



UNIVERSITAT DE
BARCELONA



Universitat
Pompeu Fabra
Barcelona

Acercando la fauna chilena a la comunidad a través de tecnología móvil

Joan Valenzuela Rubilar

Tutor: Rubén Alcaraz Martínez

Curso 2017-2018

Máster en Gestión de Contenidos Digitales

Facultad de Biblioteconomía y Documentación

Joan Valenzuela Rubilar
Enero 2018.



Los contenidos de este documento están sujetos a una licencia Creative Commons Reconocimiento - No Comercial - Compartir Igual 4.0.

Usted es libre de copiar, distribuir, comunicar públicamente y transformar esta obra, siempre y cuando reconozca los créditos de la obra a Joan Valenzuela Rubilar y al Máster en Gestión de Contenidos Digitales UB/UPF, no utilice la obra para fines comerciales y la comparta bajo la misma licencia si altera o modifica su contenido. Puede consultar el texto completo de la licencia en:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Agradecimientos

A Rubén Alcaraz Martínez, profesor de la Facultad de Biblioteconomía y Documentación de la Universidad de Barcelona y tutor de este trabajo.

A mi familia y amigos, por estar siempre presentes a pesar de la distancia.

Resumen

Objetivos: explorar las diferentes posibilidades de desarrollo de aplicaciones móviles, así como las herramientas y tecnologías implicadas en este proceso, con el fin de generar una aplicación sobre fauna chilena, de carácter divulgativo, para fomentar el estudio y concientizar a la ciudadanía sobre la conservación de las especies animales y la protección del hábitat en el que coexisten.

Metodología: desarrollo informático de una prueba de concepto consistente en una aplicación móvil híbrida sobre fauna chilena, con un enfoque divulgativo. Lo anterior a partir del estudio de iniciativas similares, en base a características y condicionantes propios del diseño de interfaces para dispositivos móviles y luego de la valoración de las diferentes aproximaciones para la distribución de contenidos en dispositivos móviles.

Resultados: se ha obtenido un prototipo de aplicación móvil enfocada en los animales vertebrados de Chile, que permite categorizar diferentes grupos de animales, realizar búsquedas sobre las especies incluidas en la aplicación y acceder a una ficha informativa sobre estas. Es necesario señalar que la base de diseño y programación de esta aplicación puede ser reutilizada en iniciativas similares sobre otros grupos de especies.

Palabras clave: Dispositivos móviles, aplicaciones móviles, JQuery Mobile, Fauna chilena

Abstract

Objectives: To explore the different possibilities for developing mobile applications as well as the tools and technologies involved in this process, in order to generate an application about Chilean fauna, with an informative character, to encourage the study and raise awareness between citizens about the conservation of animal species and the protection of the habitat they coexist in.

Methodology: Computing development of a proof of concept that consist in a hybrid mobile application about Chilean fauna with an informative focus. This from the study of similar initiatives based on characteristics and conditions typical of interface design for mobile devices and then on the valuation of the different approximations for the distribution of contents in mobile devices.

Results: A prototype of mobile applications focused on vertebrate animals has been obtained. This prototype permits to categorize different groups of animals, to search about the species contained in the application, and to have access to a fact sheet of the different species. It is important to say that the base for the design and programming of this application can be reused for similar initiatives about other groups of species.

Key words: Mobile devices, mobile applications, JQuery Mobile, Chilean fauna

Sumario

1.	Resumen ejecutivo.....	1
2.	Alcance y objetivos	3
2.1.	Definición	3
2.2.	Alcance	3
2.3.	Objetivos	3
3.	Justificación.....	4
3.1.	Protección y conservación de la fauna chilena	4
3.2.	Contexto tecnológico	5
3.3.	Aplicación móvil sobre fauna chilena	6
4.	Marco teórico.....	8
4.1.	Escenario móvil actual	8
4.2.	Dispositivos móviles.....	11
4.3.	Aplicaciones móviles.....	15
4.3.1.	Lenguajes de programación.....	18
4.3.2.	<i>Frameworks</i> para el desarrollo móvil multiplataforma	24
4.4.	Decisión de desarrollo.....	29
5.	Análisis de aplicaciones similares	30
5.1.	Panorama internacional.....	30
5.1.1.	Field Guide apps to Australian Fauna	30
5.1.2.	Guía de aves e itinerarios ornitológicos de España	33
5.2.	Panorama nacional	37
5.3.	Tendencias respecto a la identificación de especies	39
6.	Desarrollo de la aplicación.....	41
6.1.	Diseño	41
6.1.1.	Arquitectura de la información.....	41
6.1.2.	Usabilidad.....	51
6.1.3.	Patrones de diseño	54
6.2.	Implementación.....	64
6.2.1.	Elementos transversales al desarrollo	64
6.2.2.	Pantallas de la aplicación	66

6.3.	Optimización en tiendas de aplicaciones.....	76
6.3.1.	Definición de ASO	76
6.3.2.	Factores ASO	76
6.3.3.	Propuesta ficha para Google Play	79
7.	Planificación y presupuesto	80
7.1.	Escenario ficticio	80
7.2.	Recursos humanos necesarios y etapas de implementación	80
7.2.1.	Recursos humanos	80
7.2.2.	Fases de implementación	81
7.2.3.	Cronograma	85
7.2.4.	Presupuesto	87
8.	Conclusiones y trabajo futuro.....	89
8.1.	Conclusiones	89
8.2.	Trabajo futuro	90
9.	Bibliografía	91

1. Resumen ejecutivo

La geografía particular de Chile ha permitido el desarrollo de ecosistemas naturales inmensamente ricos en biodiversidad, con especies animales únicas en el planeta. Lo anterior, hace necesaria la búsqueda e implementación de iniciativas que permitan poner al alcance de la ciudadanía información referente a la fauna del país, con el fin de generar conciencia y contribuir a su conservación y protección. La masificación de los dispositivos móviles durante los últimos años hace de la tecnología móvil una alternativa válida para distribuir este tipo de contenidos. Lo anterior, teniendo en cuenta las considerables prestaciones técnicas de *smartphones* y *tablets*, similares a las de los ordenadores menos potentes, y que, junto a factores como la portabilidad y la posibilidad de una conexión permanente a Internet, permiten a los usuarios de dispositivos móviles llevar a cabo sus actividades habituales prácticamente desde cualquier lugar.

Este acceso masivo a los dispositivos móviles ha provocado un incremento exponencial en el uso de aplicaciones móviles. Una aplicación móvil es un software con funcionalidades muy específicas, diseñado para ejecutarse en un sistema operativo móvil determinado, y cuyo objetivo es que el usuario pueda realizar una tarea concreta como, por ejemplo, gestionar su correo electrónico. Al realizar determinadas búsquedas clave en las tiendas de aplicaciones de Android y iOS, es posible evidenciar la ausencia de una aplicación móvil sobre fauna chilena, por lo que este proyecto busca explorar las diferentes posibilidades de desarrollo de aplicaciones, con el fin de generar una primera versión prototipo que permita, en un futuro, suplir la necesidad informativa antes señalada. En este punto, es necesario considerar que, en otras regiones geográficas del planeta ya se están implementando proyectos relevantes de tecnología móvil relacionados con la identificación y preservación de fauna. Entre ellos destacan, la iniciativa Field Guide apps to Australian Fauna del Gobierno de Australia en conjunto con los museos más importantes del país y el proyecto Guía de las aves e itinerarios ornitológicos de España, desarrollado en conjunto por la Sociedad Española de Ornitología y la Fundación BBVA.

En relación con las posibilidades de desarrollo de aplicaciones móviles, es preciso mencionar la existencia de tres categorías de construcción: 1) aplicaciones web, que corresponden esencialmente a sitios web optimizados para su consulta desde dispositivos móviles, desarrollados en base a estándares de formato abierto impulsados por el W3C (HTML, CSS y JavaScript) 2) aplicaciones nativas desarrolladas para un sistema operativo específico utilizando el lenguaje de programación propio de cada plataforma, por ejemplo, Java para Android y Swift u Objective-C en el caso iOS. Es preciso mencionar que las aplicaciones nativas se adaptan totalmente a las funcionalidades y características del dispositivo 3) aplicaciones híbridas que se construyen mediante tecnologías propias del desarrollo web junto a la utilización de un *framework*, el cual corresponde a un esquema que permite agilizar y “estandarizar” el proceso de desarrollo de software. Las aplicaciones híbridas poseen una mayor versatilidad que las aplicaciones web al momento de acceder a los sensores del

dispositivo y también permiten obtener un aspecto y comportamiento semejante al de las aplicaciones nativas. Las razones antes señaladas, junto a un tiempo de desarrollo menor al del desarrollo nativo hacen de las aplicaciones híbridas una opción ideal para llevar a cabo una prueba de concepto sobre la aplicación antes descrita, considerando también que este tipo de desarrollo deja abierta la posibilidad de generar una versión para otros SO, de manera mucho más rápida que en el caso de tener que crear una versión nativa. Es necesario señalar que este proyecto se centrará en el desarrollo de una aplicación móvil para el SO Android, ya que es el que actualmente domina el mercado móvil en Latinoamérica con un 81% de penetración (IMS Corporate, 2016). Con respecto al *framework* utilizado para llevar a cabo esta prueba de concepto, luego de valorar distintas opciones, se optó por el *framework* de código abierto JQuery Mobile, principalmente porque, a nivel de diseño, se basa en una cuadrícula flexible, lo que facilita la implementación de un *Responsive Web Design*, e incluye una serie de componentes gráficos reutilizables y personalizables que agilizan la construcción de aplicaciones móviles; a nivel de programación, facilita el acceso a los elementos de un documento HTML mediante el uso de selectores, incorpora eventos específicos para dispositivos móviles y permite la utilización de AJAX.

Desarrollar una aplicación móvil, ya sea un prototipo o una versión definitiva, implica llevar a cabo a cabo una etapa de diseño que, por una parte, permita definir la presentación de los elementos y contenidos al usuario visitante, con el fin de satisfacer de manera efectiva sus necesidades de información (arquitectura de la información) y por otra parte posibilite la creación de la interfaz gráfica de la aplicación en base a la facilidad de interacción del usuario (usabilidad). Lo anterior considerando que ambos factores repercutirán de manera directa en la percepción del usuario, respecto a la aplicación desarrollada (experiencia de usuario). Con respecto a la experiencia de usuario, se consideraron para el diseño de la aplicación, tanto el comportamiento del usuario y las secuencias de acciones que puede realizar (diseño de interacción) como la disposición de los elementos visuales en pantalla y la estética de la interfaz gráfica (diseño de interfaz). En este sentido, es necesario mencionar que, cada sistema operativo móvil posee una identidad gráfica y funcional propia, planteando formas específicas de diseñar los elementos de una interfaz gráfica y de interactuar con el dispositivo móvil. Para esta implementación, se consideraron las pautas ofrecidas de Material Design Ofrecidas por Google, ya que es lo que los usuarios esperarían encontrar en una interfaz Android.

Publicar una aplicación en la tienda de aplicaciones móviles correspondiente, implica implementar algún tipo de estrategia que permita que la aplicación aparezca en los primeros puestos de las páginas de resultados. Para lo cual es necesario trabajar una serie de factores que influyen directamente en el posicionamiento de aplicaciones. Es por lo anterior que en este trabajo también se presenta una propuesta de ficha Google Play para la aplicación móvil sobre fauna chilena.

Por último, se desarrolla la planificación cronológica y el coste monetario del proyecto en el marco de un escenario ficticio.

2. Alcance y objetivos

2.1. Definición

El proyecto consiste en explorar las diferentes posibilidades de desarrollo de aplicaciones móviles, así como las herramientas y tecnologías implicadas en este proceso, con el fin de generar una aplicación, de carácter divulgativo, sobre fauna chilena, para fomentar el estudio y concientizar a la ciudadanía sobre la conservación de las especies animales y la protección del hábitat en el que coexisten.

2.2. Alcance

Si bien se trata de un proyecto esencialmente indagatorio, se pretende llevar a cabo el desarrollo de un prototipo en fase beta que permita poner a prueba parte de la información recabada durante la investigación.

2.3. Objetivos

Los objetivos del proyecto son los siguientes:

- Contextualizar acerca del uso de dispositivos y aplicaciones móviles en Chile y el mundo.
- Presentar y valorar las diferentes aproximaciones a la hora de distribuir contenidos a través de dispositivos móviles (aplicaciones web, nativas e híbridas).
- Analizar el estado actual y las tendencias en cuanto a la identificación de especies de flora y fauna, mediante aplicaciones móviles.
- Revisar la literatura científica sobre el diseño de interfaces de aplicaciones para dispositivos móviles con el objetivo de conocer las características y condicionantes propios a tener en cuenta.
- Desarrollar una prueba de concepto (prototipo) de la aplicación descrita, en base a las características y condicionantes del diseño de interfaces móviles.
- Realizar una propuesta de ficha de la aplicación para su distribución en la tienda de aplicaciones de Google Play.

3. Justificación

3.1. Protección y conservación de la fauna chilena

Así como las obras arquitectónicas, la escultura, la pintura y otras expresiones humanas forman parte del patrimonio cultural, los ecosistemas naturales y la flora y fauna contenida en ellos constituyen el patrimonio natural de la humanidad, tienen un valor universal y es preciso generar conciencia sobre su protección y conservación. Chile es el país más largo y estrecho del mundo¹, con una longitud de más de 4.300 km de norte a sur y un ancho promedio de 180 km. La extensa longitud del país da pie a diversos climas y geografías, encontrándose, por ejemplo, en el norte el desierto de Atacama, el más árido del mundo, y en el extremo sur innumerables glaciares, islas y fiordos. Esta geografía tan particular ha permitido el desarrollo de ecosistemas naturales inmensamente ricos en biodiversidad. Ahora bien, políticamente, Chile se divide en 16 regiones, las que a su vez pueden ser agrupadas en 5 macro-zonas geográficas: Norte Grande, Norte Chico, Zona Centro, Zona Sur y Zona Austral, donde cada una posee especies animales únicas en el planeta como el cucuro² o el zorro chilote³.

En Chile, es el Ministerio del Medio Ambiente (en adelante, MMA) la entidad gubernamental encargada del diseño y de la ejecución de políticas, planes y programas vinculados con la protección del medio ambiente y de la diversidad biológica y de los recursos naturales existentes en el país. Según el MMA existen alrededor de 33.000 especies silvestres nativas (que se originaron o llegaron naturalmente al país, sin la intervención del hombre), tanto de flora como de fauna, que habitan el país. Dentro del MMA es, específicamente, la División de Recursos Naturales y Biodiversidad⁴ el área encargada de la conservación y la protección de la biodiversidad y el patrimonio ambiental.

Además de lo anterior, la legislación vigente en Chile también pretende contribuir a la regulación de la relación entre la ciudadanía y los animales. Es así como la Ley número 20.280 Sobre Protección de Animales⁵, publicada en Chile en octubre de 2009, “establece normas destinadas a conocer, proteger y respetar a los animales, como seres vivos y parte de la naturaleza, con el fin de darles un trato adecuado y evitarles sufrimientos innecesarios.”

¹ Universidad de Chile. *Acerca de Chile: presentación territorial*. <<http://www.uchile.cl/portal/presentacion/la-u-y-chile/acerca-de-chile/8035/presentacion-territorial>>. [Consulta: 11/10/2017].

² MMA. *Spalacopus cyanus*. <http://especies.mma.gob.cl/CNMWeb/Web/WebCiudadana/ficha_indepen.aspx?EspeciId=772>. [Consulta: 12/10/2017].

³ MMA. *Lycalopex fulvipes*. <http://especies.mma.gob.cl/CNMWeb/Web/WebCiudadana/ficha_indepen.aspx?EspeciId=16>. [Consulta: 12/10/2017].

⁴ MMA. *División de Recursos Naturales y Biodiversidad*. <<http://portal.mma.gob.cl/division-recursos-naturales-ecosistemas-y-agua/>>. [Consulta: 12/10/2017].

⁵ Gobierno de Chile. *Ley 20380 sobre protección de animales*. <<https://www.leychile.cl/Navegar?idNorma=1006858>>. [Consulta: 12/10/2017].

3.2. Contexto tecnológico

El acceso masivo a los dispositivos móviles se ha convertido en una tendencia a nivel mundial en los últimos años, modificando la forma en que las personas se relacionan y desarrollan su vida cotidiana. Por ejemplo, según el *Informe mobile en España y en el mundo 2017*, que cada año genera la consultora de proyectos de marketing digital ditrendia⁶, en 2016, un 61% de los usuarios de móvil del mundo afirmó que miraba su dispositivo durante los 5 minutos posteriores a haber despertado. Esta tendencia global, se explica en parte por las considerables capacidades de procesamiento de *smartphones* y *tablets*, similares a las de los ordenadores menos potentes, y por otros factores como la portabilidad y la posibilidad de una conexión permanente a Internet. Lo anterior, hace de la tecnología móvil una herramienta ideal tanto para acceder, como para distribuir información de manera eficiente y eficaz en cualquier momento y casi desde cualquier lugar.

El estudio realizado por ditrendia entrega datos sumamente relevantes, que destacan las principales tendencias globales relacionadas con los dispositivos móviles. Según este informe, 4,9 mil millones de personas (un 66% de la población mundial) poseen un móvil, superando incluso a la cantidad de personas que cuentan con acceso a Internet (3,7 mil millones). Además, de acuerdo a este mismo estudio, un usuario de móvil utiliza su dispositivo en promedio 170 minutos al día, lo que se encuentra por encima de los 120 minutos dedicados al ordenador. Finalmente, otro dato relevante que aporta el Informe es que, al año 2016 más del 50% de las consultas que se realizaron en el buscador Google en el mundo se llevaron a cabo desde un móvil (más de 50 mil millones de búsquedas mensuales).

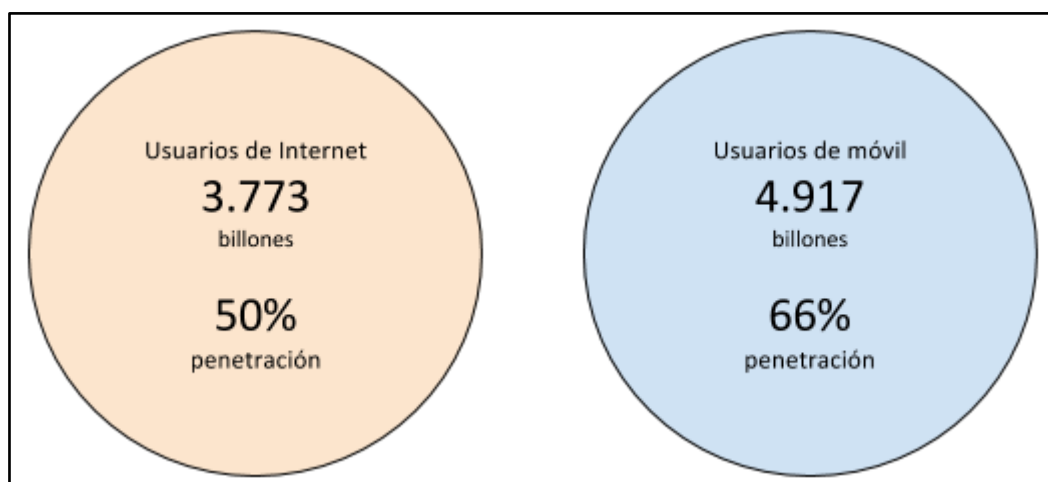


Fig. 1. Porcentaje de usuarios de Internet frente a porcentaje de usuarios móviles, a nivel mundial. Fuente: ditrendia (2017).

En Chile, el panorama no es distinto, según el estudio *IMS Mobile in LatAm Study*, publicado en septiembre de 2016 por la empresa de comunicación y marketing digital IMS Corporate⁷ en conjunto

⁶ Ditrendia. *Nosotros*. <<https://ditrendia.es/nosotros/>>. [Consulta: 11/10/2017].

⁷ IMS Corporate. *IMS Group*. <<https://www.ims corporate.com/ims-group/>>. [Consulta: 11/10/2017].

con la compañía de investigación de marketing en Internet comScore⁸, Chile ocupa el tercer puesto de Latinoamérica en cuanto a la utilización de *smartphones* con un promedio de 11,6 horas semanales de uso, siendo superado solo por Brasil y Colombia con promedios de 11,8 y 12,2 horas semanales de uso respectivamente. Además, según esta investigación los teléfonos inteligentes son utilizados habitualmente (4-5 veces por semana o más) para realizar una gran variedad de actividades *online* cotidianas como efectuar búsquedas en Internet, leer determinados contenidos, mirar programas de tv o películas, ver o generar videos, trabajar de manera colaborativa, revisar finanzas personales o realizar pagos en línea, entre otras actividades.

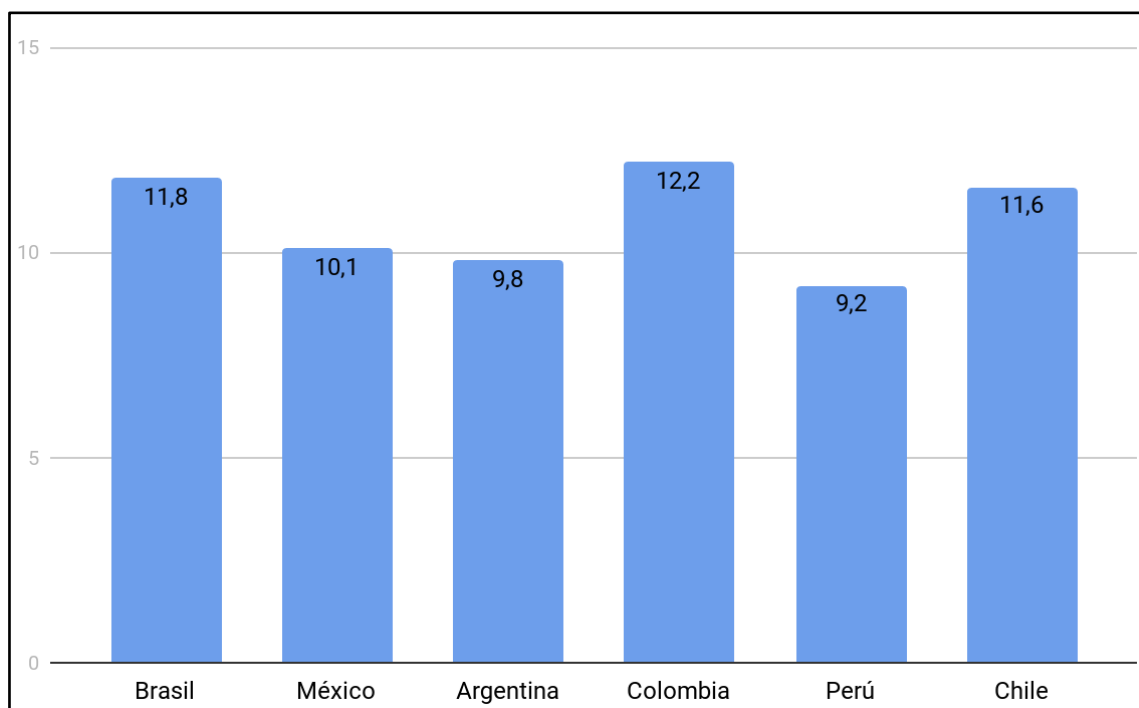


Fig. 2. Horas promedio de utilización de los dispositivos móviles en Latinoamérica. Fuente: IMS Corporate (2016).

