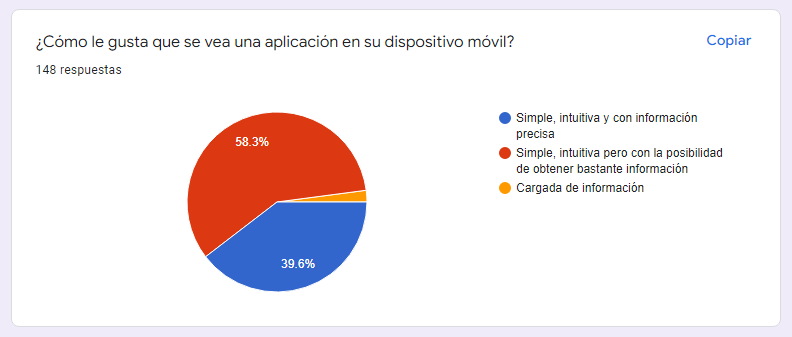
1. Total de personas que participaron en una encuesta realizada mediante la herramienta web Formularios de Google. (Captura de pantalla de https://docs.google.com/forms/)



1. Gráfico de tiempo de uso del dispositivo móvil por parte de las personas. (Captura de pantalla de https://docs.google.com/forms/)



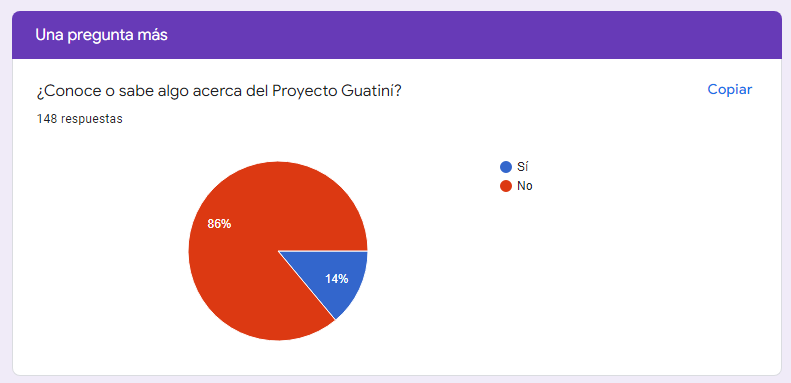
1. Gráfico de nivel de vinculación del dispositivo móvil a las tareas cotidianas. (Captura de pantalla de https://docs.google.com/forms/)



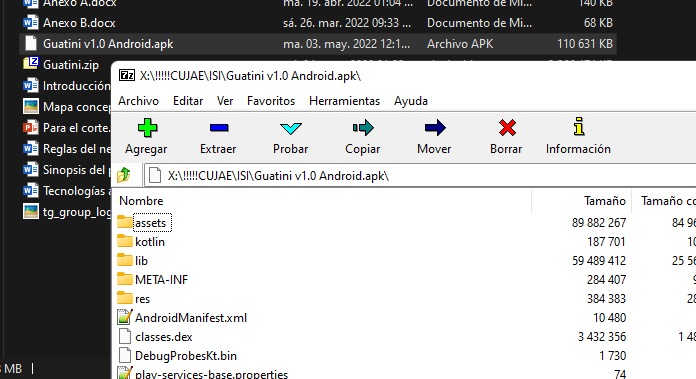
1. Gráfico de gustos de nivel de información mostrado en una aplicación. (Captura de pantalla de https://docs.google.com/forms/)



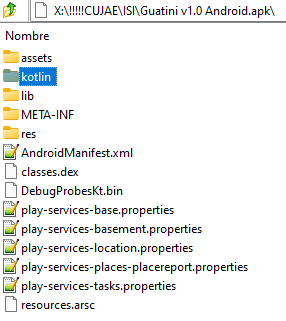
1. Gráfico de preferencias en el modo de buscar información. (Captura de pantalla de https://docs.google.com/forms/)



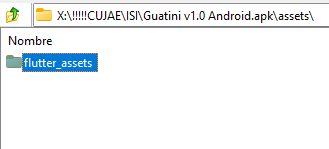
1. Gráfico de nivel de difusión del Proyecto Guatiní. (Captura de pantalla de https://docs.google.com/forms/)