3.3. Aplicación móvil sobre fauna chilena

Un paso primordial para comprometer a la ciudadanía y contribuir a la conservación y a la protección de las especies animales del país, consistiría en difundir y poner al alcance de la comunidad, de forma fácil y asequible, información referente a la fauna, con el objetivo de educar y concientizar sobre el rol fundamental que esta cumple dentro del ecosistema. Considerando la masificación de los dispositivos móviles en Chile y el mundo, parece apropiado considerarlos como una opción perfectamente válida para lograr lo anteriormente señalado.

Es posible evidenciar que actualmente no existe una solución de tecnología móvil que ofrezca a la comunidad acceso a información relevante y confiable acerca de las diferentes especies animales de

⁸ comScore. *Sobre comScore*. <<http://www.comscore.com/lat/Sobre-comScore/Hoja-informativa-sobre-comScore>>. [Consulta: 11/10/2017]

Chile, al consultar en las tiendas de aplicaciones móviles más utilizadas. Por ejemplo, al realizar búsquedas por términos clave como “fauna chilena⁹” o “animales de Chile¹⁰” en la plataforma de distribución digital Google Play Store, se puede observar que estas consultas no devuelven resultados acorde a lo que se esperaría y en su lugar aparecen juegos, aplicaciones de turismo, o aplicaciones de viajes, entre otras. Algo similar ocurre al introducir estos mismos términos clave en la App Store de Apple, donde simplemente no existen resultados.

Considerando tanto el escenario tecnológico como el contexto ambiental del país, es que surge la idea de desarrollar un prototipo de aplicación móvil sobre fauna chilena. Esta aplicación permitirá a los usuarios acceder a información relevante acerca de las especies animales de Chile, como por ejemplo, fotografías, características distintivas de cada animal, hábitat en el que se desenvuelven o su estado actual de conservación.

La implementación de un proyecto de estas características permitiría contribuir, por ejemplo, al desarrollo del turismo ecológico en los diferentes parques y reservas nacionales de la Corporación Nacional Forestal (CONAF)¹¹, apoyando la apreciación de las especies y del medio ambiente en el que habitan. Cabe señalar que, según la Cuenta Pública 2015¹² del Servicio Nacional de Turismo (SERNATUR)¹³, durante el año 2015 el turismo fue la sexta actividad económica del país, convirtiéndose en uno de los pilares fundamentales de la economía chilena. En el corto plazo, el turismo probablemente continúe siendo una de las bases principales de la economía chilena, considerando que, por segundo año consecutivo, Chile es reconocido como el mejor destino de turismo de aventura del mundo¹⁴, en los World Travel Awards 2017, instancia anual en la que se galardona a más de un centenar de categorías del sector turístico.

Asimismo, una aplicación móvil de este tipo podría ser utilizada con fines educativos en escuelas y colegios del país para que los estudiantes se familiaricen con la fauna del lugar en el que viven, acorde a lo descrito en el Artículo 2º de la Ley 20.280 que indica que “El proceso educativo, en sus niveles básico y medio, deberá inculcar el sentido de respeto y protección a los animales, como seres vivientes y sensibles que forman parte de la naturaleza.”

⁹ Google Play. *Búsqueda: fauna chilena*.

<<https://play.google.com/store/search?q=fauna%20chilena&c=apps&hl=es>>. [Consulta: 12/10/2017].

¹⁰ Google Play. *Búsqueda: animales de Chile*.

<<https://play.google.com/store/search?q=animales%20de%20chile&c=apps&hl=es>>. [Consulta: 12/10/2017].

¹¹ CONAF. *Quiénes somos*. <<http://www.conaf.cl/quienes-somos/>>. [Consulta: 13/10/2017].

¹² SERNATUR. *Cuenta Pública 2015*. <<http://www.sernatur.cl/wp-content/uploads/2015/11/CUENTA-PUBLICA-2015-SERNATUR.pptx>>. [Consulta: 13/10/2017]

¹³ SERNATUR. *¿Qué hacemos?*. <<https://www.sernatur.cl/que-hacemos/>>. [Consulta: 13/10/2017].

¹⁴ World Travel Awards. *World's leading adventure tourism destination 2017*.

<<https://www.worldtravelawards.com/award-worlds-leading-adventure-tourism-destination-2017>>. [Consulta: 13/10/2017].

4. Marco teórico

4.1. Escenario móvil actual

El avance en los protocolos de comunicación y el incremento en las capacidades técnicas de los circuitos integrados incorporados en los teléfonos móviles, son los factores fundamentales que han favorecido al desarrollo acelerado de las redes inalámbricas de telefonía móvil (Mártel, 2017) y de las telecomunicaciones en general. A su vez, la proliferación de las redes de telefonía móvil ha provocado un incremento en la cantidad de lugares desde los que es posible conectarse a Internet, otorgando al usuario una mayor libertad para acceder a la Red, favoreciendo así su movilidad. Este contexto tecnológico, hace posible que hoy en día, se puedan realizar una gran variedad de trámites de forma virtual, sin necesidad de desplazarnos al lugar donde tradicionalmente se llevan a cabo. Esta es una tendencia global que se puede verificar al observar que, en Latinoamérica, el 56,1% de la población tiene acceso a Internet. En el caso concreto de Chile, la penetración de este medio alcanza el 71,7%, siendo el país latinoamericano donde un mayor porcentaje de la población tiene acceso a Internet. Además, del total de usuarios con acceso a Internet en América Latina, el 90% se conecta desde un dispositivo móvil, ya sea desde un *smartphone* o desde una *tablet*, aunque solo entre el 1% y el 5% de los usuarios, lo hace desde este último dispositivo (IMS Corporate, 2016).

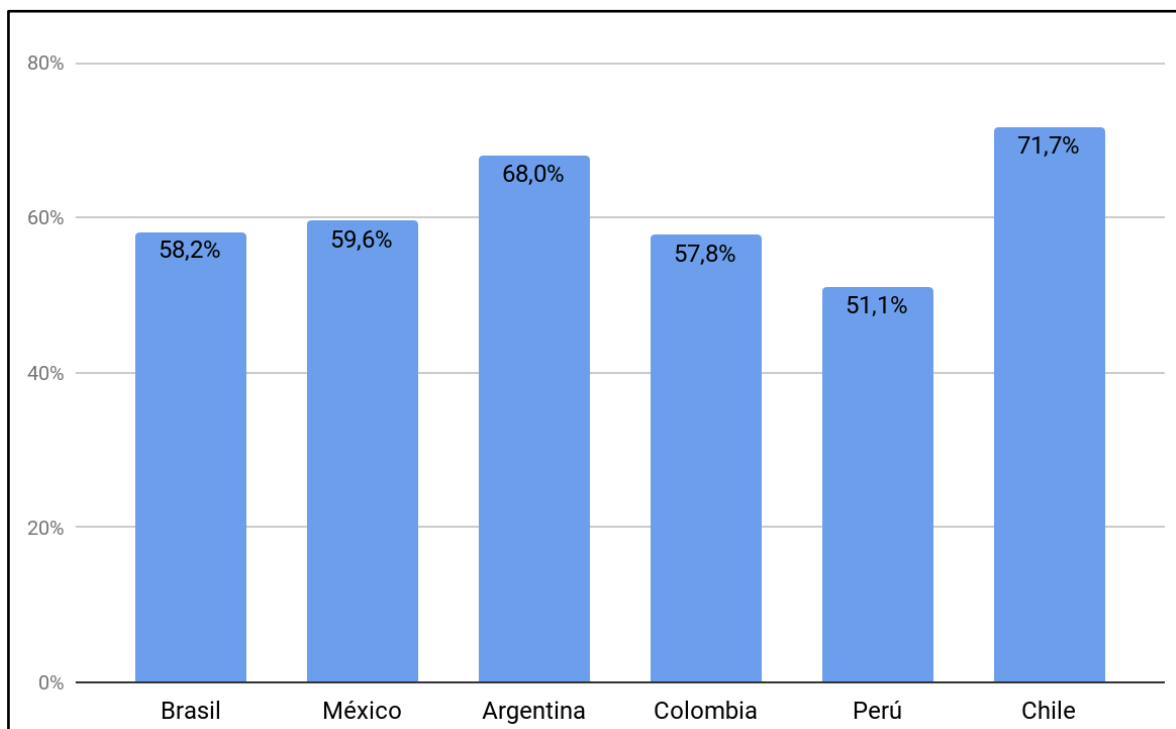


Fig. 3. Porcentaje de la población con acceso a Internet en Latinoamérica, por país. Fuente: IMS Corporate (2016).

Es en este escenario de movilidad, en el que la gran mayoría de los usuarios se conecta desde dispositivos móviles y no desde ordenadores, y en el que el rol de Internet como plataforma de servicios y contenidos para móviles es cada vez más importante, donde el término Web Móvil cobra relevancia. Este concepto, puede ser interpretado desde dos perspectivas. Por una parte, y desde un enfoque centrado en los dispositivos, el concepto de Web Móvil hace referencia al acceso a la Web desde cualquier dispositivo, cuya principal característica sea la movilidad (Kroski, 2008), como por ejemplo, un teléfono móvil o una *tablet*. Por otra parte, y en contraposición al enfoque anterior la idea de Web Móvil se refiere a un acceso a la Web desde cualquier lugar, independientemente del tipo de dispositivo que utilice para ello (W3C, 2007). En otras palabras, el término “móvil” se refiere fundamentalmente al usuario y no al dispositivo (Ballard, 2007), en un contexto donde la movilidad es algo más que ser inalámbrico, y que trasciende la libertad de los cables y sugiere una experiencia de usuario completamente diferente (Moll, 2008). Este último enfoque, centrado en el usuario y no en los dispositivos, parece ser el que más se ajusta al entorno tecnológico actual, en el que existe más de una vía para acceder al contenido de la Web.

Las vías de acceso a la Web corresponden a los distintos dispositivos desde los que el usuario puede consultar contenidos, como por ejemplo, consolas de videojuegos, smart TVs, ordenadores de sobremesa, portátiles, *smartphones*, *tablets*, etc. Cada uno de estos dispositivos tiene instalado distintos navegadores web y posee diferentes tamaños de pantalla, incluso, los dispositivos móviles permiten cambiar la orientación de esta (horizontal o vertical, en inglés *landscape* o *portrait*), por lo que, lograr una visualización correcta de la información en cada dispositivo resulta una tarea crucial, y es justamente aquí donde el diseño web desempeña un rol fundamental. Actualmente, existen dos tendencias de diseño web que persiguen un acceso al contenido web desde cualquier dispositivo. Por una parte, el *Adaptive Web Design* (en adelante, AWD), corresponde a una técnica de diseño que utiliza distintas plantillas web estáticas, con información preseleccionada, que se visualizarán de acuerdo a tamaños de pantalla preestablecidos. En otras palabras, se emplean un código fuente y un diseño gráfico optimizados o adaptados según el dispositivo en uso. Es el servidor, en que está alojado el sitio web, el que detecta el dispositivo del usuario y carga la versión más adecuada para este. Un inconveniente con el AWD es que el contenido mostrado para uno u otro dispositivo, probablemente, sea distinto según la plantilla utilizada. Por otra parte, el *Responsive Web Design* (en adelante, RWD), es una técnica de diseño web que se lleva a cabo mediante el uso de tres elementos básicos, un diseño de página basado en una cuadrícula flexible, contenido multimedia flexible y el módulo *media queries*, de la tercera versión de la especificación de las hojas de estilo en cascada (CSS, por su sigla en inglés) (Marcotte, 2011). Las *media queries* permiten establecer reglas de estilo según el tipo de medio¹⁵ desde el cual se está accediendo al contenido, y a la orientación y el tamaño de pantalla de este, propiedades que serán detectadas por el navegador web del usuario. Al utilizar un diseño basado en cuadrícula, los contenidos se muestran en secciones, las que se

¹⁵ Los medios corresponden a los dispositivos de acceso al contenido. Actualmente, los tipos de medio más utilizados son *screen*, para definir la presentación de la página web en pantalla, y *print* para definir el aspecto de impresión de la página web en pantalla.

reorganizan según el tamaño de pantalla del dispositivo en uso o el tamaño de la ventana del navegador. El RWD, permite acceso siempre al mismo contenido, pero representado de manera diferente, según el tamaño y la resolución de pantalla del dispositivo.

Además de estas vías de acceso, es necesario considerar, en el desarrollo de contenido web, las diferentes capacidades físicas del público objetivo, como problemas de visión, auditivos, cognitivos o motores, y también ciertos factores sociales como rangos etarios, legislaciones, el idioma, etc. Las diferentes capacidades, tanto de usuarios como de dispositivos, formarán distintos contextos de acceso al contenido web, que deberían resultar en distintas representaciones para mostrar la misma información. Es en este punto donde el concepto *One Web* cobra sentido. Esta visión, promovida por la entidad de estandarización de las tecnologías ligadas a la web, Consorcio World Wide Web (en adelante, W3C), hace referencia a la existencia de una única Web, lo que además concuerda con otras ideas de esta organización como la construcción de una Web para todo el mundo (*Web for all*) y una Web accesible desde cualquier dispositivo (*Web on everything*)¹⁶. A partir de este planteamiento, parece correcto afirmar que no es necesario construir varias versiones alternativas de un mismo contenido web, según las características del dispositivo o las capacidades del usuario, sino que se deben proporcionar los medios para que, en la medida de lo posible, cualquier persona, desde cualquier dispositivo, pueda consultar un determinado servicio o contenido web.

Además, todo parece indicar que la tendencia *One Web* también incorporará al ámbito de las aplicaciones móviles. Prueba de esto es la iniciativa Firebase App Indexing de Google¹⁷, servicio que indexa el contenido de una determinada aplicación móvil y lo incluye en las páginas de resultados (SERP, por su sigla en inglés) de Google, para consultas relacionadas con ese contenido. Google lleva a cabo esta indexación mediante los enlaces internos (en inglés *deep links*) de las aplicaciones, cada uno de los cuales está vinculado a un contenido específico dentro estas. Actualmente, se encuentran indexados 50 mil millones de *deep links* y un 25% de las consultas realizadas desde dispositivos Android los devuelven como resultado en las SERP de Google (Google, 2017). Una vez que un *deep link* aparece en las SERP de Google, los usuarios tienen la posibilidad de instalar la aplicación o acceder al contenido, directamente desde el navegador, si ya la tienen instalada. Es preciso señalar que los contenidos indexables por Firebase deben tener su correspondencia en un sitio web tradicional. En otras palabras, es necesario estructurar el sitio web y la aplicación móvil de tal manera que los localizadores uniforme de recursos (URL, por su sigla en inglés) que lleven a las páginas del sitio sean los mismos que lleven a determinadas vistas en la aplicación. De esta manera, Google rastreará el sitio web, encontrará estos vínculos y los usará para enviar a los usuarios directamente a la aplicación móvil (Google, 2017).

¹⁶ W3C. *W3C mission*. <<https://www.w3.org/Consortium/mission.html>>. [Consulta: 26/10/2017].

¹⁷ Google. *Firebase App Indexing*. <<https://firebase.google.com/docs/app-indexing/?hl=es>>. [Consulta: 30/10/2017].

4.2. Dispositivos móviles

Los dispositivos móviles, son dispositivos que por sus características físicas pueden ser transportados de un lugar a otro conservando su funcionalidad, permitiendo al usuario llevar a cabo sus actividades habituales. Como se mencionó anteriormente, la mayoría de los usuarios se conecta a la Web a través de dispositivos móviles y, probablemente, esta tendencia se deba, en parte, a la portabilidad y versatilidad de estos equipos en comparación con ordenadores portátiles y de sobremesa. Sin embargo, si de prestaciones técnicas se trata, las computadoras superan ampliamente a los dispositivos móviles, o al menos esto era así hasta hace algunos años, ya que, debido a la miniaturización de componentes electrónicos como circuitos integrados, resistencias y transistores, la diferencia entre las capacidades técnicas de ordenadores y dispositivos móviles tiende a diluirse cada vez más, haciendo de *smartphones* y *tablets* verdaderos ordenadores de bolsillo. Con el objetivo de visualizar de mejor manera lo señalado anteriormente, se realiza a continuación una comparativa entre las prestaciones técnicas de uno de los móviles más potentes que existen a la fecha, versus las capacidades de dos de los ordenadores más solicitados por el público a través de comercio electrónico. En concreto, se comparan el terminal Android, OnePlus 5 que al mes de agosto de 2017 era el smartphone del mercado que ofrecía un mayor rendimiento (Antutu, 2017), con el portátil Acer Aspire E15 E5-575-33BM¹⁸ y el ordenador de sobremesa HP 8300 SFF¹⁹, modelos que según datos de la compañía de comercio electrónico Amazon Inc., fueron los más solicitados por el público al mes de octubre de 2017.

	OnePlus 5	Acer Aspire E15	HP 8300
Cantidad de núcleos/ velocidad de la CPU	4 núcleos a 2,45 GHz + 4 núcleos a 1,9 GHz	2 núcleos a 2,4 GHz	4 núcleos a 3,2 GHz
Velocidad base de la GPU	710 MHz	1150 MHz	650 MHz
Cantidad de RAM	6/8 GB	4 GB	8 GB
Capacidad de almacenamiento interno	64/128 GB	1 TB	500 GB
Resolución de pantalla	1920 x 1080 px	1920 x 1080 px	No aplica
Capacidad de batería	3.300 mAh	2800 mAh	No aplica

¹⁸ Amazon. *Best sellers in laptop computers*. <<https://www.amazon.com/Best-Sellers-Computers-Accessories-Laptop/zgbs/pc/565108/>>. [Consulta: 01/11/2017].

¹⁹ Amazon. *Best sellers in desktop computers*. <<https://www.amazon.com/Best-Sellers-Computers-Accessories-Desktop/zgbs/pc/565098/>>. [Consulta: 01/11/2017].

	OnePlus 5	Acer Aspire E15	HP 8300
Conectividad	LTE Cat 12 NFC Bluetooth v5.0 Wifi 2x2 MIMO USB Type-C 2.0	Bluetooth 4.1 Wifi 802.11 USB 3.0	LAN Ethernet USB 3.0

Tabla 1. Comparación de las prestaciones de un móvil con las de un ordenador portátil y uno de sobremesa.

Como se observa en la tabla anterior, en relación con la ejecución de tareas, tanto la velocidad de la unidad central de procesamiento (CPU, por su sigla en inglés) como la de la unidad de procesamiento gráfico (GPU, por su sigla en inglés) y la cantidad de RAM del OnePlus 5 están a la par con las de ambos ordenadores. Si bien, en cuanto a la capacidad de almacenamiento interno, tanto el portátil como el equipo de sobremesa son ampliamente superiores, 128 GB son más que suficientes para cubrir las necesidades de almacenamiento de un móvil. En este sentido además, esta capacidad suele complementarse mediante distintas posibilidades de almacenamiento en la nube, las que implican una cierta liberación de esta carga para los *smartphones*. En cuanto a la resolución de pantalla, tanto la del OnePlus 5 como la del Acer Aspire E15 alcanzan el estándar 1080p, permitiendo ambos la visualización de contenidos en alta definición (HD, por su sigla en inglés). En relación con la capacidad de la batería, la del terminal OnePlus es superior a la del ordenador portátil Acer, por lo que teóricamente, debería tener un tiempo mayor de autosuficiencia energética. Con respecto a la conectividad, se puede apreciar que mientras mayor sea la movilidad de un equipo, más serán las opciones de este, tanto para conectarse a la Red como para vincularse con otros dispositivos. Continuando esta misma idea, es necesario mencionar que, uno de los puntos fuertes de los dispositivos móviles respecto a los ordenadores, son los sensores que los primeros integran, los cuales permiten acceder a una serie de funcionalidades que las computadoras posibilitan en menor medida o simplemente no permiten. Entre los sensores más utilizados por los desarrolladores de aplicaciones y, prácticamente integrados en todos los dispositivos móviles actuales, se encuentran, el sensor de sistema de posicionamiento global (GPS, por su sigla en inglés) que, tal como su nombre indica permite determinar la ubicación del terminal móvil sobre un mapa base a partir de sus coordenadas geográficas y es ampliamente utilizado en aplicaciones de geolocalización como Google Maps o Apple Maps; el giroscopio, sensor capaz de medir el movimiento de un móvil en todos sus ejes y empleado habitualmente en juegos donde se haga uso de capacidades de rotación o en conjunto con el GPS, para facilitar la navegación en un mapa base; el sensor de proximidad, usado normalmente para apagar la pantalla del móvil cuando se detecta que el usuario lo tiene pegado a la oreja; el sensor de luz utilizado para ajustar automáticamente el brillo de la pantalla del dispositivo móvil y el magnetómetro empleado en aplicaciones tipo brújula, ya que es capaz de medir los campos magnéticos terrestres. Esta comparación técnica, entre equipos contemporáneos, permite corroborar el planteamiento de que, la brecha entre las prestaciones de dispositivos móviles y las de ordenadores es cada vez menor.

Esta brecha cada vez más reducida y la utilización de los dispositivos móviles en el ámbito laboral, han impulsado a algunas compañías tecnológicas al desarrollo de iniciativas que posibiliten utilizar los terminales como si de equipos de sobremesa se tratase. Probablemente, los proyectos más destacados en este contexto sean Microsoft Surface²⁰ y Windows Continuum²¹ de Microsoft Corporation. Por una parte, Surface corresponde a una línea de *tablets* híbridas capaces de ser utilizadas como portátiles, como *tablets* o incluso para llevar a cabo labores de mayor precisión como escribir o dibujar, ya que integran panel táctil (en inglés *touchpad*), teclado y bolígrafo digital. Por otra parte, Continuum es un accesorio hardware que permite convertir cualquier *smartphone* Windows en un ordenador de escritorio totalmente funcional, ya que permite conectar periféricos como ratón, teclado y monitor e incluso posibilita el uso de la propia pantalla del móvil como teclado y panel táctil virtual. Además, Continuum permite continuar utilizando el móvil mientras está en modo escritorio. Otra iniciativa destacada en esta área es el proyecto Samsung Dex Station²², *gadget* para *smartphones*, lanzado en 2017, que permite pasar, sin mayores complejidades, de una experiencia móvil a una experiencia de escritorio. Este implemento integra varios tipos de conexiones que permiten usar los terminales Galaxy S8/S8+ y Note8 como equipos de escritorio, en concreto, posee un puerto USB Type-C para conectar el móvil, dos puertos USB 2.0 para conectar dispositivos de almacenamiento o periféricos como ratón y teclado, una salida HDMI que permite conectar un monitor y un puerto Ethernet para conectarse a la Red mediante cable. Además, posee un ventilador en caso de ser necesaria la refrigeración del conjunto, cuando la carga de trabajo sea intensa. Con respecto al software modo DeX de Samsung, este posee un “escritorio”, similar al de los sistemas operativos más utilizados como Microsoft Windows o MacOS, el que integra una barra de herramientas que permite navegar entre las aplicaciones instaladas y las utilizadas recientemente y visualizar notificaciones si corresponde. También permite la utilización de combinaciones de tecla habituales como ALT + TAB, para cambiar de aplicaciones, y permite guardar el estado de uso del sistema una vez desconectado el dispositivo móvil. Finalmente, es preciso señalar que DeX Station está diseñado para un uso informático no muy exigente y que por ahora no todas las aplicaciones instaladas en el móvil son compatibles en modo DeX, sin embargo, parece ir por buen camino en busca de la convergencia entre móvil y escritorio.

Al igual que los ordenadores, los dispositivos móviles son esencialmente sistemas informáticos, los que como es sabido, corresponden a sistemas que reciben y procesan datos para transformarlos en información útil para los seres humanos y, están conformados por una serie de componentes físicos denominados hardware, y por un soporte lógico llamado software. Entre el software de cualquier equipo informático destaca el sistema operativo (en adelante, SO), el cual posibilita la administración de los recursos de hardware y también sirve de nexos, permitiendo a los usuarios interactuar con el dispositivo en el que se encuentra instalado, a través de la ejecución de aplicaciones informáticas.

²⁰ Microsoft. *Surface*. <<https://www.microsoft.com/es-es/surface>>. [Consulta: 02/11/2017].

²¹ Microsoft. *Continuum*. <<https://www.microsoft.com/es-es/windows/continuum>>. [Consulta: 02/11/2017].

²² Samsung. *Samsung DeX*. <<http://www.samsung.com/es/apps/samsung-dex/>>. [Consulta: 02/11/2017].

Los SO orientados específicamente a dispositivos móviles se denominan sistemas operativos móviles (en adelante, SO móviles). Los SO móviles que han liderado el mercado en los últimos años son Android, basado en núcleo Linux y actualmente desarrollado por Google, y iOS, desarrollado por Apple Inc. Una pequeña cuota de mercado la dominan otros Sistemas Operativos como BlackBerry OS y Windows Phone/Windows 10 Mobile de Microsoft. Según el estudio *IMS mobile in LatAm study*, en Latinoamérica el SO móvil más utilizado en *smartphones* es Android con una penetración del 81%, mientras que iOS tiene un impacto de solo el 17%. Android también lidera el mundo de las *tablets* con una penetración del 72% versus el 25% de iOS. En el caso concreto de Chile, es posible verificar la tendencia regional latinoamericana, al visualizar los datos proporcionados por la herramienta de análisis de tráfico web StatCounter, mediante los cuales se observa que, al mes de septiembre de 2017, el 84,29% del tráfico web desde *smartphones* en Chile provenía de dispositivos Android, mientras que solo el 14,92% procedía de equipos con iOS²³.

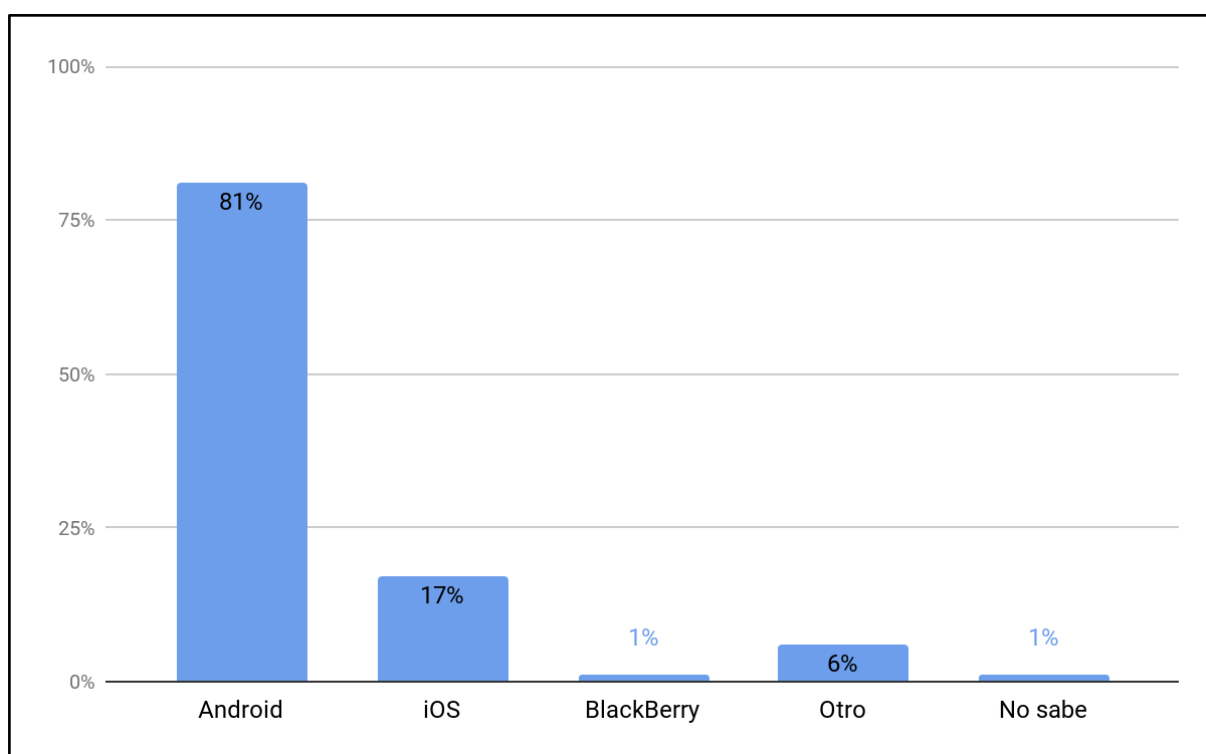


Fig. 4. SO móvil líder en Latinoamérica. Fuente: IMS Corporate (2016).

²³ StatCounter. *Mobile operating system market share in Chile - October 2017*. <<http://gs.statcounter.com/os-market-share/mobile/chile>>. [Consulta: 13/10/2017].

4.3. Aplicaciones móviles

El escenario móvil antes descrito ha provocado un incremento exponencial en la utilización de aplicaciones móviles y, en este sentido, todo parece indicar que nos encontramos en la época dorada de las aplicaciones móviles. En concreto, el uso de aplicaciones en teléfonos inteligentes ha aumentado en un 111% a nivel global durante el periodo 2014 - 2016, representando casi el 60% del tiempo total que los usuarios dedican al mundo digital, superando ampliamente el uso de la Web Móvil (8,4%) y del ordenador (33,2%) (Ditrendia, 2017). Se pronostica que esta tendencia de crecimiento en el mercado de aplicaciones móviles continúe y que el sector crezca un 380%, sobre las cifras generadas en 2016 y las que se alcanzarán en 2021, pasando de 1,3 mil millones de dólares a 6,3 mil millones dólares en ese periodo de tiempo (Levitas, 2017). En el caso concreto de Chile, las aplicaciones también juegan un rol relevante para los usuarios de *smartphones*, quienes, en promedio, tienen instaladas 17 aplicaciones móviles (IMS Corporate, 2016).

Una aplicación móvil o app corresponde esencialmente a una pieza de software con funcionalidades muy específicas, diseñada y desarrollada para ejecutarse en dispositivos móviles con un sistema operativo determinado, y cuyo objetivo fundamental es que el usuario pueda realizar una tarea concreta desde su terminal, por ejemplo, gestionar su correo electrónico o escuchar música. Las aplicaciones móviles se encuentran disponibles para descarga desde plataformas de distribución digital denominadas tiendas de aplicaciones o *markets*. Estos portales especializados están asociados al sistema operativo del dispositivo móvil, por ejemplo, en Google Play Store se distribuyen aplicaciones para dispositivos con sistema operativo Android, mientras que en la App Store de Apple se pueden descargar aplicaciones desarrolladas para dispositivos con sistema operativo iOS.

Con respecto al desarrollo de aplicaciones móviles, es preciso mencionar la existencia de tres categorías distintas de construcción: aplicaciones web, aplicaciones nativas y aplicaciones híbridas, considerando factores de desarrollo de software como el presupuesto, el tiempo de desarrollo, o el público objetivo, entre otros.

Las aplicaciones web (en inglés *web apps*), corresponden esencialmente a sitios web, generalmente desarrollados en base a estándares tecnológicos web, y optimizados para su consulta desde dispositivos móviles. En cuanto a tiempo y coste, suelen ser la opción más rápida y económica de desarrollar. Se accede a ellas a través de una URL, mediante cualquier navegador web compatible con las tecnologías antes mencionadas. Dentro de sus principales ventajas se encuentran la independencia del sistema operativo del dispositivo desde el cual se esté accediendo, además, no es necesario descargar ni instalar nada en el dispositivo móvil, dado que las aplicaciones web se encuentran alojadas en un determinado servidor web, lo que además facilita su actualización y mantenimiento, ya que al tratarse de un sitio web los usuarios siempre tendrán la última versión del sistema. Una última ventaja que es preciso mencionar es que, al tratarse de sitios web, no es necesario someterlas a algún proceso de aprobación para publicarlas en las tiendas de aplicaciones.

En relación con los inconvenientes de las aplicaciones web es necesario mencionar que, al encontrarse alojadas en un servidor externo, su rendimiento será menor al de una aplicación instalada en local, además esto provoca una necesidad de conexión permanente a Internet. Aunque, es preciso señalar que, hoy en día existen mecanismos como el almacenamiento de datos en HTML5 (en inglés *HTML5 Web Storage*), que permiten utilizar las características de los navegadores web actuales para almacenar información en el lado del cliente y acceder a ella más tarde, sin necesidad de conexión a Internet. Internet Explorer 8 por ejemplo, permite guardar hasta 10 MB de información. Con respecto al aspecto y comportamiento (en inglés *look and feel*) de las aplicaciones web, éste se asemejará al de un sitio web tradicional, y en algunos casos muy concretos intentará emular el *look and feel* de las aplicaciones nativas. Un claro ejemplo de lo anterior es la aplicación web del periódico británico *Financial Times*²⁴, la cual posee una apariencia muy similar a la aplicación para iOS²⁵ de esta misma cabecera. Una última desventaja de las aplicaciones web es que, al no estar instaladas en el dispositivo móvil, el acceso a sensores como cámara, acelerómetro, giroscopio, etc. es limitado, lo que puede restringir la creación de determinadas funcionalidades.

Las aplicaciones nativas (en inglés *native apps*), se desarrollan utilizando el lenguaje nativo de cada sistema operativo móvil por lo que poseen siempre un alto rendimiento y un *look and feel* nativo. Lo anterior se lleva a cabo mediante un conjunto de herramientas determinado, proporcionado por cada plataforma, denominado kit de desarrollo de software (SDK, por su sigla en inglés). El coste de desarrollo de este tipo de aplicaciones suele ser bastante elevado, sobre todo si se construyen versiones para distintos SO, lo cual, además requiere de un tiempo mayor de construcción. Con respecto a su distribución, esta se lleva a cabo por medio de la tienda de aplicaciones correspondiente a cada SO. Antes de ser publicadas en las tiendas, es necesario que las aplicaciones nativas pasen por un proceso de revisión y aprobación, realizado por un equipo humano cualificado. Este proceso de revisión puede tardar un par de horas o varios días dependiendo de cada plataforma. Dentro de las principales ventajas del desarrollo de aplicaciones nativas se encuentra la posibilidad de utilizar la totalidad de los recursos de hardware del dispositivo móvil, al estar alojadas en el dispositivo no es necesaria una conexión permanente a Internet, a menos que una determinada funcionalidad lo requiera, la instalación en local de las aplicaciones nativas además hace posible el envío de notificaciones o avisos al usuario en determinadas situaciones. Con respecto a los principales inconvenientes de las aplicaciones móviles nativas, es preciso mencionar que son dependientes de la plataforma para la que fueron desarrolladas, en otras palabras, una aplicación nativa solo funcionará en un sistema operativo concreto, por lo que el desarrollo multiplataforma de aplicaciones móviles nativas implica tanto una inversión económica como de tiempo. Una desventaja de las aplicaciones nativas respecto de las aplicaciones web es que, para llevar a cabo la actualización de una aplicación nativa es necesario que el usuario descargue e instale la versión más reciente de esta.

²⁴ Financial Times. *Web app*. <<https://app.ft.com/>>. [Consulta: 15/10/2017].

²⁵ Financial Times. *iOS app*. <<https://itunes.apple.com/app/apple-store/id1200842933>>. [Consulta: 15/10/2017].

Una tercera opción de desarrollo corresponde a las aplicaciones híbridas, las que tal como su nombre indica mantienen características tanto de las aplicaciones web como de las aplicaciones nativas. Se podría definir una aplicación híbrida como un sitio web encapsulado, que se ejecuta mediante el visor web nativo de cada plataforma (en inglés *WebView*). Al igual que las aplicaciones web, las aplicaciones híbridas se construyen utilizando tecnologías propias del desarrollo web, pero apoyándose en la utilización de algún entorno de trabajo (en adelante, *framework*) como por ejemplo, PhoneGap o JQuery Mobile. Con respecto a este último punto, es necesario señalar que el desarrollo de aplicaciones híbridas también se puede realizar utilizando lenguajes de programación ajenos al desarrollo web como C o C++. Tanto el tiempo como el coste de desarrollo de aplicaciones híbridas suele situarse en un punto intermedio entre el de aplicaciones web y aplicaciones nativas. En cuanto a rendimiento, son más potentes que las aplicaciones web, pero menos que las aplicaciones nativas. A pesar de que su *look and feel* es emulado, es posible obtener resultados semejantes al aspecto y comportamiento nativo. Al usar tecnologías web para construirlas, se mantiene, en muchos sentidos, su carácter multiplataforma. Al igual que las aplicaciones nativas, estas se distribuyen a través de las tiendas de aplicaciones, se instalan de manera local en el dispositivo móvil, por lo que es necesaria su reinstalación en caso de actualizaciones, y permiten acceder casi a la totalidad de los recursos de hardware de este.

A continuación, se presenta una tabla resumen con las principales características de cada una de las opciones de desarrollo de aplicaciones móviles antes descritas.

	Aplicaciones web	Aplicaciones nativas	Aplicaciones híbridas
Tecnología de desarrollo	Estándares de desarrollo web	Lenguaje propio de cada plataforma	Típicamente, mediante estándares web + <i>framework</i>
Tiempo de desarrollo	Menor	Mayor	Intermedio
Coste de desarrollo	Bajo	Elevado	Intermedio
Plataforma	Multiplataforma	SO concreto (como Android o iOS)	Es posible exportar a distintas plataformas
Rendimiento	Bajo	Óptimo	Alto
Look and feel	Emulado	Nativo	Emulado
Acceso al hardware del dispositivo	Restringido	Completo	Intermedio
Necesidad de conexión a Internet	Limitada a las capacidades de almacenamiento del navegador web	Dependiente de las funcionalidades de la aplicación	Dependiente de las funcionalidades de la aplicación

	Aplicaciones web	Aplicaciones nativas	Aplicaciones híbridas
Distribución	Mediante una URL	Tiendas de aplicaciones (proceso de aprobación)	Tiendas de aplicaciones (proceso de aprobación)
Instalación	Servidor externo	Local	Local
Actualización desde el punto de vista del usuario	Acceso siempre a la versión más reciente	Es necesario reinstalar la aplicación	Es necesario reinstalar la aplicación
Actualización desde el punto de vista del desarrollador	Modificación del código fuente en servidor web correspondiente (similar al de un sitio web tradicional)	Modificación del código fuente en local y posterior proceso de aprobación	Modificación del código fuente en local y posterior proceso de aprobación

Tabla 2. Resumen de las características de los tipos de aplicaciones móviles.

4.3.1. Lenguajes de programación

Como ya se mencionó, las aplicaciones móviles corresponden, fundamentalmente, a programas informáticos, por lo que, para construirlas es necesaria la utilización de determinados lenguajes de programación. Un lenguaje de programación corresponde a un lenguaje formal²⁶ mediante el cual es posible definir secuencias de instrucciones para que sean ejecutadas por una máquina. En relación con la creación de aplicaciones móviles, según la opción de desarrollo escogida, se utilizan lenguajes de programación concretos.

Por un lado, tanto para el desarrollo de aplicaciones web como para el desarrollo de aplicaciones híbridas, se emplean las tecnologías impulsadas por el W3C, estas son HTML, CSS y JavaScript, que corresponden a estándares de formato abierto, por lo que su uso se encuentra libre de restricciones legales y económicas. HTML (sigla en inglés de HyperText Markup Language) es el lenguaje de marcado²⁷ estándar para la elaboración de páginas web, el cual describe la estructura de estas mediante elementos HTML representados por etiquetas. Las etiquetas corresponden a segmentos de contenido de una página web como encabezados, párrafos y tablas (W3Schools, 2017). Mientras que mediante HTML es posible definir la estructura de un documento web, a través de CSS se puede describir la presentación gráfica o estilo de un documento HTML, es decir, CSS describe cómo deben mostrarse los elementos HTML (W3Schools, 2017). Por su parte, JavaScript (en adelante, JS) es un

²⁶ En matemáticas y en computación, un lenguaje formal es aquel que posee un conjunto de símbolos finitos (vocabulario) y reglas de escritura (sintaxis) definidos formalmente, contruidos para expresar situaciones en un área específica del conocimiento.

²⁷ Un lenguaje de marcas o lenguaje de marcado es aquel que, a través de la codificación de etiquetas o marcas, permite definir la presentación de un texto y describir los bloques estructurales del mismo.

lenguaje de programación interpretado²⁸, desarrollado originalmente por Netscape Communications Corporation, que puede ser implementado en un documento HTML para otorgar dinamismo a este mediante la implementación de funcionalidades en sus elementos. JS ocupa el séptimo puesto del ranking de los lenguajes de programación más utilizados en el mundo (Cass, 2017). El componente fundamental que permite acceder a los diferentes elementos de un documento HTML, ya sea mediante CSS para dar estilo o a través de JS para otorgar funcionalidad, es el modelo de objetos del documento (DOM, por su sigla en inglés). El DOM corresponde a una interfaz de programación de aplicaciones (API, por su sigla en inglés)²⁹ para documentos válidos y bien formados³⁰ que define, tanto la estructura lógica de estos como el modo en que se accede y se manipulan sus elementos (Hégaret, 2004). Lo anterior, mediante la transformación de los elementos de un documento HTML en un conjunto de componentes interconectados entre sí, denominados nodos, los que forman un árbol³¹. Es así como la utilización del DOM y el empleo conjunto de HTML, CSS y JS, permiten separar, respectivamente, el contenido, la presentación gráfica y el comportamiento de un documento HTML. Si bien es posible crear y editar archivos de texto plano desde un editor de texto cualquiera como TextEdit en macOS, Gedit en sistemas Linux o bloc de notas en Microsoft Windows, lo más recomendable es la utilización de un entorno de desarrollo integrado (IDE, por su sigla en inglés), instrumento que agrupa y proporciona acceso a un conjunto de herramientas y servicios que facilitan las tareas de desarrollo de software, entre los más importantes destacan, editor inteligente de código fuente, intérprete y/o compilador, depurador de errores, diseñador de interfaces gráficas de usuario (GUI, por su sigla en inglés), plantillas de código y algunas otras. Dentro de los IDE más utilizados para el desarrollo web se encuentran, NetBeans IDE, PhpStorm y IntelliJ IDEA Ultimate edition. Con respecto a la documentación disponible sobre los estándares HTML, CSS y JS, ésta es fácilmente accesible desde el apartado *All standards and drafts*³² del sitio web del W3C, donde existe un listado con información fundamental sobre estas y otras tecnologías web. Además de lo anterior, existe una serie de otras iniciativas, de entidades vinculadas con ámbito web, que ponen a disposición de los desarrolladores información sobre la triplete tecnológica del W3C, como el proyecto W3Schools³³ o el sitio web MDN Web docs³⁴ de la Corporación Mozilla. Como se trata de estándares tecnológicos ampliamente difundidos y utilizados prácticamente desde los orígenes de la

²⁸ En computación, un lenguaje interpretado corresponde a aquel cuyas implementaciones no requieren ser compiladas para ejecutarse y normalmente lo hacen instrucción por instrucción. Son ampliamente utilizados en la programación web y para la creación de aplicaciones cliente/servidor.

²⁹ Una API corresponde a un módulo de software que posee un conjunto de funciones y procedimientos que pueden ser utilizadas por otro software. Una API, además especifica la forma en que estos módulos de software deben comunicarse.

³⁰ Un documento bien formado es aquel que no tiene errores de sintaxis, mientras que un documento válido, corresponde a aquel que además de estar bien formado cumple con las normas establecidas en su estructura.

³¹ En informática, un árbol corresponde a un conjunto abstracto de datos que permite representar elementos que poseen una relación jerárquica. En términos simples, un árbol, comienza con un elemento raíz (en inglés *root*) y se extiende en varias ramificaciones (en inglés *edge*), hasta terminar finalmente en una hoja (en inglés *leaf*).

³² W3C. *All standards and drafts*. <<https://www.w3.org/TR/>>. [Consulta: 04/11/2017].

³³ W3Schools. *About W3Schools*. <<https://www.w3schools.com/about/default.asp>>. [Consulta: 04/11/2017].

³⁴ Mozilla. *About MDN*. <<https://developer.mozilla.org/en-US/docs/MDN/About>>. [Consulta: 04/11/2017].

Web, la comunidad de desarrolladores existente es muy amplia, por lo que la información extraoficial referente a la construcción de documentos web resulta de fácil acceso también. Finalmente, en relación con el aprendizaje de estos lenguajes respecto al tiempo de dedicación, cabe señalar que, en general, las tecnologías para el desarrollo web presentan una curva de aprendizaje bastante pronunciada (rápida) para la mayoría de los usuarios.

Por otro lado, para construir aplicaciones nativas se usa el lenguaje de programación propio de cada plataforma. En este caso nos centraremos en los SO móviles más utilizados del mercado, Android y iOS. Para el desarrollo nativo en Android, se utiliza Java, lenguaje de programación orientado a objetos³⁵ desarrollado inicialmente por Sun Microsystems, y actualmente a cargo de la compañía líder en bases de datos, Oracle Corporation. Hoy en día Java es el tercer lenguaje de programación más popular del mundo, en cuanto a uso, solo por debajo de Python y C (Cass, 2017). Java posee licencia pública general de GNU en su tercera versión (en adelante, GNU GPL 3), licencia de código abierto ampliamente utilizada, que garantiza a los usuarios finales poder modificar el software. En relación con la curva de aprendizaje de Java, cabe señalar que la mayoría de los desarrolladores coinciden en que es bastante más lenta, en comparación, por ejemplo, con las tecnologías web. Para poder programar en Java para Android, es necesario implementar, en el entorno donde se trabajará, dos conjuntos de herramientas, Java Development Kit (en adelante, JDK), necesario para desarrollar programas en lenguaje Java en general, y Android SDK, que incluye bibliotecas³⁶, depurador de código, emulador de móvil, documentación y ejemplos de código entre otras características necesarias para desarrollar y testear aplicaciones Android. Con el objetivo de facilitar el acceso a los servicios ofrecidos por su SDK, Android ofrece, como IDE oficial, Android Studio, que incorpora otras funcionalidades para facilitar la programación, como editor inteligente de código, editor GUI y conexión con servicios de Google como Firebase³⁷ y Google Cloud Platform³⁸. Cabe señalar que Android Studio está basado en IntelliJ IDEA, por lo que también es posible utilizar este último IDE para desarrollar aplicaciones nativas con Java. Antes de la existencia de Android Studio el desarrollo Java en Android se realizaba comúnmente utilizando Eclipse IDE y en algunos casos Netbeans. En relación con la documentación relacionada con el desarrollo de aplicaciones nativas con Java, desde su sitio web para desarrolladores³⁹, Android ofrece información técnica, lineamientos de desarrollo, ejemplos y tutoriales para desarrollar aplicaciones nativas utilizando Android Studio IDE y Java como lenguaje de programación. Oracle, por su parte hace lo suyo, entregando, a los desarrolladores que utilicen Java, información relevante sobre esta tecnología y otros servicios relacionados, desde su

³⁵ En programación, la orientación a objetos es una forma de programar, cuyas bases se sustentan en los objetos y en la definición de estos a través de clases. Un objeto representa un determinado concepto y contiene la información necesaria para definirlo, es decir posee propiedades y comportamientos concretos.