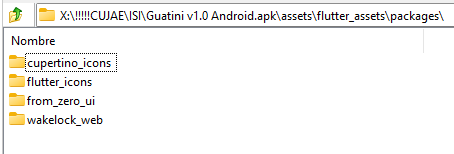
1. Captura de pantalla del interior del archivo APK abierto con 7-zip



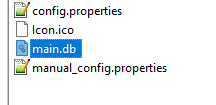
1. Captura de pantalla 1 de la aplicación móvil abierta con 7-zip



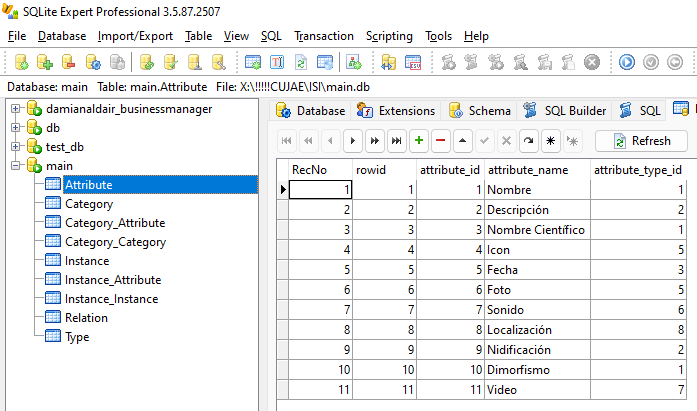
1. Captura de pantalla 2 de la aplicación móvil abierta con 7-zip



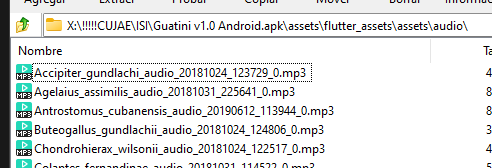
1. Captura de pantalla 3 de la aplicación móvil abierta con 7-zip



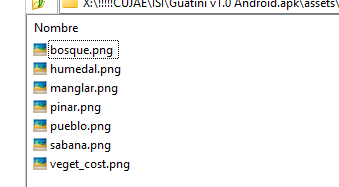
1. Captura de pantalla 4 de la aplicación móvil abierta con 7-zip



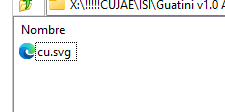
1. Captura de pantalla de la base de datos de la aplicación móvil abierta con SQLite Expert.



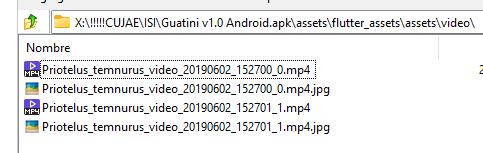
1. Captura de pantalla 5 de la aplicación móvil abierta con 7-zip



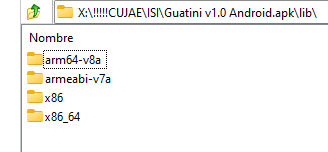
1. Captura de pantalla 6 de la aplicación móvil abierta con 7-zip



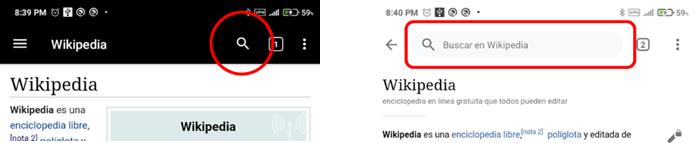
1. Captura de pantalla 7 de la aplicación móvil abierta con 7-zip



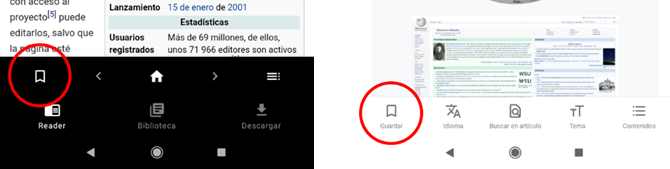
1. Captura de pantalla 8 de la aplicación móvil abierta con 7-zip



1. Captura de pantalla 9 de la aplicación móvil abierta con 7-zip



1. Captura de pantalla del botón y barra de búsqueda de Kiwix y Wikipedia, respectivamente.



1. Captura de pantalla del botón de favoritos en Kiwix y Wikipedia, respectivamente.



1. Captura de pantalla del botón para buscar en el artículo en Wikipedia.



1. Captura de pantalla de la tabla resumen con las características importantes reflejadas en el artículo

1. Software para wikis libre programado en el lenguaje PHP. [↑](#footnote-ref-1)