³⁶ En computación, una biblioteca corresponde a un pequeño archivo, que posee un conjunto de funcionalidades muy concretas, necesarias para el desarrollo de software, y que proporciona servicios a otros programas.

³⁷ Google. *Firebase*. <<https://firebase.google.com/>>. [Consulta: 04/11/2017].

³⁸ Google. *Google Cloud Platform*. <<https://cloud.google.com/>>. [Consulta: 04/11/2017].

³⁹ Android. *Android developers*. <<https://developer.android.com/develop/index.html>>. [Consulta: 04/11/2017].

Help Center⁴⁰. En este centro de ayuda es posible encontrar documentación sobre la sintaxis y la semántica del lenguaje Java, sobre el uso de JDK, entre varios otros tópicos relevantes tanto para usuarios que se inician en este lenguaje como para desarrolladores experimentados. Finalmente, y con respecto a la comunidad de desarrolladores Java, al ser uno de los lenguajes de programación más populares del mundo, esta es muy amplia y es fácil resolver dudas de programación en foros y a través de la visualización de tutoriales.

Si bien desde el nacimiento de Android el lenguaje de programación nativo para este SO ha sido Java, durante el congreso para desarrolladores Google I/O 2017⁴¹, Google hizo oficial la incorporación de Kotlin como lenguaje de programación para el desarrollo de aplicaciones nativas en Android. Kotlin es un lenguaje con licencia de software libre permisiva Apache 2, desarrollado por la compañía JetBrains para interoperar con código Java, por lo que, para utilizarlo en un determinado SO, es necesaria la existencia de la máquina virtual de Java (JVM, por su sigla en inglés)⁴². Con respecto a los IDE para el desarrollo con Kotlin, es preciso señalar que es soportado, mediante el uso de complementos (en inglés *plugins*), por Android Studio, IntelliJ IDEA y Eclipse. Al ser un lenguaje relativamente nuevo, lanzado a principios de 2012, la documentación disponible tiende a ser un poco más escasa que la de las tecnologías descritas hasta ahora, aun así, desde el sitio web de Android para desarrolladores, es posible acceder a información sobre las características básicas de este lenguaje, lineamientos de desarrollo y una serie de tutoriales para comenzar a crear proyectos Kotlin en Android Studio. En el sitio web oficial de Kotlin existe documentación más completa y detallada sobre este lenguaje e incluso es posible probar su funcionamiento desde el propio navegador web⁴³. Debido a lo reciente de Kotlin, su comunidad de desarrollo es aún bastante restringida, aunque esto se compensa con una curva de aprendizaje mucho más rápida que la del propio Java, por ejemplo.

Para el desarrollo nativo en la plataforma iOS se usan tanto Objective-C como Swift. Objective-C es un lenguaje de programación orientado a objetos basado en C, desarrollado originalmente por la corporación StepStone, actualmente liberado bajo licencia GNU GPL 3 y utilizado por Apple Inc., desde sus inicios, tanto para desarrollar aplicaciones iOS y macOS como para la creación de muchas de las bibliotecas esenciales de ambos SO. Posee una sintaxis compleja, sobre todo para programadores que provienen de entornos ajenos al mundo de la orientación a objetos, por lo que se podría señalar que posee una curva de aprendizaje baja (lenta). Swift por su parte es un lenguaje

⁴⁰ Oracle. *Help Center*. <<https://docs.oracle.com/en/>>. [Consulta: 04/11/2017].

⁴¹ Google. *Google I/O`17*. <<https://events.google.com/io/>>. [Consulta: 04/11/2017].

⁴² La JVM es un elemento fundamental de la plataforma Java, ya que permite interpretar y ejecutar instrucciones expresadas en Java Bytecode (un tipo de código binario) generado al utilizar un compilador de lenguaje Java.

⁴³ Kotlin. *Try Kotlin*. <<https://try.kotlinlang.org/>>. [Consulta: 04/11/2017].

de programación multiparadigma⁴⁴, desarrollado por Apple y presentado oficialmente a mediados de 2014, posee licencia de software libre Apache 2, y se presenta, no como sustitutivo, sino como alternativa a Objective-C, por lo que es totalmente compatible con este último. Si bien, la sintaxis de Swift es mucho más ligera que la de Objective-C vale decir, es posible escribir las mismas instrucciones en menos líneas de código, su complejidad de comprensión es bastante más elevada, por lo que para aprenderlo es necesario dedicar más tiempo que a otros lenguajes como el propio Objective-C.

Independientemente de si el desarrollo nativo se efectúa mediante Objective-C o usando Swift, los desarrolladores iOS deben considerar una serie de elementos comunes para llevarlo a cabo. Con respecto al hardware, construir aplicaciones para iOS, implica utilizar un ordenador Mac que permita instalar el software necesario. Dentro de este software, un componente esencial es Cocoa Touch, una API que incluye la biblioteca de ejecución⁴⁵ de Objective-C y los *frameworks* Foundation y UIKit (Apple, 2017). El primero permite representar estructuras de datos complejos y el segundo proporciona los elementos gráficos necesarios para la creación de GUIs. Cocoa Touch forma parte del iOS SDK, que a su vez está incluido dentro del IDE Xcode, conjunto de herramientas de desarrollo, que permite entre otras cosas, la posibilidad de vista previa en tiempo real de lo que se está programando, la incorporación de un emulador de dispositivos iOS y el acceso a una serie de otras características del iOS SDK, que facilitan el proceso de desarrollo de software. Cabe señalar que para publicar aplicaciones móviles en la App Store es necesario contar con una licencia de desarrollador Apple, la cual tiene asociado un coste monetario anual. Con respecto a la documentación para desarrollo nativo iOS, Apple pone a disposición de sus desarrolladores toda la información necesaria, como tutoriales, guías, videos, códigos de ejemplo, etc., a través del iOS Dev Center⁴⁶. Además, desde este sitio web es posible llevar a cabo la descarga de Xcode y todos los recursos necesarios para construir aplicaciones para iOS, sin embargo, es necesario mencionar que, para acceder al iOS Dev Center, es obligatorio contar con un Apple ID. Finalmente, con respecto a la comunidad de desarrollo de Apple en general, es preciso afirmar que, en comparación con otras comunidades que trabajan con software libre o bajo estándares tecnológicos, esta es mucho más acotada. Probablemente, la necesidad de adquirir licencias para poder desarrollar o tener que hacerlo imperativamente sobre hardware Apple son barreras que condicionan e impiden que muchos desarrolladores se decidan a construir software para macOS o iOS. En el caso específico del lenguaje Swift, ocurre algo similar a lo de Kotlin, su comunidad de desarrollo apenas está iniciando, y si bien, luego de su oficialización por parte de Apple, se ha convertido en uno de los lenguajes de mayor adopción, el número de desarrolladores Swift aún es reducido.

⁴⁴ Un lenguaje de programación multiparadigma corresponde a un lenguaje basado en más de un paradigma de programación (filosofía y metodología particular para diseñar soluciones de software), en otras palabras, es aquel que permite desarrollar programas utilizando más de un estilo de programación.

⁴⁵ Apple. *Objective-C runtime*.

<https://developer.apple.com/documentation/objectivec/objective_c_runtime>. [Consulta: 04/11/2017].

⁴⁶ Apple. *Developer*. <<https://developer.apple.com/>>. [Consulta: 04/11/2017].

A continuación, se presenta una tabla resumen con las principales características de los lenguajes de programación explicados en los párrafos anteriores, según las tres categorías para el desarrollo de aplicaciones móviles.

	Aplicaciones web e híbridas	Aplicaciones nativas iOS		Aplicaciones nativas Android	
Lenguaje	HTML, CSS y JS	Objective-C	Swift	Java	Kotlin
Organización encargada del lenguaje	W3C	Apple Inc.	Apple Inc.	Oracle Corporation	JetBrains
Documentación	Abundante / asequible	Abundante/ Focalizada en un solo lugar	Abundante/ Focalizada en un solo lugar	Abundante / asequible	Escasa / asequible
Comunidad de desarrollo	Amplia	Reducida	Reducida	Amplia	Reducida
Curva de aprendizaje	Pronunciada	Baja	Baja	Baja	Media
Licencia	Formato abierto	GNU GPL 3	Apache 2	GNU GPL 3	Apache 2
Principales IDE	Netbeans IDE, PhpStorm y IntelliJ IDEA Ultimate edition	Xcode	Xcode	Android Studio, IntelliJ IDEA y Eclipse	Android Studio, IntelliJ IDEA y Eclipse
Otros requisitos	<i>Framework</i> (apps híbridas)	- Ordenador Mac - Licencia de desarrollador si se quiere publicar - Iphone SDK - Cocoa Touch API	- Ordenador Mac - Licencia de desarrollador si se quiere publicar - Iphone SDK - Cocoa Touch API	- JDK - Android SDK	- JVM - <i>Plugin</i> para IDE

Tabla 3. Resumen de las características de los lenguajes de programación según enfoque de desarrollo.

4.3.2. *Frameworks* para el desarrollo móvil multiplataforma

Facilitar y agilizar el proceso de desarrollo es una de las premisas fundamentales del desarrollo de software actual. En un contexto donde prima siempre la productividad, la reutilización de componentes de software resulta clave para lograr un desarrollo ágil de aplicaciones y, herramientas como los *frameworks* contribuyen a aliviar el exceso de carga asociada a la creación de funcionalidades y elementos gráficos de uso común. Un *framework* corresponde a un esquema o patrón, diseñado para agilizar y “estandarizar” el proceso de desarrollo de software, el cual permite definir una estructura global para la elaboración de aplicaciones. Para lograr esto, normalmente los *frameworks* incluyen un conjunto de bibliotecas, clases, imágenes y otros recursos que facilitan el desarrollo. Programar utilizando estos esquemas permite obtener código ordenado y fácilmente entendible por otros programadores. Aunque en la mayoría de los casos, un *framework* se encuentra ligado a un lenguaje de programación concreto esto no siempre es así, por ejemplo, Cocoa Foundation Framework, además de Objective-C, también admite programación en Python y Ruby. Los *frameworks* pueden usarse tanto para elaborar aplicaciones web, nativas e híbridas, como para desarrollar cualquier otro tipo de software. Por ejemplo, tal y como se señaló anteriormente, con el fin de estandarizar la construcción de aplicaciones nativas para iOS, Apple integra por defecto, en Cocoa Touch API, un conjunto de *frameworks*. En el caso del RWD, también es común el uso de *frameworks web* (WF, por su sigla en inglés), entre los que destacan, por ejemplo, Bootstrap o Foundation Zurb.

En el caso particular del desarrollo de aplicaciones híbridas, junto con la utilización de la tripleta tecnológica del W3C, es fundamental el uso de un *framework* que, además de facilitar la definición de la estructura global de las aplicaciones, posibilite el desarrollo multiplataforma, en otros términos, que permita escribir una sola versión del código fuente de la aplicación para luego ejecutarla en las plataformas que sea necesario. Este tipo de herramientas se conocen como *cross-platform development frameworks* (en adelante, *frameworks*). Es necesario señalar que, si bien la mayoría de los *frameworks* se basan en HTML, CSS y JS, el desarrollo multiplataforma también se puede llevar a cabo usando otros lenguajes de programación, como ocurre con los *frameworks* Xamarin, cuya programación se realiza en lenguaje C#, Qt, herramienta mediante la cual se desarrolla usando lenguaje C++ o MoSync, que admite C y C++.

Los *frameworks* basados en HTML, permiten hacer uso de las características de la quinta especificación de este lenguaje de etiquetas, HTML5, la cual incorpora nuevos elementos estructurales con significado semántico que buscan la creación de sitios web mejor estructurados y fácilmente interpretables por una máquina. Un ejemplo de estos elementos semánticos son las etiquetas <header> y <footer>, la primera permite representar la cabecera visible del documento y la segunda permite definir el pie de página (en inglés *footer*). Otra característica destacable de HTML5 es la integración nativa de reproductores de vídeo y audio. Cabe mencionar que no todas las características ofrecidas por HTML5 y CSS3 son admitidas por los navegadores web actuales y

podrían presentarse problemas de incompatibilidad sobre todo en versiones antiguas de estos navegadores. Por ejemplo, la etiqueta semántica <header>, antes descrita, sólo se encuentra parcialmente soportada en las últimas versiones de navegadores web tan utilizados como Google Chrome o Safari y simplemente no es soportada en ninguna de las versiones del navegador para dispositivos móviles Opera Mini⁴⁷.

Existen multitud de herramientas para el desarrollo móvil multiplataforma con HTML5, a continuación, se describen cuatro de los *frameworks* más ampliamente divulgados entre la comunidad de desarrollo en los últimos años. En 2009, la empresa Nitobi creó el *framework* PhoneGap, con el objetivo de permitir el desarrollo móvil multiplataforma mediante tecnologías web genéricas como HTML, CSS y JS. En septiembre de 2011, Nitobi decidió convertirlo en un proyecto de código abierto, por lo que donó el código fuente de este a la organización sin fines de lucro, Apache Software Foundation (en adelante, Apache). Un mes más tarde la empresa de software, Adobe Systems Incorporated (en adelante, Adobe), compró Nitobi y con ello se hizo dueña de la marca PhoneGap. Con el fin de evitar problemas legales Apache renombró el proyecto de código abierto como Apache Cordova. A partir de lo anterior, es correcto señalar que ambos productos son esencialmente idénticos y que PhoneGap corresponde a una distribución de Apache Cordova.

En lo que se refiere a la utilización de Apache Cordova, para instalarlo en un SO concreto, es necesario contar previamente con el entorno en tiempo de ejecución (en inglés *runtime system*)⁴⁸ bajado en JS Node.js⁴⁹, más específicamente con el gestor de paquetes de este (npm, por su sigla en inglés), el cual permite manejar las dependencias de un determinado proyecto y, además, buscar y reutilizar paquetes de código de diferentes desarrolladores, disponibles en repositorios, de manera similar a lo que ocurre con la herramienta avanzada de empaquetado (APT, por su sigla en inglés) en SO GNU/Linux. Una herramienta relevante, disponible como un paquete npm, es la interfaz de línea de comandos (CLI, por su sigla en inglés), la cual permite llevar a cabo una serie de tareas fundamentales, vinculadas con el desarrollo de cualquier aplicación Apache Cordova como por ejemplo, la creación del directorio de la aplicación y todos los ficheros necesarios (comando *create*), la incorporación de los SDK de cada SO móvil (comando *platform*) o la gestión de los *plugins* que se utilizarán (comando *plugin*). Respecto a este último punto, los *plugins* corresponden a una característica clave del proyecto de Apache, por lo que la compañía pone a disposición de los desarrolladores más de 2800 de estos elementos⁵⁰. Los *plugins* permiten acceder a características nativas de un SO móvil mediante el uso de código JS, como por ejemplo, a los sensores del dispositivo o a la lista de contactos. Es preciso señalar que, si bien existen *plugins* Apache Cordova

⁴⁷ Caniuse. *Header*. <<https://caniuse.com/#search=header>>. [Consulta: 04/11/2017].

⁴⁸ En informática, un entorno en tiempo de ejecución corresponde a un software que proporciona servicios para un programa en ejecución (proceso) pero no es considerado como parte del SO.

⁴⁹ Node. *Acerca de Node.js*. <<https://nodejs.org/es/about/>>. [Consulta: 15/11/2017].

⁵⁰ Apache Cordova. *Cordova plugins*. <<https://cordova.apache.org/plugins/>>. [Consulta: 15/11/2017].

para cubrir la mayoría de las necesidades comunes de desarrollo, en caso de ser necesario, también es posible desarrollar *plugins* propios.

Con respecto a PhoneGap, es preciso mencionar que Adobe ha extendido las características del producto de Apache para diferenciarlo del mismo. Una de estas características extendidas, y fundamentales para el desarrollo de cualquier proyecto multiplataforma HTML5, corresponde al servicio de compilación en la nube, PG Build App. Este servicio permite compilar aplicaciones multiplataforma HTML5 para los principales SO móviles, Android, Windows y iOS, aunque cabe señalar que, para encapsular aplicaciones para iOS, mediante este servicio, es necesario contar con un certificado de desarrollador proporcionado por Apple. PhoneGap también ofrece la posibilidad de integrarse con otros servicios de Adobe como, por ejemplo, con el editor web Dreamweaver CC. Al igual que Apache Cordova, PhoneGap posee su propia CLI, denominada PhoneGap CLI, pero, además como alternativa a esta, dispone de una aplicación de escritorio, llamada PhoneGap Desktop App. Esta aplicación simplifica, tanto el proceso de instalación y configuración, ya que integra todos los componentes necesarios (Node.js, por ejemplo), como el propio desarrollo, mediante el uso de una GUI. Otra característica importante de este *framework* es la aplicación móvil PhoneGap Developer App, la cual agiliza el testeo de las aplicaciones desarrolladas sin necesidad de compilar el código, realizando una conexión cliente servidor⁵¹ entre el ordenador que contiene proyecto PhoneGap (servidor) y un dispositivo móvil determinado (cliente) en el que se visualizará en tiempo real el estado de la aplicación móvil en desarrollo. Al estar basado en Apache Cordova, PhoneGap permite la utilización de cualquiera de sus *plugins*. Finalmente, es preciso señalar que, además de PhoneGap, son varios otros los *frameworks* construidos en base a Apache Cordova, entre los más destacados se encuentran Framework7, Intel XDK y Ionic.

En relación con este último software, a pesar de ser relativamente reciente, ha alcanzado gran popularidad entre los desarrolladores de aplicaciones multiplataforma. Ionic es una herramienta desarrollada inicialmente en 2013 por la compañía estadounidense Drifty Co., sobre el *framework* de código abierto, AngularJS⁵² de Google. Posee licencia de software libre permisiva MIT, la cual permite, entre otras cosas, reutilizar software dentro de proyectos propietarios. Una característica importante de Ionic es que incluye una serie de clases CSS predefinidas, que permiten acceder a componentes gráficos⁵³, reutilizables y personalizables, que agilizan la construcción GUI, como botones, listas y menús. Al estar basado en Apache Cordova, Ionic comparte algunas similitudes con PhoneGap, como que requiere Node.js para ejecutarse, permite la utilización de los *plugins* de Apache, cuenta con su propia CLI, denominada Ionic CLI y posee una aplicación móvil para testeo llamada Ionic DevApp. Pero además del *framework* base, Ionic ofrece otras herramientas que

⁵¹ En diseño de software, el modelo cliente-servidor corresponde a una estructura de sistemas distribuidos que separa las funciones de sus componentes entre, proveedores de servicio, denominados servidores y quienes solicitan y utilizan estos servicios, llamados clientes.

⁵² AngularJS. *What is AngularJS?*. <<https://docs.angularjs.org/guide/introduction>>. [Consulta: 15/11/2017].

⁵³ Ionic. *Components*. <<https://ionicframework.com/docs/components/>>. [Consulta: 15/11/2017].

facilitan el desarrollo multiplataforma agrupadas en la plataforma Ionic Pro, dentro de esta *suite* destacan, Ionic Package, que permite compilar aplicaciones para Android y iOS; Ionic View App, diseñada para facilitar el testeo masivo, permitiendo compartir aplicaciones con usuarios externos como clientes o compañero de trabajo; y Ionic Creator⁵⁴, la cual permite construir aplicaciones directamente desde el navegador web, sin necesidad de tocar código fuente, mediante un editor visual arrastrando y soltando componentes, aunque cabe mencionar que es bastante limitada en su versión gratuita.

Probablemente, uno de los *frameworks* para desarrollo móvil multiplataforma más utilizado y más difundido, en los últimos años, entre la comunidad de desarrolladores sea JQuery Mobile. Esta herramienta fue desarrollada por el equipo de JQuery Foundation y lanzada inicialmente en octubre de 2010 bajo licencia de software libre permisiva MIT. Son dos los componentes fundamentales sobre los que JQuery Mobile fue construido. El primero, corresponde a la biblioteca de código abierto de JS, JQuery⁵⁵, optimizada para el desarrollo web, específicamente para facilitar la programación JS dentro de documentos HTML, ya que simplifica la manipulación del DOM mediante la utilización de selectores. Los selectores son uno de los componentes fundamentales de JQuery, se basan en los selectores CSS, y al igual que estos permiten seleccionar y manipular elementos HTML a través de sus diferentes atributos como nombre (*name*), clase (*class*) o identificador (*id*) (W3School, 2017). El uso de selectores JQuery permite la implementación de eventos⁵⁶ para detectar la interacción del usuario con el documento HTML. Además de los eventos JQuery, JQuery Mobile incorpora eventos específicos para dispositivos móviles⁵⁷, que permiten detectar acciones propias de la interacción con estos, como por ejemplo, el evento “orientationchange”, mediante el cual es posible implementar diferentes funcionalidades según la orientación del dispositivo. Finalmente, otra característica importante de la biblioteca JQuery es su excelente integración con Asynchronous JS And XML (AJAX, por su sigla en inglés), tecnología que permite realizar cambios sobre las páginas, sin la necesidad de recargarlas, por ejemplo, permite activar eventos a partir de interacciones táctiles o realizar transiciones entre páginas sin la necesidad de volver a cargarlas. El segundo componente de JQuery Mobile es la biblioteca JQuery User Interface (en adelante, JQuery UI)⁵⁸, la que incorpora diversos componentes gráficos, efectos visuales e interacciones a través del uso de clases CSS en combinación con JQuery. Para facilitar la maquetación de elementos dentro del documento HTML, JQuery Mobile trabaja el diseño web sobre una cuadrícula flexible (en inglés *grid*)⁵⁹ que permite RWD⁶⁰. Esta estructura permite dividir la pantalla del dispositivo hasta en 5

⁵⁴ Ionic. *Ionic creator*. <<https://creator.ionic.io>>. [Consulta: 15/11/2017].

⁵⁵ Mozilla. *JQuery*. <<https://developer.mozilla.org/en-US/docs/Glossary/jQuery>>. [Consulta: 16/11/2017].

⁵⁶ En programación, los eventos corresponden a las diferentes acciones que un usuario puede realizar al interactuar con un software. Por ejemplo, hacer clic o *tap* sobre un determinado elemento.

⁵⁷ JQuery Mobile. *Events*. <<https://api.jquerymobile.com/category/events/>>. [Consulta: 16/11/2017].

⁵⁸ The JQuery Foundation. *About JQuery UI*. <<http://jqueryui.com/about/>>. [Consulta: 16/11/2017].

⁵⁹ JQuery Mobile. *Grids*. <<http://demos.jquerymobile.com/1.4.1/grids/>>. [Consulta: 16/11/2017].

⁶⁰ JQuery Mobile. *Responsive Web Design*. <<http://demos.jquerymobile.com/1.4.5/rwd/>>. [Consulta: 16/11/2017].

columnas basadas en CSS, sin embargo, esto no se recomienda para pantallas pequeñas, ya que podría dificultar el uso de la aplicación por parte de los usuarios. A diferencia de los *frameworks* descritos anteriormente, JQuery Mobile no requiere de Node.js por lo que su configuración resulta bastante ágil, por esta razón tampoco incluye una CLI. Finalmente, cabe señalar que, un aspecto en el que queda en deuda, en comparación con las herramientas antes descritas, es en la integración de componentes extras que contribuyan al proceso de desarrollo, como aplicaciones de testeo o un servicio de compilación.

	Apache Cordova	Phonegap	Ionic	JQuery Mobile
Organización encargada	Apache Software Foundation	Adobe Systems Incorporated	Drifty Co.	JQuery Foundation
Licencia	Apache 2	Apache 2	MIT	MIT
Servicio de compilación	De terceros	Propio	Propio	De terceros
Requisitos previos	Node.js	Node.js	Node.js	Ninguno
Extras	- <i>Plugins</i> Cordova - Apache Cordova CLI	- PG Build App - PhoneGap CLI - PhoneGap Desktop App - PhoneGap Developer App	- Ionic CLI - Ionic DevApp - <i>Suite</i> Ionic Pro	Jquery UI

Tabla 4. Resumen de las características de los *frameworks* para el desarrollo móvil multiplataforma.

4.4. Decisión de desarrollo

Luego de analizar los diferentes elementos tecnológicos que influyen en el proceso de desarrollo de aplicaciones móviles multiplataforma como SO móviles, lenguajes de programación, IDE, *frameworks*, etc., es correcto afirmar que, ninguna opción de desarrollo es mejor o peor que otra y que la inclinación por una determinada tecnología dependerá tanto del contexto de desarrollo como de las necesidades particulares de cada proyecto. En este caso específico, se ha optado por desarrollar un prototipo de aplicación híbrida, pero enfocada en el SO Android, utilizando el *framework* JQuery Mobile y el IDE NetBeans por las siguientes razones:

- Se realizará el desarrollo de una aplicación híbrida, principalmente porque la construcción de este tipo de software implica, tanto un tiempo como un coste de desarrollo menor al del desarrollo nativo, y al mismo permite obtener mejores resultados, en cuanto a rendimiento y *look and feel*, que el desarrollo de una aplicación web. Además, el desarrollo híbrido permite acceder a casi la totalidad de los sensores disponibles en dispositivos móviles, en caso de que alguna funcionalidad lo requiera.
- El desarrollo híbrido generalmente se lleva a cabo usando los estándares web HTML5, CSS3 y JS, cuya documentación es basta y fácilmente accesible, tienen una comunidad de desarrollo amplia que promueve y respalda su utilización y poseen una curva de aprendizaje pronunciada (rápida).
- Este proyecto se centrará en el desarrollo de una aplicación móvil para el SO Android, ya que es el que actualmente domina el mercado móvil en Latinoamérica con un 81% de penetración (IMS Corporate, 2016). Llevar a cabo un desarrollo híbrido, también deja abierta la posibilidad de generar en el futuro, una versión para otros SO, de manera mucho más rápida que en el caso de tener que crear una versión nativa.
- Se utilizará el *framework* de código abierto JQuery Mobile, principalmente porque, a nivel de diseño, se basa en una cuadrícula flexible, lo que facilita la implementación de un RWD, e incluye una serie de componentes gráficos reutilizables y personalizables que agilizan la construcción de aplicaciones móviles; a nivel de programación, facilita el acceso al DOM mediante el uso de selectores, incorpora eventos específicos para dispositivos móviles, permite la utilización de AJAX; y finalmente su proceso de instalación y configuración es bastante sencillo, ya que básicamente consiste en la descarga del directorio JQuery Mobile desde un repositorio en línea.
- Finalmente, con respecto al IDE, se usará NetBeans, fundamentalmente, porque da soporte a los lenguajes implicados en este desarrollo (HTML, CSS y JS) y, además se trata de un proyecto de código abierto sin restricciones de uso.

5. Análisis de aplicaciones similares

5.1. Panorama internacional

Normalmente, con el fin de identificar una determinada especie, los aficionados a la naturaleza llevan a cabo la observación de animales, libro en mano. Lo mismo ocurre, en el ámbito educacional, cuando se enseña sobre fauna, libros y enciclopedias de animales son los protagonistas. El contexto tecnológico descrito en apartados anteriores permite la posibilidad de trasladar la información contenida en estas obras impresas a aplicaciones móviles, pudiendo utilizarse como un complemento a los libros, pero conformándose también como una alternativa factible para difundir información sobre especies animales, aprovechando las prestaciones técnicas de los dispositivos móviles actuales y también las características propias de las aplicaciones de software, como la posibilidad de agilizar las búsquedas mediante la implementación de buscadores o filtros o la incorporación de recursos multimedia como audios y videos. En diferentes regiones geográficas del planeta existen iniciativas tecnológicas relevantes, tanto a nivel de gobierno como de instituciones, que pretenden concientizar sobre la preservación de la fauna, distribuyendo información de las especies animales a través de aplicaciones móviles. A continuación, se describen algunos de los proyectos más destacados en el área.

5.1.1. Field Guide apps to Australian Fauna

El proyecto Field Guide apps to Australian Fauna⁶¹ es una iniciativa del Gobierno de Australia, en conjunto con los museos más importantes del país, cuyo principal objetivo es acercar la fauna australiana a la comunidad, para fomentar el estudio y la conservación de las especies animales del país oceánico. El proyecto fue lanzado en mayo de 2014, y consiste en un conjunto de 8 aplicaciones móviles que contienen información relevante acerca de más de 2100 animales (mamíferos, aves, reptiles, anfibios, peces e invertebrados) de las distintas zonas geográficas de Australia (Museum Australia, 2014). Field Guide apps to Australian Fauna fue ganador de la categoría *The best of the Web - mobile*⁶² de la conferencia Museums and the Web 2015, organizada anualmente por la compañía Archives and Museum Informatics para reconocer las mejores iniciativas web vinculadas con el sector cultural. En el desarrollo de la *suite* de aplicaciones participaron alrededor de 350 profesionales de todo el país entre los que se encuentran fotógrafos, programadores y expertos en ciencias biológicas. El rol de estos últimos es esencial, ya que contribuyeron al desarrollo de las fichas informativas de cada especie animal, por lo cual las aplicaciones pueden ser consideradas

⁶¹ Museums Victoria. *Field guide to Victorian fauna*.

<<https://play.google.com/store/apps/details?id=au.com.museumvictoria.fieldguide.vic>>. [Consulta: 10/11/2017].

⁶² Museums and the Web Conference. *Field guides to Australian fauna - a suite of eight apps*.

<<http://mw2015.museumsandtheweb.com/bow/field-guides-to-australian-fauna-a-suite-of-eight-apps/>>. [Consulta: 10/11/2017].

fuentes confiables de información. Las aplicaciones son de descarga gratuita y se encuentran disponibles para las plataformas iOS y Android. Cabe destacar que el código fuente de las aplicaciones está disponible en la plataforma de desarrollo colaborativo GitHub, bajo licencia de código abierto con el nombre de Genera Project⁶³, lo que ha permitido que otras entidades lo utilicen como cimiento para el desarrollo de sus propios proyectos, tal es el caso de las aplicaciones móviles para iOS, New Zealand Marine Life⁶⁴ y Grand Canyon National Park⁶⁵.

Con respecto a las funcionalidades de las aplicaciones de esta *suite*, para acceder a las fichas informativas de las especies disponibles, los usuarios tienen tres opciones. Una primera posibilidad, consiste en navegar por los diferentes grupos disponibles en la pantalla de inicio (en inglés *home screen*) de la aplicación, denominada *groups*. Esta pantalla, se divide en cuatro secciones: *vertebrates* (vertebrados), *terrestrial invertebrates* (invertebrados terrestres), *freshwater invertebrates* (invertebrados de agua dulce) y *marine invertebrates* (invertebrados marinos). Cada una de estas secciones posee una serie de grupos de animales, por ejemplo, la sección *vertebrates* se compone por los grupos *birds* (aves), *mammals* (mamíferos), *fishes* (peces), *frogs and toads* (ranas y sapos), *lizards* (lagartijas), *snakes* (serpientes) y *turtles* (tortugas). Al ingresar en alguno de los grupos disponibles se desplegará una lista con el nombre y una fotografía de distintos animales pertenecientes a ese grupo, ordenados alfabéticamente por subcategoría. Por ejemplo, al ingresar en la categoría *mammals*, se visualizarán las subcategorías *bats* (murciélagos), *cats* (felinos), *deer* (ciervos), etc.

Una segunda alternativa es navegar a través del listado disponible en la pantalla *all species*, en el cual aparecen todas las especies animales disponibles en la aplicación, ordenadas alfabéticamente. Esta es una opción apropiada para usuarios que deseen explorar entre las especies y no busquen un animal en específico.

⁶³ Museums Victoria. *Documentation for Museum Victoria's genera project*. <<https://github.com/museumsvictoria/genera>>. [Consulta: 10/11/2017].

⁶⁴ Auckland Museum. *New Zealand marine life*. <<https://itunes.apple.com/us/app/new-zealand-marine-life/id661468952>>. [Consulta: 10/11/2017].

⁶⁵ ParksConnect. *Grand Canyon National Park field guide*. <<https://itunes.apple.com/us/app/grand-canyon-national-park/id654819217>>. [Consulta: 10/11/2017].

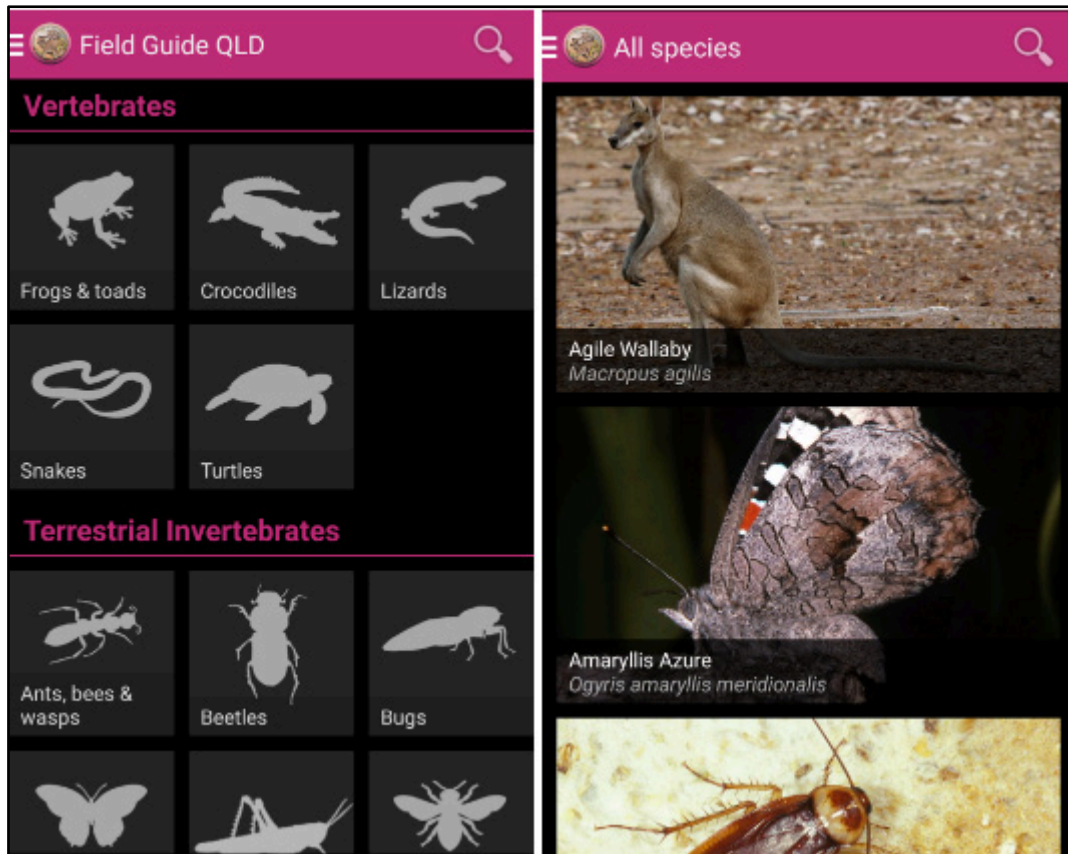


Fig. 5. Grupos de animales de la *home screen* y listado de especies de la pantalla *all species*.

Una última opción para acceder a la ficha informativa de algún animal es utilizar el buscador de la aplicación, disponible tanto en la pantalla *groups* como en la pantalla *all species*, el cual permite al usuario realizar consultas escritas y búsquedas por voz, ya que integra el servicio Google Voice Search. Este buscador cuenta además con constructores de consulta (en inglés *query builders*), que ayudan al usuario a crear su consulta a medida que ingresa texto en la caja de búsqueda. Esta es una manera rápida para encontrar un animal en particular.

Independientemente de la opción utilizada, al escoger algún animal, se despliega una ficha con tres pestañas (en inglés *tab*) con información acerca de este. En la primera *tab*, *details* (detalles), se muestra una o más fotografías del animal y se entregan datos como nombre común y científico de la especie, características físicas, alimentación, comportamiento y clasificación científica. La segunda *tab*, *habitat* (hábitat), permite visualizar, mediante un mapa, la distribución territorial del animal, además se indica si es una especie originaria o introducida en el país. Por último, en la *tab scarcity* (escasez) se indica el grado de amenaza o riesgo de extinción en que se encuentra la especie.

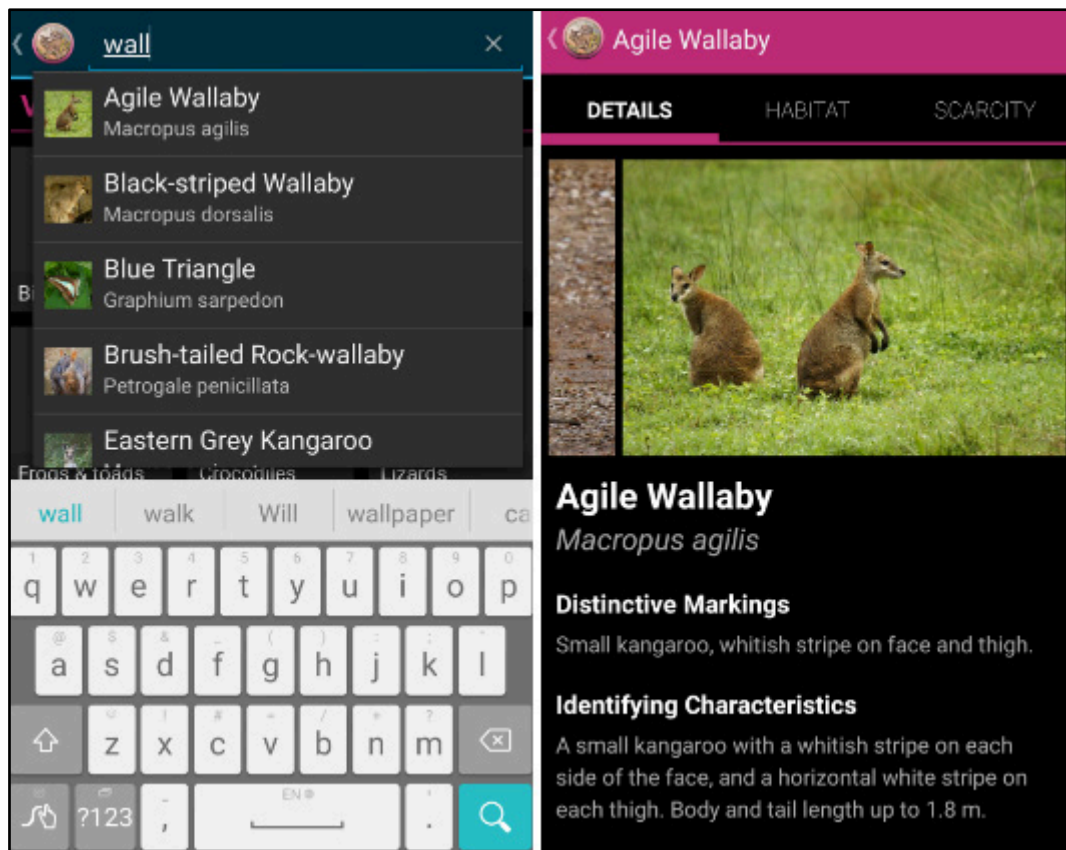


Fig. 6. Query builder del buscador y ficha informativa de la especie escogida.

Finalmente, un aspecto técnico que resulta relevante de mencionar, sobre el proyecto Field Guide apps to Australian Fauna, es que cada una de las aplicaciones de la *suite* tiene un peso aproximado de entre 180 y 200 MB luego de su instalación, tamaño considerable para aplicaciones de tipo divulgativo o informativo, considerando que por ejemplo, la aplicación Android de Wikipedia pesa aproximadamente 43 MB post instalación. Lo anterior, se debe principalmente a la gran cantidad de fotografías de animales que incorpora la aplicación, las que se almacenan localmente. Sin embargo, un aspecto positivo de almacenar todo el contenido multimedia de manera local es que se puede prescindir de una conexión permanente a Internet para consultarlo. Lo que, claramente favorece la movilidad del usuario y en el caso de una guía de campo sobre animales, resulta sumamente útil cuando este está en terreno y no cuenta con una buena conexión a la Red o simplemente no dispone de una.

5.1.2. Guía de aves e itinerarios ornitológicos de España

Otro proyecto en este mismo ámbito, pero enfocado en una especie concreta de vertebrados, es la aplicación móvil, Guía de las aves e itinerarios ornitológicos de España⁶⁶, desarrollada en conjunto

⁶⁶ SEO/BirdLife. *Aves de España*. <<https://play.google.com/store/apps/details?id=com.alborgis.seo>>. [Consulta: 15/11/2017].

por la Sociedad Española de Ornitología (actualmente SEO/BirdLife)⁶⁷ y la Fundación BBVA⁶⁸. La aplicación fue lanzada en 2014, está basada en el proyecto web de 2008, La enciclopedia de las aves de España⁶⁹ y se encuentra disponible de manera gratuita para plataformas Android y iOS. El objetivo de este proyecto es que las personas “comunes y corrientes”, no necesariamente ornitólogos, conozcan las aves existentes en España y sean capaces de identificarlas al escuchar su canto o verlas fugazmente. Para lograr esto, la aplicación móvil permite el acceso a fichas informativas de 563 especies de aves, que de forma habitual u ocasional están presentes en el territorio español, estas fichas incluyen información acerca del comportamiento, la distribución y el estado de conservación de las especies. El contenido textual disponible en la aplicación se acompaña de numerosos recursos sonoros y gráficos como cantos, vídeos, fotografías, ilustraciones y mapas. La aplicación se construyó con datos basados en información científica publicada, y en los conocimientos facilitados por una amplia red de ornitólogos colaboradores (SEO/BirdLife, 2014), pero también gracias al rol fundamental que jugaron los más de 3000 aficionados que, de forma voluntaria, contribuyen con los distintos proyectos de SEO/BirdLife. Se trata, por lo tanto, de un proyecto colectivo que impulsa el concepto de ciencia ciudadana⁷⁰, y que permite de esta manera agrupar información que, de otra manera, sería prácticamente imposible de obtener.

En relación con las funcionalidades de la aplicación, desde la *home screen* es posible acceder a los apartados guía de aves e itinerarios ornitológicos. El primero, se denomina guía de aves y consiste en una pantalla, que contiene un listado con las especies de aves disponibles en la aplicación. Para facilitar la búsqueda de una determinada especie, la pantalla guía de aves permite al usuario aplicar una serie de filtros que reordenan o acotan los resultados de búsqueda. Estos filtros permiten ordenar por nombre común, por nombre científico o por familia⁷¹, o mostrar todas las aves, sólo las habituales (esta es la vista por defecto) o sólo las ocasionales. Esta pantalla también cuenta con un buscador que acota automáticamente los resultados de búsqueda del listado de aves a medida que el usuario ingresa texto.

Al acceder a la ficha de una determinada ave se despliega, en el área visible de la pantalla, una ilustración de la especie y una serie de íconos que permiten acceder a fotografías, mapa de distribución, sonidos de canto y videos. Además, se muestra el nombre común y el nombre científico

⁶⁷ SEO/BirdLife. *¿Quiénes somos?*. <<https://www.seo.org/conocenos/quienes-somos/>>. [Consulta: 15/11/2017].

⁶⁸ Fundación BBVA. *Sobre la Fundación*. <<https://www.fbbva.es/sobre-fundacion-bbva/>>. [Consulta: 15/11/2017].

⁶⁹ SEO/BirdLife. *La enciclopedia de las aves de España*. <<http://www.encyclopediadelasaves.es/>>. [Consulta: 15/11/2017].

⁷⁰ Se entiende por ciencia ciudadana a las investigaciones que se lleva a cabo involucrando al público en general en actividades de carácter científico y que fomentan la contribución activa de los ciudadanos ya sea mediante esfuerzo intelectual o soporte de recursos o herramientas.

⁷¹ En biología, la familia corresponde a una categoría de clasificación de los seres vivos, que busca diferenciarlos a partir de rasgos comunes o relaciones de parentesco. La familia se encuentra ubicada entre el orden (superfamilia) y el género (subfamilia).

del ave y los meses en los que es posible encontrarla en territorio español, indicando además si se trata de una especie residente o si está de paso. Al deslizarse hacia abajo en esta pantalla es posible acceder a cinco diferentes *tabs* con información textual sobre la especie como descripción general, distribución en España y el mundo, hábitat y se reproducción, amenazas y estado de conservación y especies similares por canto o por aspecto.



Fig. 7. Filtros disponibles desde el listado de aves y ficha informativa de la especie escogida.

El segundo apartado de la *home screen* se llama itinerarios ornitológicos y permite acceder a la pantalla itinerarios, en la cual se muestra en la parte superior, un mapa digital con los diferentes enclaves ornitológicos, o puntos de avistamiento de aves, de las distintas provincias de España. Además, mediante el sensor GPS del terminal, el usuario puede ver su ubicación dentro del mapa, identificando así los lugares de avistamiento más cercanos a su posición. Esta pantalla ofrece además, bajo el mapa digital, un listado en orden alfabético, con los enclaves ornitológicos de España, que permite navegar entre estos y acceder directamente a la información de cada uno.

Al acceder a un determinado enclave se visualiza una fotografía del lugar y al igual que en el caso de la ficha de las aves, es posible navegar por cinco diferentes *tabs* que permiten el acceso a datos generales del enclave, una galería de imágenes, rutas o itinerarios para la observación de aves, especies de aves que se encuentran en el enclave y una funcionalidad de realidad aumentada para la orientación de puntos referenciados mediante GPS.



Fig. 8. Mapa digital y listado de enclaves ornitológicos de la pantalla itinerarios y ficha informativa del enclave escogido.

Un aspecto técnico negativo respecto de esta aplicación es la necesidad de conexión permanente a Internet para acceder a los diferentes recursos multimedia y utilizar las diferentes funcionalidades vinculadas con el sensor GPS del dispositivo. También cabe señalar que, a pesar de que el contenido multimedia se almacena en un servidor web, la aplicación pesa más de 300 MB luego de su instalación en local. Una característica destacable de la Guía de las aves e itinerarios ornitológicos de España, es que las fichas informativas de las 363 especies de aves habituales, se encuentran disponibles en todas las lenguas oficiales del Estado Español (castellano, catalán, euskera y gallego) y además en Inglés, favoreciendo así el acceso de las minorías lingüísticas del país, a la información contenida en esta aplicación. Finalmente, un aspecto interesante vinculado con este proyecto de SEO/BirdLife es que, desde el año 2016, la información de La enciclopedia de las aves de España, también se encuentra disponible a través de una aplicación web denominada Identificador de aves⁷², la cual, mediante diferentes filtros como distribución, tamaño, color o hábitat, permitiendo reconocer el ave que se está observando.

⁷² SEO/BirdLife. *Identificador de aves*. <<https://www.seo.org/identificador-aves/>>. [Consulta: 20/11/2017].

5.2. Panorama nacional

Tal y como se mencionó en el apartado Justificación, luego de realizar determinadas consultas clave en las tiendas de aplicaciones Google Play y App Store, fue posible corroborar que actualmente, no existe en Chile una aplicación móvil que entregue información sobre las especies animales del país. Sin embargo, tras realizar búsquedas más específicas, sobre especies animales como mamíferos, aves, reptiles o peces, fue posible evidenciar la existencia de iniciativas relacionadas con el mundo de las aves, pero a diferencia de los proyectos antes descritos, en cuyo desarrollo se involucran decenas o cientos de profesionales y existe además el respaldo de grandes entidades gubernamentales o culturales e inclusive de la propia ciudadanía, estas iniciativas corresponden a pequeños desarrollos llevados a cabo por particulares aficionados a las aves. Las aplicaciones móviles *Aves Chile*⁷³ y *Aves chilenas*⁷⁴ son aplicaciones móviles para terminales Android, cuyo objetivo es entregar información acerca de las aves que habitan Chile.

Por una parte, *Aves Chile* permite seleccionar, desde la *home screen*, entre 25 especies de aves. Al escoger un ave se accede a una nueva pantalla en la que se observa una fotografía del animal e información textual correspondiente a nombre común, nombre científico, alimentación, distribución, tamaño, lugares en que habita y finalmente es posible reproducir un sonido con el canto del ave.

Por otra parte, la aplicación móvil *Aves chilenas*, tiene un estilo de diseño de una sola página (en inglés, *one page*), en este caso una única pantalla, donde se integra todo el contenido. La aplicación incorpora información acerca de 12 especies de aves de Chile, que se encuentran divididas en las categorías aves urbanas, aves nocturnas y aves rapaces. Cada categoría cuenta con una breve descripción y además se muestra una fotografía del ave, textos con el nombre común, el nombre científico y una pequeña descripción del ave y también un reproductor con el canto del ave.

⁷³ Concha, Juan. *Aves Chile*.

<https://play.google.com/store/apps/details?id=appinventor.ai_j_conchav0685.AvesChile>. [Consulta: 20/11/2017].

⁷⁴ Henríquez, Jorge. *Aves chilenas*.

<https://play.google.com/store/apps/details?id=appinventor.ai_jorge.AVESCHILENAS>. [Consulta: 20/11/2017].



Fig. 9. De izquierda a derecha, ficha de especie de la aplicación Aves Chile y ficha de especie de la aplicación Aves chilenas.

Cabe señalar que el contenido textual de ambas aplicaciones es de autoría de los propios desarrolladores por lo que la información incluida podría resultar errónea o imprecisa. También es necesario indicar que ninguna de las dos aplicaciones descritas posee buscador o funcionalidades que permitan realizar algún tipo de filtro, aunque dada la pequeña cantidad de especies disponibles en cada una probablemente estas herramientas no sean necesarias. Finalmente, respecto al tamaño de ambas aplicaciones, este no supera los 20 MB, principalmente debido a la poca cantidad de recursos multimedia que ambas incluyen.

Un tercer proyecto chileno vinculado al mundo de las aves, corresponde a la aplicación móvil BuscAves⁷⁵. Este proyecto, actualmente en fase de desarrollo, corresponde a una iniciativa de un grupo de profesionales de la Facultad de Ciencias Veterinarias y Pecuarias (en adelante, FAVET)⁷⁶ de la Universidad de Chile, cuyo principal objetivo es facilitar, a los aficionados al avistamiento de aves, la identificación de las especies. Además, el proyecto pretende servir como herramienta educativa por lo que una vez finalizada la fase de desarrollo, se llevará a cabo un proceso de difusión en

⁷⁵ Universidad de Chile. *Casa de Bello crea la primera aplicación chilena para la identificación de aves.* <<http://www.uchile.cl/noticias/138376/crean-la-primera-aplicacion-chilena-para-la-identificacion-de-aves>>. [Consulta: 25/11/2017].

⁷⁶ Universidad de Chile. *Facultad de Ciencias Veterinarias y Pecuarias.* <<http://www.veterinaria.uchile.cl/>>. [Consulta: 25/11/2017].

diferentes establecimientos educacionales del país. En relación con el funcionamiento de la aplicación, mediante distintos filtros, como tamaño o color del ave, forma del pico o de las patas, estación del año y zona geográfica del país, el usuario podrá acotar el listado de especies disponibles hasta obtener una serie de alternativas que le permitan discernir de qué ave se trata, de manera similar al proyecto Identificador de aves de SEO/BirdLife, descrito en párrafos anteriores. Respecto a cuáles serán las especies disponibles en la aplicación, los profesionales de FAVET pretenden formar una base de datos con las aves más comunes o representativas del país. El proceso de construcción de esta base de datos se desarrollará en colaboración con instituciones como el Parque Metropolitano, el Museo de Historia Natural y el Comité Nacional Pro Defensa de la Flora y Fauna. Además, la información disponible en la aplicación podrá complementarse accediendo a un portal web que dispondrá de datos adicionales sobre las especies. Por último, cabe señalar que FAVET, pretende que la aplicación móvil esté disponible para descargarse de forma gratuita y funcione sin necesidad de conexión a Internet.

5.3. Tendencias respecto a la identificación de especies

Finalmente, con respecto al futuro de las aplicaciones móviles para la identificación de animales, probablemente la tendencia conduzca a que el reconocimiento de una determinada especie se lleve a cabo, en terreno, haciendo uso de los sensores de los dispositivos móviles, como la cámara o el micrófono. Lo anterior es claramente posible dadas las prestaciones técnicas actuales de los dispositivos móviles y ya se está aplicando para la identificación de aves y otros elementos del entorno natural. A continuación, se describen brevemente algunas de estas iniciativas.

El proyecto Leafsnap⁷⁷ corresponde a un desarrollo conjunto entre la Columbia University, la University of Maryland, y la Smithsonian Institution. Esta aplicación móvil permite, basándose en los mismos principios del reconocimiento facial, identificar una determinada especie de árbol a partir del reconocimiento visual de alguna sus hojas. Para lograr lo anterior, es necesario que el usuario tome una fotografía de la hoja, contra una superficie blanca. Actualmente solo se encuentra disponible para la plataforma iOS e incluye información sobre árboles que se encuentran en el noreste de los Estados Unidos y en Canadá, aunque también existe una versión con árboles de Reino Unido (Leafsnap UK).

PeakVisor⁷⁸ es una aplicación de pago desarrollada por la empresa Routes Software SRL y disponible para plataformas Android, iOS y que también cuenta con una versión web. Esta aplicación de realidad aumentada (AR, por su sigla en inglés)⁷⁹ permite identificar montañas, y otras elevaciones

⁷⁷ Leafsnap. *Leafsnap: An Electronic Field Guide*. <<http://leafsnap.com/>>. [Consulta: 25/11/2017].

⁷⁸ Routes Software SRL. *PeakVisor*. <https://peakvisor.com/index_es.html>. [Consulta: 25/11/2017].

⁷⁹ La realidad aumentada es una tecnología que permite la incorporación de elementos virtuales, en tiempo real, dentro del universo de elementos físicos tangibles. De esta manera, es posible, entregar al usuario información añadida sobre los elementos a su alrededor, a través de un dispositivo como un móvil o gafas de realidad aumentada por ejemplo.

naturales del terreno, a través de la cámara del dispositivo móvil. Haciendo uso del sensor GPS y del magnetómetro del dispositivo, PeakVisor determina la ubicación del usuario y luego, solo basta con apuntar una determinada montaña, con la aplicación abierta, para obtener información esencial de esta como su nombre o su altura. Puede ser utilizada de manera offline ya que permite almacenar la información de las montañas en local. Respecto de su alcance y cobertura, PeakVisor cuenta con información acerca de la mayoría de las cadenas montañosas del planeta.

Warblr⁸⁰ corresponde a un proyecto de ciencia ciudadana para aficionados a las aves, desarrollado por la empresa Glastonbridge Software Ltda. y respaldado por investigadores de la Universidad Queen Mary de Londres. Esta aplicación es conocida popularmente como el Shazam⁸¹ de las aves, ya que, su funcionamiento es similar al de este servicio de identificación de música, y permite, usar el micrófono del dispositivo móvil para grabar el sonido de un ave cercana e identificarla entre una lista de posibles candidatos, entregando imágenes y descripciones acerca de estos. Se encuentra disponible, previo pago por descarga, tanto desde Google Play como de App Store y por ahora permite el reconocimiento de más de 220 especies de aves británicas a través de su canto.

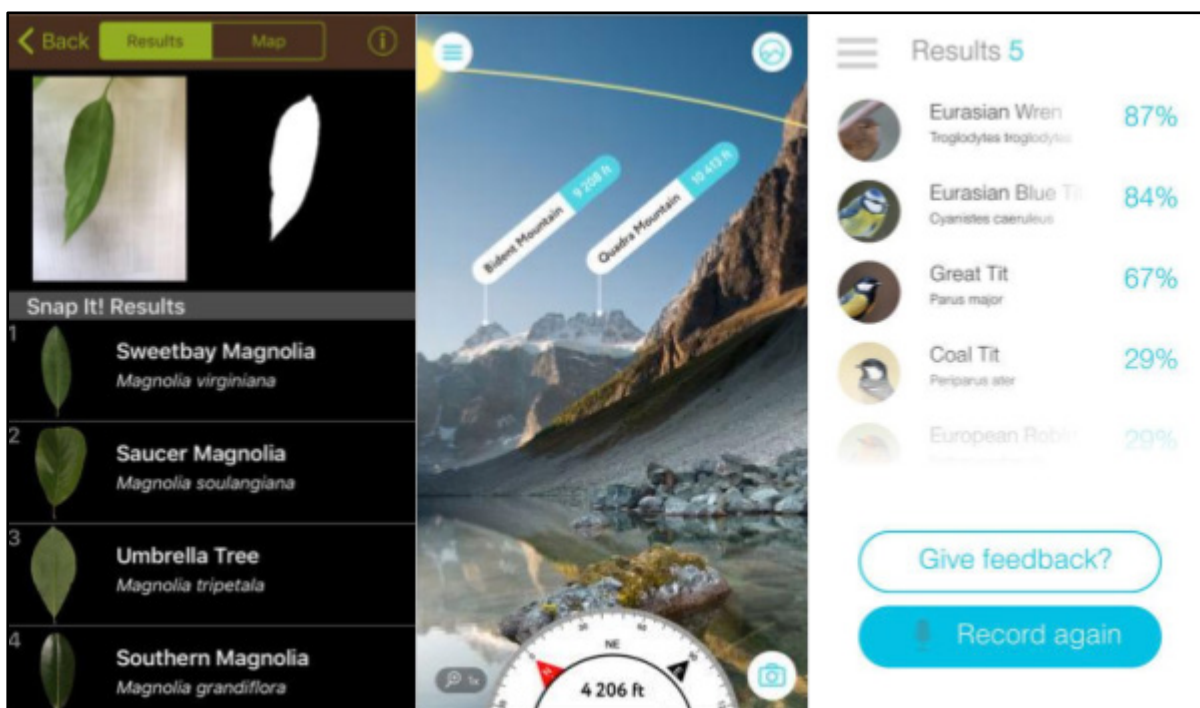


Fig. 10. De izquierda a derecha, capturas de pantalla de Leafsnap, PeakVisor y Warblr. Fuente: App Store (2017).

⁸⁰ Glastonbridge Software. *Warblr*. <<https://warblr.net/>>. [Consulta: 25/11/2017].

⁸¹ Shazam. *Sahazam apps*. <<https://www.shazam.com/es/apps>>. [Consulta: 25/11/2017].

6. Desarrollo de la aplicación

6.1. Diseño

Luego de detectar la ausencia de una aplicación móvil sobre fauna chilena, analizar el contexto tecnológico actual, definir las tecnologías a utilizar para llevar a cabo el desarrollo y revisar el estado de las aplicaciones móviles relacionadas con el estudio y la identificación de especies animales, el siguiente paso lógico corresponde a la etapa de diseño del prototipo de aplicación móvil señalada anteriormente.

6.1.1. Arquitectura de la información

La arquitectura de la información (en adelante AI) es una disciplina que se encarga de estructurar, organizar y etiquetar los elementos que conforman los entornos informacionales, para facilitar de esta manera la localización (o el acceso) de la información contenida en ellos y mejorar, así, su utilidad y su aprovechamiento por parte de los usuarios (Pérez-Montoro, 2010). En otras palabras, la AI se centra en el estudio y en el análisis de un determinado entorno informacional en relación con la presentación de sus elementos y sus contenidos al visitante, con el fin de satisfacer de manera efectiva sus necesidades de información. De esta manera, la AI es un factor determinante sobre la capacidad, de un entorno informacional, de permitir al usuario encontrar la información que busca, es decir, la AI permite diseñar entornos que resulten de utilidad a los usuarios. En este contexto, un entorno informacional corresponde a un medio a través del cual se entrega o distribuye información sobre un determinado tema. Tanto sitios web como aplicaciones móviles se integran perfectamente dentro de esta definición. Cada uno de estos medios de distribución de información, tendrá sus características particulares según el público objetivo al que se dirigen y los objetivos comunicativos propios de cada proyecto. En el caso específico de las aplicaciones móviles la AI permite definir la estructura que, de mejor manera, permita al usuario asimilar el contenido de la aplicación⁸².

Una buena arquitectura de información está basada en tres pilares, el contexto organizacional en el que se desarrolla, el contenido que alberga y los usuarios que la visitan y consultan (Morville; Rosenfeld, 2006). A continuación, se lleva a cabo una descripción general de los tres pilares fundamentales de la IA, aplicados al prototipo de aplicación móvil que se pretende desarrollar.

Con respecto al contexto, si bien en este caso concreto el objetivo del proyecto es llevar a cabo el desarrollo de una aplicación prototipo sobre fauna chilena, se pretende también sentar las bases estructurales que puedan ser aplicadas en el desarrollo de cualquier aplicación móvil que tenga un enfoque divulgativo y que no necesariamente se centre en especies animales, ya que se entiende

⁸² En adelante, y considerando la naturaleza de este TFM, se tratarán los componentes de la AI desde la perspectiva de aplicaciones móviles, entendiendo que la AI es aplicable a cualquier entorno informacional, no sólo del ámbito digital.

que una iniciativa con estas características puede ser extrapolable a otros contextos, como por ejemplo, la identificación de otros seres vivos como plantas, hongos, etc.

En relación con el contenido y las funcionalidades de la aplicación móvil, y considerando la revisión de iniciativas, llevada a cabo en el apartado Estado del arte, como mínimo, una aplicación móvil sobre fauna chilena, debería contar con, una *home screen* que permita al usuario navegar por los diferentes grupos de especies disponibles, una pantalla con un listado de todas las especies animales de la aplicación móvil; una ficha informativa de cada animal, que contenga imágenes e información básica sobre estos; un sistema de navegación que permita al usuario moverse entre las diferentes pantallas de la aplicación; algún mecanismo que posibilite a los usuarios encontrar de manera rápida el animal buscado; y alguna funcionalidad que permita acotar el listado total de especies o los resultados de búsqueda.

Finalmente, en referencia a los usuarios finales de este proyecto, si bien la implementación de una aplicación móvil de este tipo debería estar dirigida a la comunidad chilena en general, se pretende que el público objetivo sean los aficionados al avistamiento de especies animales en terreno. Aunque, también se le podría otorgar un enfoque más educativo y utilizarla como herramienta formativa en instituciones educacionales del país.

6.1.1.1. Componentes de la AI

Los componentes esenciales que, de manera articulada, permiten estructurar cualquier entorno informacional son los sistemas de organización, los sistemas de etiquetado, los sistemas de navegación, los sistemas de búsqueda y los lenguajes documentales. Todos estos componentes contribuyen a la “encontrabilidad” de contenidos (en inglés, *findability*) de un entorno digital, la cual corresponde a la capacidad que poseen los contenidos digitales de poder ser localizados y navegados para permitir descubrir y recuperar la información relevante que contienen (Morville, 2005).

Un primer componente de la AI son los sistemas de organización, estos permiten dividir o clasificar el contenido de un entorno concreto en grupos de entidades o elementos. Los sistemas de organización se dividen en esquemas exactos y ambiguos. Por una parte, los esquemas de organización exacta se refieren a aquellos que tienen sólo una interpretación, ya que se encuentran claramente definidos, son excluyentes entre sí y permiten hacer cosas previamente conocidas. Los esquemas de organización exactos pueden ser alfabéticos, como por ejemplo, un directorio de un sitio web o un diccionario en línea; cronológicos como blogs o periódicos; y geográficos como portales comerciales que muestran determinados contenidos según la ubicación del usuario. Por otra parte, los esquemas de organización ambiguos o subjetivos corresponden a aquellos esquemas de organización subjetiva que pueden basarse en diferentes criterios. Dentro de estos esquemas el más habitual y también el más subjetivo es aquel que permite organizar los contenidos en base a diferentes temáticas, los esquemas ambiguos permiten organizar la información por tareas o

funcionalidades, por ejemplo, las diferentes tareas que un usuario con rol docente puede llevar a cabo desde un Sistema de Gestión de Aprendizaje (LMS, por su sigla en inglés); por audiencia, según el público objetivo del proyecto; por metáforas, por ejemplo, utilizando símbolos o iconos; y finalmente, se encuentran los esquemas híbridos, que corresponden a una combinación de dos o más de los esquemas antes mencionados.

Además de definir la manera en que se organizarán los contenidos de un entorno informacional, los sistemas de organización permiten determinar la forma de acceso a esta información. Lo anterior se lleva a cabo por medio de las estructuras de organización, las que permiten organizar los grupos de elementos resultantes de los esquemas, mostrando las dependencias lógicas que existen entre ellos. Una estructura de organización puede ser jerárquica, si existe una relación de jerarquía o taxonomía⁸³ entre sus componentes; basada en registros si los contenidos se encuentran disponibles como si se tratase de una base datos; hipertextual o en red, si posible navegar entre contenidos por medio de vínculos o enlaces entre ellos; secuencial si la información se consume como una línea de contenidos, a través de un único itinerario de consulta y con un principio y un final bien definido; y en tabla o matriz, si es que los contenidos se presentan a través de estas estructuras. Finalmente, cabe señalar que ninguna de estas estructuras de organización es excluyente entre sí.

Respecto a los sistemas de organización, en el caso específico de esta aplicación móvil sobre fauna, se propone utilizar un esquema de organización temático para los contenidos de la *home screen*, organizando a los animales disponibles en grupos. Por ejemplo, si la aplicación se centrara específicamente en animales vertebrados, podrían generarse los grupos temáticos, anfibios, aves, mamíferos, peces y reptiles. Lo anterior también podría derivar en la utilización de un esquema híbrido que, además de la clasificación temática, permitiera una clasificación por metáforas, mediante el uso de íconos o imágenes de animales representativos de cada uno de los grupos mencionados. Los esquemas de organización temático también podrían ser aplicados en la ficha informativa de cada especie, en dónde se podría clasificar el contenido a partir de diferentes rasgos de las especies y su forma de vida, como por ejemplo, características físicas, hábitat y el estado de conservación. Para la pantalla que contenga el listado de especies se sugiere emplear un esquema de organización alfabético, ya que es el tipo de ordenación que, algunos estudios, destacan como el que permite encontrar un elemento con mayor rapidez (Card, 1984) y, además, probablemente resulte reconocible y entendible para los usuarios finales.

En relación con la estructura de organización y considerando el uso de un esquema temático para clasificar los animales, la utilización de una estructura jerárquica basada en las categorías taxonómicas de las especies existentes en el planeta⁸⁴, parece una alternativa lógica, ya que

⁸³ Una taxonomía corresponde a la clasificación de grupos de elementos que poseen características comunes.

⁸⁴ En biología, las categorías taxonómicas, corresponden a los grupos de clasificación jerárquica de los seres vivos. En orden descendente se encuentran: Dominio, Reino, División o Filo, Subfilo, Clase, Orden, Familia, Género y Especie.

permitiría distintos niveles de profundidad (crecimiento vertical) y de amplitud (crecimiento horizontal), dependiendo del nivel de detalle y del alcance temático de la aplicación. Además, probablemente esta es la forma de organización que un especialista en especies animales esperaría encontrar al navegar por una aplicación móvil con estas características. Con el objetivo de mostrar una propuesta de los componentes de AI de la aplicación móvil sobre fauna, se usarán *blueprints*. Los *blueprints* son diagramas que permiten mostrar las relaciones entre páginas y otros componentes de contenido, y se pueden utilizar para representar sistemas de organización, navegación y etiquetado (Morville; Rosenfeld, 2006). Se presenta a continuación un *blueprint* sobre la estructura organizacional de los contenidos de la aplicación, basada en las categorías taxonómicas animales, dónde las categorías “Clase” y “Especie” permitirían organizar el contenido de las pantallas home screen y listado de especies respectivamente.

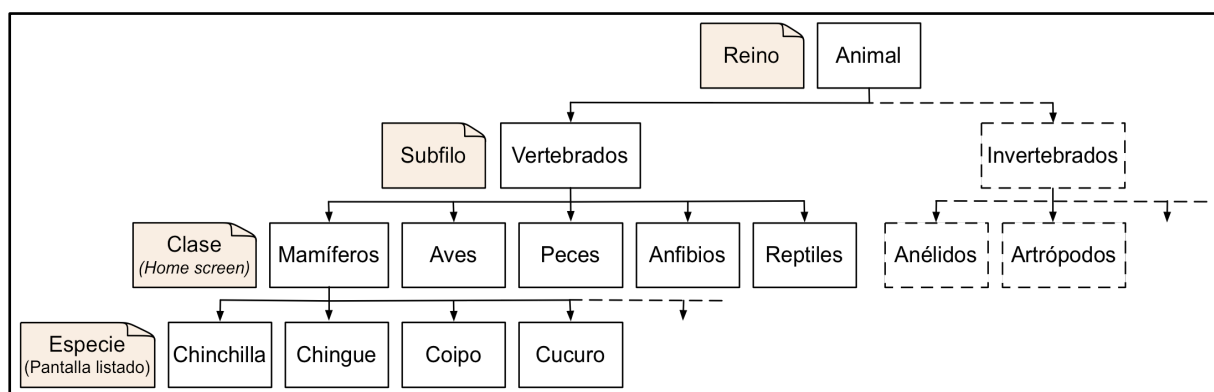


Fig. 11. *Blueprint*: estructura de organización de la aplicación, basada en las categorías taxonómicas animales.

Un segundo componente esencial de la AI son los sistemas de navegación. Un sistema de navegación es aquel que posibilita ordenar y agrupar los contenidos de un entorno informacional bajo categorías, permitiendo al usuario desplazarse a través de la arquitectura creada. El sistema de navegación debe lograr que el usuario comprenda, dónde se encuentra, hacia dónde puede ir y cómo se organizan los contenidos de la aplicación móvil que está utilizando. Una consideración importante respecto a los sistemas de navegación es que siempre deben ser constantes y aparecer en el mismo lugar, en todas las pantallas (Morville; Rosenfeld, 2006). Existen tres tipologías de sistemas de navegación, integrados, complementarios y avanzados. Los sistemas de navegación integrados son aquellos que están estrechamente vinculados con la estructura de la aplicación y que al eliminarlos cambiaría el aspecto visual de esta. Dentro de los sistemas integrados, se encuentran, el constante o principal, que se encontrará presentes en toda (o en casi toda) la aplicación; los locales o sectoriales que corresponden a sistemas de navegación específicos para determinadas pantallas; y los contextuales que permiten identificar contenidos relacionados y dirigirse hacia ellos, ya sea en otra pantalla o afuera de la aplicación (contenido relacionado), o en una subsección dentro de la pantalla visitada (contenido incrustado). Típicamente los sistemas de navegación contextuales corresponden a enlaces. Los sistemas de navegación complementarios son aquellos que, tal como su nombre lo indica, facilitan al usuario la navegar por los contenidos. Claros ejemplos de sistemas

complementarios son los mapas de sitio web, los índices alfabéticos (listado de términos que representan el contenido de un sitio web) o las guías tutoriales sobre cómo navegar o cómo utilizar determinadas funcionalidades. Finalmente, los sistemas de navegación avanzados pueden ser sistemas de personalización, que se adapten según el perfil del usuario que visitante; sistemas de navegación visual que permiten al usuario explorar contenidos mediante recursos icónicos o visuales; y sistemas de navegación social que se basen en el comportamiento o en la evaluación de usuarios anteriores.

En el caso de esta aplicación sobre fauna chilena, se propone el uso de un sistema de navegación de tipo integrado, específicamente un sistema de navegación constante compuesto por un menú de navegación, que posibilite visitar las principales secciones de la aplicación, y por un *footer* que esté siempre presente y permita al usuario regresar a la *home screen* cuando lo estime conveniente. Como ya se mencionó, se pretende usar en la *home screen* recursos icónicos que faciliten al usuario diferenciar los grupos de animales disponibles, además en el listado de especies se utilizará una imagen del animal, por lo que en ambos casos parece correcto hablar de navegación visual. Respecto a los sistemas de navegación complementarios resultaría útil para los usuarios nuevos, la incorporación de una breve guía de uso de la aplicación. Considerando las recomendaciones propuestas por Google para definir la navegación en un entorno móvil⁸⁵, se presenta a continuación un *blueprint* con las tareas esenciales que el usuario podría llevar a cabo al abrir la aplicación sobre animales chilenos. Lo anterior, estableciendo la secuencia principal de acciones del usuario, teniendo en cuenta que su objetivo fundamental debería ser ver la ficha informativa de una determinada especie. Considerando que, tanto desde la *home screen*, que contendrá los grupos de animales (por ejemplo, anfibios, aves, etc.), como desde la pantalla que contendrá el listado completo de especies, sería posible escoger un animal determinado y, por consiguiente, acceder a la ficha informativa de este, el acceso a estas pantallas, que permitirán llevar a cabo las acciones “acceder a un grupo concreto de especies” y “acceder al listado completo de especies”, debe estar disponible en toda la aplicación. En otras palabras, son estas las pantallas que deben formar parte del sistema de navegación constante (menú de navegación y *footer*) de esta. Además, se propone incorporar en el sistema de navegación una acción que permita al usuario acceder a información referente a los objetivos y el desarrollo de la aplicación móvil.

⁸⁵ Google. *Defining your navigation*. <<https://material.io/guidelines/patterns/navigation.html#navigation-defining-your-navigation>>. [Consulta: 01/12/2017].

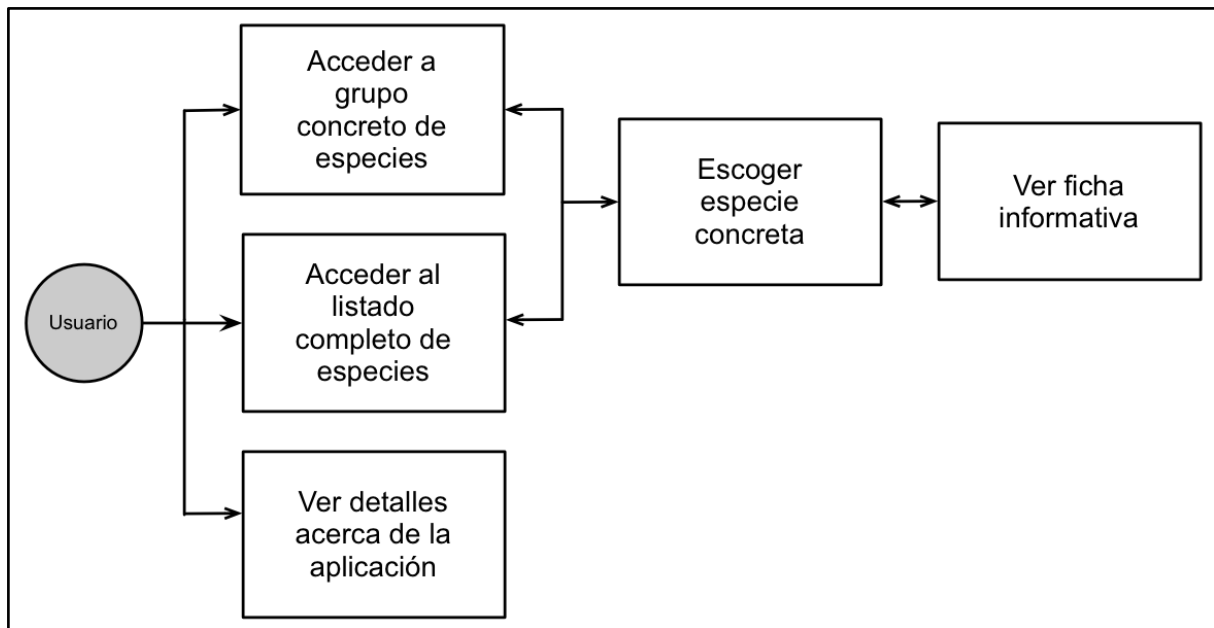


Fig. 12. *Blueprint*: tareas que deberían estar presentes en el sistema de navegación de la aplicación sobre fauna.

Un tercer elemento que constituye la AI de un entorno informacional son los sistemas de búsqueda, estos se definen como componentes de una aplicación móvil, que permiten al usuario localizar, de manera rápida, información a partir de una necesidad específica. Los sistemas de búsqueda ofrecen, al usuario, resultados acorde a la consulta ingresada por este. Con respecto a la recuperación de información en los sistemas de búsqueda, existen dos problemas a superar, estos son el ruido y silencio. El ruido corresponde a los contenidos recuperados que no son pertinentes para la consulta realizada por el usuario, mientras que el silencio ocurre cuando existen documentos pertinentes a la consulta del usuario, pero no son recuperados. Normalmente, los sistemas de búsqueda se basan en la implementación de una caja de búsqueda que permite al usuario llevar a cabo consultas simples sobre los contenidos de un determinado entorno informacional. Es común también, ofrecer la opción de llevar a cabo búsquedas avanzadas mediante recursos de recuperación de información como por ejemplo, operadores booleanos o la aplicación de filtros previos a la búsqueda.

Dos herramientas fundamentales en los sistemas de búsqueda son los desambiguadores semánticos y los *query builders*. Los desambiguadores semánticos permiten al usuario escoger entre una serie de alternativas cuando un término o concepto tiene más de un significado, por ejemplo, al buscar la palabra “hoja” en Wikipedia, es posible observar una página de desambiguación para este término⁸⁶. Los *query builders* son un componente de los sistemas de búsqueda que ayuda al usuario a construir una consulta a medida que escribe en el cuadro de texto de un buscador. En otras palabras, los constructores de consultas autocompletan lo que el usuario escribe con sugerencias de palabras, normalmente las más buscadas o las más populares.

⁸⁶ Wikipedia. *Hoja (desambiguación)*. <[https://es.wikipedia.org/wiki/Hoja_\(desambiguacion\)](https://es.wikipedia.org/wiki/Hoja_(desambiguacion))>. [Consulta: 01/12/2017].

En el caso de la aplicación móvil sobre fauna chilena, se implementará, tanto en el listado total de especies como al acceder a cada grupo de animales, una caja de búsqueda que permita al usuario realizar consultas concretas, sin la necesidad de navegar, o por el listado total de especies o por los diferentes grupos, en busca de un animal determinado, con el objetivo de, precisamente, facilitar la localización de especies. Respecto a la implementación de herramientas para el sistema de búsqueda de la aplicación móvil, por una parte, la incorporación de *query builders* parece ser una opción apropiada en este caso, ya que permitiría agilizar la búsqueda de animales dentro de la aplicación, sobre todo pensando en implementaciones futuras que contengan un gran número de especies. Por otra parte, la incorporación desambiguadores semánticos no parece ser imprescindible en este caso, ya que, debido a la existencia de las categorías taxonómicas de los seres vivos, es poco probable que existan dos especies de animales con el mismo nombre. Cabe señalar que el diseño de algunos de los componentes de AI para la aplicación sobre fauna chilena se plasmará mediante *wireframes*. Los *wireframes* permiten transmitir ideas en cuanto a contenido y diseño a nivel de página para sitios web y aplicaciones móviles que tienen pocas páginas/pantallas (Laubheimer, 2016). A continuación, se presenta un *wireframe* de la implementación de un sistema de búsqueda que incorpora *query builders*, basado en el buscador con autocompletado propuesto por Google para aplicaciones móviles⁸⁷.

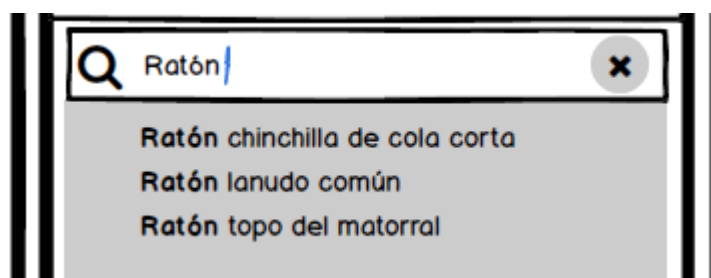


Fig. 13. *Wireframe*: buscador propuesto para la aplicación sobre fauna.

Un cuarto componente de la AI son los sistemas de etiquetado, los que permiten representar los contenidos de una aplicación móvil, es decir, mediante el uso de etiquetas es posible representar fragmentos de contenido con el fin de que el usuario comprenda la forma en que está organizada una aplicación y pueda de esta manera navegar sin inconvenientes por la misma. Por ejemplo, habitualmente se utiliza en sitios web y aplicaciones móviles la etiqueta “acerca de nosotros”, la cual, normalmente, consiste en una breve reseña del propio proyecto y sobre la o las entidades que respaldan la iniciativa. Dentro de los inconvenientes más comunes al momento de llevar a cabo el etiquetado de un entorno digital, se encuentran, la ambigüedad propia del lenguaje natural, esto es una situación en la que la información podría entenderse o interpretarse de más de una manera; la arbitrariedad que se produce al utilizar términos con un significado diferente al que normalmente se

⁸⁷ Google. *In-app search*. <<https://material.io/guidelines/patterns/search.html#search-in-app-search>>. [Consulta: 01/12/2017].

asocian; la desorientación producida al utilizar etiquetas que a simple vista no permiten deducir lo que hay tras ellas; y finalmente, la mala utilización de etiquetas relacionadas con la imagen o la marca de una determinada institución. El etiquetado puede ser de tipo textual o de tipo icónico. Por un lado, el etiquetado textual es usado habitualmente para definir, las etiquetas del sistema de navegación, que deben ser claras, sin la utilización de literales⁸⁸ extensos y autoexcluyentes (que no lleven a confusión); las etiquetas de los enlaces, que deben ser fácilmente reconocibles, con una apariencia normalizada que solo se aplique a ellas; y las etiquetas de las cabeceras o títulos, que deben ser significativas y coherentes con el contenido y su diseño debe ser jerárquico. Por otro lado, el etiquetado icónico consiste en trasladar figuras y aspectos del mundo real al mundo digital, por ejemplo, normalmente el ícono de ubicación dentro de un mapa digital corresponde a un pin, haciendo alusión a la utilización de estos elementos para marcar ubicaciones en mapas físicos. La utilización de iconografía, además de ser un aporte para el aspecto estético de las aplicaciones, constituye una forma de lenguaje universal que permite hacer comprensible un concepto con solo ver una figura. Ambas opciones de etiquetado no son excluyentes y es recomendable usarlas en conjunto para facilitar la comprensión de un determinado concepto, por parte de los usuarios. Finalmente, cabe señalar que el etiquetado para el contenido de una aplicación móvil debe escogerse de acuerdo a las características del público objetivo del proyecto.

Con respecto al etiquetado de la aplicación móvil sobre fauna chilena, tanto para los grupos de la *home screen* como para las pantallas que contengan listados de especies, se utilizará un etiquetado de tipo textual basado en las categorías taxonómicas de los seres vivos, lo que además se acompañará con elementos visuales, íconos en los grupos y fotografías o ilustraciones en los listados, todo lo anterior, con el fin de evitar la ambigüedad propia del lenguaje natural y la arbitrariedad. En relación con los títulos y subtítulos de la ficha informativa de cada especie, se empleará un etiquetado textual jerárquico, cuyo elemento padre corresponderá al nombre del animal y en un nivel jerárquico más bajo, se encontrarán otras secciones con datos inherentes a la especie, como el hábitat o el estado de conservación. Respecto al etiquetado del sistema de navegación, se utilizarán literales breves y concisos que hagan referencia a las diferentes pantallas de la aplicación, como por ejemplo, grupos, especies, acerca de, etc. A continuación, se presenta una propuesta del etiquetado para la ficha informativa de cada especie.

⁸⁸ En AI, las etiquetas de menús, enlaces, cabeceras, botones, etc. reciben el nombre técnico de literales.

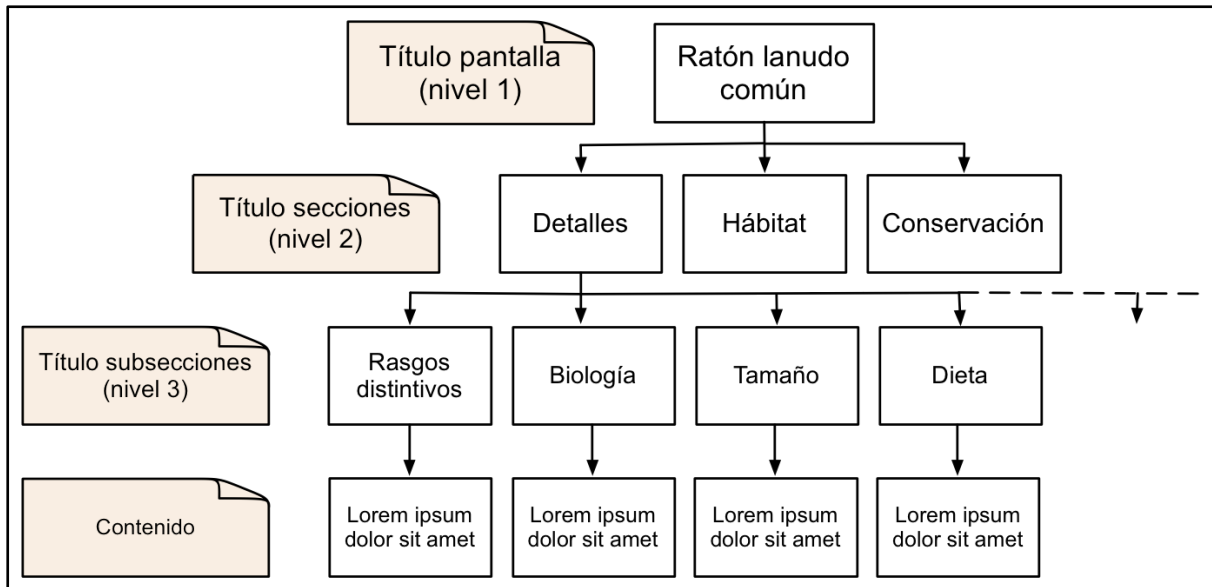


Fig. 14. *Blueprint*: etiquetado jerárquico de la ficha informativa de cada especie.

Un último componente elemental de la AI son los vocabularios o lenguajes documentales, los que permiten articular la navegación y la búsqueda dentro de un entorno informacional. Para generar un lenguaje documental, es necesario llevar a cabo un proceso de indización, el cual consiste en asignar a cada contenido único del entorno informacional, uno o más términos clave (en inglés *keyword*) que representen la información sobre la que trata el documento o recurso. Normalmente, la información resultante de la indización se incorpora al contenido mediante metadatos⁸⁹ asociados, con el fin de utilizarla en los sistemas de organización, navegación, búsqueda y etiquetado. Indizar el contenido de un entorno informacional, permite la reducción del ruido y del silencio en las búsquedas realizadas por los usuarios. También posibilita establecer relaciones entre los términos de indización, las que pueden ser, de equivalencia para términos con el mismo significado; de jerarquía, como por ejemplo, los términos árbol, rama y hoja (todo-parte); y asociativa en el caso de afinidades semánticas como ocurre con los términos dentista y caries. Los lenguajes documentales, pueden ser libres o controlados. Los lenguajes libres se componen de términos no definidos previamente que se generan a partir de la realización de procesos de indización. Mientras que los lenguajes controlados son aquellos que poseen un listado predefinido de términos (vocabulario), que establece relaciones unívocas y precisas entre ellos y entre los conceptos representados.

Con respecto a la implementación de un lenguaje documental controlado en la aplicación móvil sobre fauna chilena, su implementación parece lógica, ya que permitiría utilizar la información indizada en los diferentes sistemas de AI. En este caso, las categorías taxonómicas de los seres vivos, también parecen ser la opción más evidente, sobre todo si se considera que corresponden a un índice de términos ya existente, elaborado por especialistas y ampliamente difundido incluso en

⁸⁹ En campos como la recuperación de información o la web semántica, los metadatos corresponden conjuntos de datos que proporciona la información mínima necesaria para identificar un recurso. En palabras simples, los metadatos son información acerca de otra información.

áreas no académicas. En el caso particular del sistema de búsqueda de la aplicación, la implementación de un lenguaje controlado permitiría establecer, por ejemplo, relaciones de equivalencia entre el nombre común y el nombre científico de una determinada especie, logrando que, tanto al buscar uno como otro, el resultado devuelto sea el mismo. También se podrían generar relaciones asociativas entre un animal y sus rasgos o comportamientos distintivos, logrando que, por ejemplo, al consultar por el término “excavar”, uno de los posibles resultados de búsqueda sea “cucuro”, ya que este es un hábito característico de esta especie de roedor. Esta misma relación asociativa podría aplicarse a otras características de una especie como su alimentación o hábitat. Finalmente, se presenta a continuación, un mapa que muestra la propuesta de AI para la aplicación móvil a desarrollar, que contiene todos los componentes anteriormente descritos.

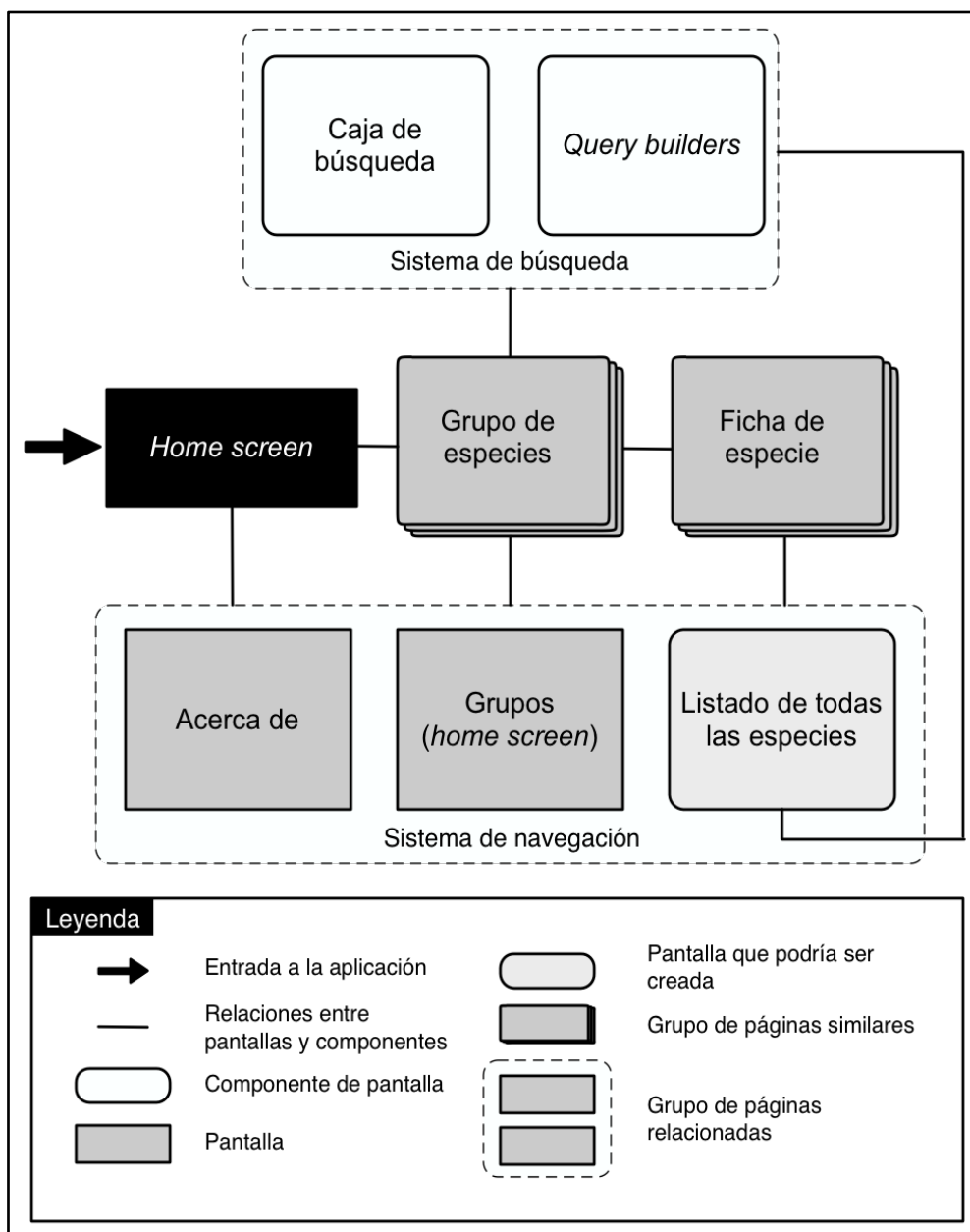


Fig. 15. *Blueprint*: mapa arquitectónico de la aplicación Fauna chilena.

6.1.2. Usabilidad

Además de la definición de la AI, la creación de la GUI de la aplicación móvil a desarrollar corresponde a otra etapa fundamental dentro del diseño de la misma. Es en este punto donde el concepto de usabilidad cobra relevancia. La usabilidad es un atributo de calidad que permite evaluar qué tan fáciles de usar son las GUI (Nielsen, 2012). En otras palabras, la usabilidad corresponde a la facilidad de interacción de un usuario con una herramienta o entorno digital, con el fin de alcanzar un objetivo concreto. La usabilidad está definida por cinco componentes esenciales, la facilidad de aprendizaje, o qué tan fácil resulta para los usuarios realizar tareas básicas la primera vez que se encuentran con el diseño; la eficiencia de uso, que dice relación con la rapidez con que los usuarios pueden realizar tareas, una vez que aprendieron el diseño del software; la facilidad de recordar cómo funciona, vinculada con la facilidad de uso de una GUI luego de un periodo de no utilización por parte del usuario; los errores que los usuarios cometen al interactuar con la GUI, el nivel de gravedad de estos y la facilidad con que pueden recuperarse de los errores; y la satisfacción, que se relaciona con que tan agradable resulta la utilización de un GUI (Nielsen, 2012).

Tanto la usabilidad, como la AI son factores que repercuten directamente en la percepción del usuario, respecto de un determinado producto o servicio, lo que se conoce como experiencia de usuario (UX, por su sigla en inglés). La UX se centra en obtener información detallada sobre los usuarios (sus necesidades, lo que valoran, sus limitaciones, etc.), pero además considera los objetivos de negocio y las posibilidades técnicas del sistema dónde se implementará determinada GUI. Sin bien en este caso concreto no existe un conocimiento extenso sobre los usuarios, si están claramente definidos los objetivos la aplicación móvil sobre fauna chilena y las posibilidades técnicas, tanto a nivel de hardware como a nivel de software.

6.1.2.1. Diseño de interacción y diseño de interfaz

La UX no solo está constituida por lo que el usuario puede ver directamente. Para crear un servicio valioso, atractivo, utilizable y coherente, es necesario considerar el diseño en muchas capas diferentes (Rowland, 2015). Dos capas UX fundamentales corresponden al diseño de interacción y al diseño de la interfaz.

Por una parte, el diseño de interacción (con frecuencia abreviado IxD), como su nombre lo indica, se centra en el comportamiento del usuario al interactuar con un determinado dispositivo o sistema. El IxD permite definir secuencias de acciones entre el usuario y el dispositivo, necesarias para lograr objetivos o actividades particulares (Rowland, 2015). En el caso específico de los terminales móviles, la interacción de los usuarios plantea ciertas diferencias respecto a la interacción con otros dispositivos. Esto se debe principalmente a las características físicas de *smartphones* y *tablets*, fundamentalmente a la portabilidad y a su tamaño reducido en comparación con portátiles y equipos de escritorio; y también a la forma de interacción del usuario, que se lleva a cabo

sosteniendo el dispositivo con una o ambas manos y tocando directamente la pantalla de este, sin necesidad de usar periféricos como teclado y ratón.

Respecto a la forma de sostener el teléfono móvil, las personas lo sostienen de muchas maneras, según las características físicas de este, sus propias necesidades y el contexto en el que se encuentran. Con respecto a esto, cabe señalar que el 75% de los usuarios toca la pantalla solo con un dedo; menos del 50% sostienen su teléfono con una sola mano; el 36% de los usuarios “acuna” (sostiene y asegura) su teléfono, usando su otra mano para obtener un mayor alcance y estabilidad; y el 10% de los usuarios sostiene su teléfono con una mano y toca la pantalla con el dedo de la otra mano (Hooper, 2017). A continuación, se presentan las seis formas más comunes en que las personas sostienen y tocan su *smartphone*.

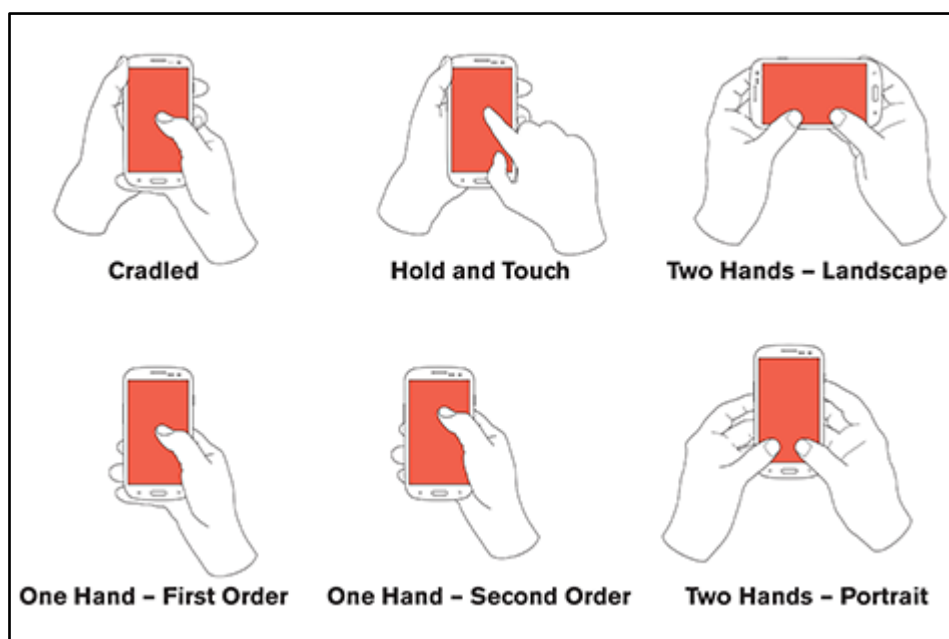


Fig. 16. Formas comunes en que las personas sostienen y tocan su teléfono móvil.⁹⁰

Además de la forma de sostener y de tocar el teléfono móvil, otro factor relevante para llevar a cabo el IxD de una aplicación móvil es el sector de la pantalla que los usuarios prefieren ver el contenido e interactuar con él. En ese sentido, independientemente del tipo de dispositivo, las personas prefieren tocar el centro de la pantalla y lo harán cada vez que se les dé esa opción, además, la gente prefiere ver el contenido en el centro de la pantalla, lo notan más rápidamente y lo leen con mayor precisión. (Hooper, 2017). A partir de lo anterior, resulta evidente que, tanto el contenido principal de una aplicación como las interacciones primarias que el usuario puede llevar a cabo, deberían ubicarse en el centro de la pantalla.

⁹⁰ Hooper, Steven. *Design for fingers, touch, and people, part 1*.
<<https://www.uxmatters.com/mt/archives/2017/03/design-for-fingers-touch-and-people-part-1.php>>.
[Consulta: 01/12/2017].

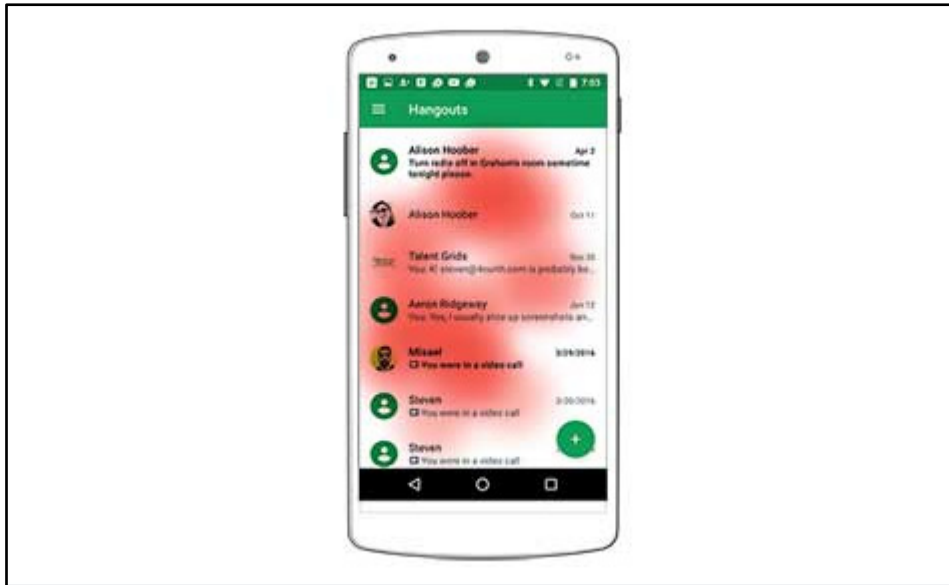


Fig. 17. Mapa de calor que muestra los sectores donde las personas interactúan con más frecuencia.⁹¹

Además de las interacciones táctiles, los dispositivos móviles ofrecen al usuario otras posibilidades de interacción, tanto para ingresar como para obtener datos a partir de las acciones realizadas. En relación con el ingreso de datos, si bien lo más habitual es que se lleve a cabo mediante toques en la pantalla del dispositivo, existen otras opciones de entrada como la realización de gestos sobre la pantalla táctil, la utilización del habla en sistemas de reconocimiento de voz o la detección de la proximidad entre el usuario y el dispositivo a través del sensor de proximidad de este. Respecto a la salida de datos generada a partir de las interacciones del usuario, ésta puede ser, visual, mediante la pantalla del dispositivo móvil, o a través de señales luminosas emitidas por el led del terminal, habitualmente utilizadas para entregar notificaciones; auditiva, cuando se produce un sonido o salida de voz; y táctil, cuando el usuario detecta vibración. En el caso concreto de la aplicación móvil sobre fauna chilena, el ingreso de datos se llevará a cabo principalmente mediante toques y gestos, mientras que la retroalimentación a las interacciones del usuario será únicamente de tipo visual.

Por otra parte, el diseño de interfaz se vincula con la disposición de los elementos visuales en la pantalla y con la estética de la GUI. En términos simples, el diseño de interfaz corresponde a lo que los usuarios ven al abrir una aplicación móvil. En el caso específico de Android, Google propone utilizar la normativa de diseño gráfico Material design⁹², la cual corresponde a una filosofía de diseño basada en las propiedades de los objetos del mundo real (la materia como metáfora) y las interacciones que se pueden realizar con estos. Material design propone una serie de pautas enfocadas al diseño gráfico utilizado para Android y otros productos y servicios de Google. Material Design fue presentado durante la conferencia Google I/O 2014, integrándose ese mismo año en la

⁹¹ Hooper, Steven. *Design for fingers, touch, and people, part 2*. <<https://www.uxmatters.com/mt/archives/2017/05/design-for-fingers-touch-and-people-part-2.php>>. [Consulta: 01/12/2017].

⁹² Google. *Material design*. <<https://material.io/>>. [Consulta: 01/12/2017].

versión 5.0 del SO, Android Lollipop. Con el objetivo de generar una GUI consistente con el diseño Android, es necesario considerar las pautas ofrecidas por Google, ya que de esta manera se aporta a la facilidad de aprendizaje del diseño a elaborar, permitiendo a los usuarios que nunca hayan visto la GUI, realizar operaciones básicas e instintivas sobre ella. En otras palabras, un diseño basado en las pautas de Google es lo que los usuarios esperarían encontrar en una GUI Android. En este sentido, es necesario mencionar que, si bien existen pautas comunes al diseño de cualquier interfaz para dispositivos móviles, cada SO móvil ofrece recomendaciones propias para la creación de GUI. Por ejemplo, en el caso de iOS, Apple entrega las *Human interface guidelines*⁹³.

Finalmente, sería correcto afirmar que el IxD se centra en cómo fluye la información entre UX y UI, tal y como se observa en el siguiente diagrama.

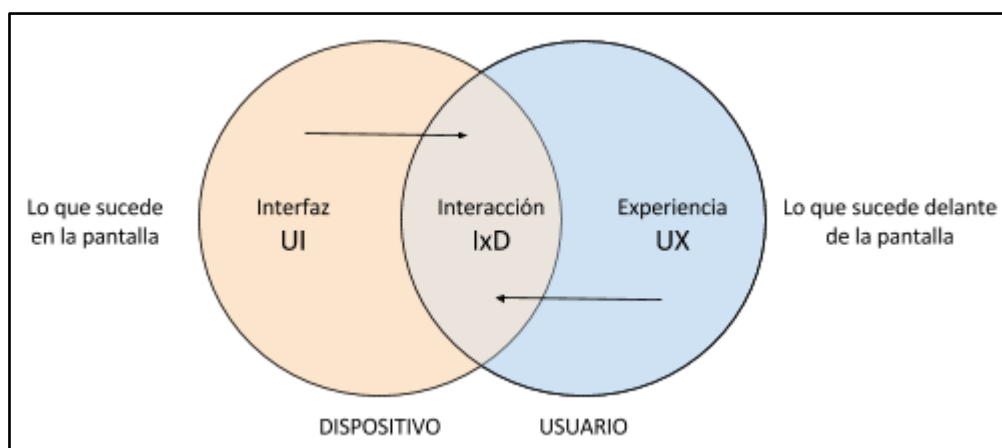


Fig. 18.. Nexo existente entre UI, IxD y UX.⁹⁴

6.1.3. Patrones de diseño

Todos los SO móviles poseen una identidad gráfica y funcional propia, planteando formas específicas de diseñar los elementos de una GUI y de interactuar con el dispositivo móvil. Esta identidad se basa en la utilización de patrones o prácticas recomendadas que contribuyen a la uniformidad de un SO. El concepto de patrón de diseño proviene originalmente del ámbito de la construcción urbanística y fue acuñado por primera vez, en el año 1979, por el arquitecto Christopher Alexander, quien lo definió como “una solución a un problema, que se usa repetidamente en contextos similares con algunas variantes en la implementación”. A principios de la década de los 90, diversos autores vincularon el concepto al contexto del diseño de software orientado a objetos y posteriormente al diseño de GUI. Los patrones son simplemente buenas prácticas bien definidas y bien investigadas, pero que siempre se deben basar en principios fundamentales del diseño, tener en cuenta al usuario

⁹³ Apple. *Human interface guidelines iOS*. <<https://developer.apple.com/ios/human-interface-guidelines/overview/themes/>>. [Consulta: 01/12/2017].

⁹⁴ Kambrica. *UI, UX, IxD: ¿Cuál es la diferencia?*. <<http://www.kambrica.com/blog/ui-ux-ixd-cual-es-la-diferencia/>>. [Consulta: 01/12/2017].

y considerar el propósito del diseño (Hoover; Berkman, 2011). A partir de lo anterior, sería correcto afirmar que los patrones, corresponden a recomendaciones que permiten resolver problemas comunes al diseño de cualquier interfaz, debido a que, sin importar la especificidad de una aplicación móvil, tanto a nivel visual como a nivel funcional, existirán inconvenientes de diseño recurrentes y transversales. Desde la perspectiva de los usuarios, la utilización de patrones permite la identificación de elementos gráficos e interacciones que resultan familiares dentro de un determinado SO, y que aportan a entregar una UX a la que cada grupo objetivo está habituado. A continuación, se ofrece una alternativa de diseño GUI para cada una de las pantallas de la aplicación móvil a desarrollar, a partir de los patrones propuestos por Hoover y Berkman (2011) y también siguiendo las guías de Material Design de Google. Cabe señalar que el diseño de cada pantalla de la aplicación se plasmará mediante *wireframes*.

En el contexto de este desarrollo específico se utilizará una serie de patrones comunes que influirán en el diseño de cada una de las pantallas de la aplicación. Un tipo de patrón transversal a todas las pantallas de la aplicación son los patrones de composición. En este sentido, la composición se entiende como un proceso de ensamblaje que permite organizar los componentes y el contenido de una interfaz de manera coherente (Hoover; Berkman, 2011) mediante el uso de contenedores (en inglés *wrapper*). Un primer patrón de composición presente en el diseño de las pantallas de aplicación móvil son los títulos. Tanto las pantallas como sus secciones de contenido deben usar títulos. Estos títulos deben ser horizontales, consistentes en cuanto a estilo y tienen que seguir pautas que los hagan legibles. En este caso concreto, el título de cada pantalla se incluirá en la barra de título de la aplicación, ubicada en la parte superior de la misma. Esto permitirá destacar el título de la pantalla para dejar en claro que se trata de un elemento clave y además aportará consistencia ya que en cada pantalla la barra de título tendrá el mismo formato. Respecto a los títulos de las secciones de contenido dentro de las diferentes pantallas, se aplicará en ellos un determinado estilo gráfico (fuente, tamaño de letra, color, etc.) que permita diferenciarlos del contenido restante y comunicar claramente su jerarquía, probablemente siguiendo el estilo de tipografía del Material Design propuestas por Google⁹⁵. Otro aspecto importante en cuanto a la presentación de los títulos es el nivel de profundidad de los mismos, en este caso se evitará el uso de más de cuatro niveles de profundidad, con el fin de evitar confusiones y lograr que el usuario diferencie entre los niveles de título. En relación con la interacción, los títulos de la barra de título funcionarán como un botón “arriba” (en inglés *up button*), siguiendo los patrones de navegación propuestos por Google⁹⁶, que permitirá al usuario regresar a la pantalla anterior cuando lo requiera, mientras que los títulos de las secciones de contenido serán sólo etiquetas de texto y no tendrán ninguna funcionalidad específica.

Un segundo patrón de composición que se empleará en la mayoría de las pantallas de la aplicación móvil a desarrollar es el desplazamiento (en inglés *scroll*). La implementación del *scroll* es necesaria

⁹⁵ Google. *Typography*. <<https://material.io/guidelines/style/typography.html>>. [Consulta: 08/12/2017].

⁹⁶ Google. *Up and back buttons*. <<https://material.io/guidelines/patterns/navigation.html#navigation-up-back-buttons>>. [Consulta: 08/12/2017].

cuando la información de una pantalla excede el área de visualización de contenido (en inglés *viewport*), por lo que se hace necesario mostrarla al usuario de algún modo. Típicamente se emplea una barra de desplazamiento (en inglés *scrollbar*) que permite al usuario moverse vertical u horizontalmente, a través del contenido de una pantalla, según lo requiera. En el caso específico de los dispositivos móviles, la *scrollbar* no debe ser manipulada directamente por el usuario, sino que, mediante algún gesto en pantalla, normalmente arrastrando el dedo sobre esta (en inglés *drag*). De esta forma la *scrollbar* sirve solo como guía para que este conozca su posición relativa dentro del área de desplazamiento. Debido a esto y a diferencia del desplazamiento en entornos de escritorio, en entornos móviles la *scrollbar* no incluye flechas con las que el usuario pueda interactuar. La *scrollbar* debe reflejar la relación entre el *viewport* y el área total de desplazamiento. En este caso específico, el *scroll* en las pantallas de la aplicación solo se implementará en el eje vertical, ya que el desplazamiento vertical suele ser el más fácil de entender y usar para los usuarios, quienes, además están más familiarizados con los elementos de desplazamiento vertical, por lo que reaccionan mejor ante ellos (Hooper; Berkman, 2011). Respecto de la interacción, el *scroll* por inercia se ha convertido en una funcionalidad esperada por los usuarios, por lo que también se empleará en las pantallas de la aplicación que permitan desplazamiento. El *scroll* por inercia consiste en que, si una vez comenzado el *drag* por parte del usuario, este quita rápidamente el dedo de la pantalla (en inglés *flick*), el *scroll* continuará por un par de segundos o se detendrá si el usuario hace *tap* sobre la pantalla. En relación con la presentación de la *scrollbar*, en este caso específico, permanecerá oculta hasta que el usuario inicie un *drag* vertical, sirviendo como indicador de posición dentro del área de desplazamiento.

Un tercer patrón de composición presente en todas las pantallas de la aplicación móvil sobre fauna chilena es el menú de navegación desplegable (navegación oculta). Considerando que el espacio es un factor valioso en las pantallas de dispositivos móviles, y que la estructura de un menú generalmente está dictada por el SO en que se implementará, la utilización de un menú de navegación fijo (navegación visible) parece no ser adecuada, debido a que este ocuparía espacio que podría ser utilizado para mostrar contenido. Es por lo anterior que una alternativa más apropiada para mostrar al usuario las opciones de navegación disponibles, es utilizar un pequeño elemento en pantalla (un botón, por ejemplo) que permita, mediante alguna interacción, revelar estas opciones. Actualmente este tipo de navegación es prácticamente un estándar en las GUI de aplicaciones Android, sobre todo debido a que el propio Google emplea menús de navegación laterales (denominados *navigation drawers*)⁹⁷ de tipo “hamburguesa” (en inglés *hamburger menu*) en sus aplicaciones móviles Android, como por ejemplo en Play Store, Google Maps y Gmail. En este caso específico y siguiendo la tendencia de Google, se opta por el uso de un *hamburger menu* (ícono compuesto por tres líneas horizontales) ubicado en la esquina superior izquierda de la pantalla. Este elemento, desplegará las opciones de navegación cuando el usuario haga un toque sobre él (en

⁹⁷ Google. *Navigation drawer*. <<https://material.io/guidelines/patterns/navigation-drawer.html>>. [Consulta: 08/12/2017].

inglés *tap*) o deslice, horizontalmente el dedo sobre la pantalla, de izquierda a derecha (en inglés *right drag*) y las ocultará si el usuario hace *tap* nuevamente en el menú o realiza un *left drag*. La principal ventaja de un menú de este tipo es que puede contener una cantidad bastante grande de opciones de navegación en un espacio pequeño y también puede admitir fácilmente submenús, si es necesario (Budiu, 2015). Si bien en este caso, las opciones del menú no contendrán submenús y estarán vinculadas directamente con otras pantallas de la aplicación o hacia enlaces externos, las posibilidades ofrecidas por un menú de navegación desplegable facilitan la incorporación futura de más elementos de navegación, en caso de ser necesario. Finalmente, cabe señalar que existen determinadas pruebas de usabilidad, realizadas tanto en dispositivos móviles como GUI de escritorio, que demuestran que la navegación oculta afecta la UX, ya que resulta menos detectable para el usuario que la navegación visible o parcialmente visible. Cuando la navegación está oculta, es menos probable que los usuarios la utilicen y si lo hacen, tardan más en llevar a cabo esta tarea que si estuviera visible (Pernice; Budiu, 2016). Es por esta razón que, con el objetivo de dar acceso a los contenidos clave de la aplicación, se repetirán en el *footer* las opciones de navegación del menú desplegable. Repetir la navegación principal del sitio en el pie de página permite a los usuarios que llegan al final de la página tener acceso rápido a la navegación, pero también expone la navegación del sitio a aquellos que no exploran el menú de hamburguesas (Schade, 2015).



Fig. 19. De izquierda a derecha, *wireframe* del menú de navegación desplegable y del *footer*.

Otro tipo de patrón que estará presente en las pantallas de la aplicación móvil a desarrollar corresponde a los patrones para el detalle o desglose (en inglés *drilldown*). Estos patrones se basan en proveer al usuario acceso a información adicional relacionada con un determinado contenido. Un primer patrón de desglose que probablemente se encuentre presente en diferentes secciones de la aplicación móvil a desarrollar son los enlaces. Los enlaces de texto permiten acceder a contenido relacionado, desde ubicaciones arbitrarias dentro de una pantalla, generalmente, cargando una nueva pantalla de contenido o saltando a una sección diferente de la pantalla actual. Respecto a su

presentación, se diferencian del resto del contenido textual ya que poseen un estilo diferente, habitualmente un cambio de color y subrayado (para texto) o un borde de color (para imágenes). En relación con la interacción del usuario, los enlaces se “abren” al hacer *tap* sobre ellos. Un segundo patrón de desglose que condicionará el diseño GUI de la aplicación a desarrollar son los botones. Los botones se utilizan para iniciar acciones, según las cuales una pantalla y/o su contenido pueden cambiar, o para activar determinadas funcionalidades como búsquedas o envío de formularios. Respecto a la interacción, en dispositivos móviles los botones se activan mediante un *tap*. En relación con la presentación de los botones, estos deben resultar fáciles de visualizar para el usuario, por lo que su estilo gráfico debe permitir destacarlos en relación con el fondo de la pantalla. Un tercer patrón de desglose necesario para esta aplicación móvil son los íconos. Los íconos corresponden a representaciones gráficas de acciones o destinos y deben ser fáciles de comprender y de recordar. Habitualmente, son utilizados como enlaces hacia otras páginas o secciones de una aplicación. Su utilización es común en la *home screen* y en otras pantallas que empleen un diseño en *grid*. Con respecto a su presentación, los íconos de un proyecto deben ser similares entre sí, tanto en diseño como en tamaño y generalmente deben acompañarse de una etiqueta de texto centrada debajo del ícono.

Otro tipo de patrón presente en la aplicación móvil a desarrollar son los patrones generales de interacción, específicamente los gestos en pantalla. En entornos móviles, los gestos que el usuario puede realizar en la pantalla del dispositivo representan una forma de interacción directa con los elementos de esta. Lo anterior, partiendo de la premisa de que, elementos en pantalla corresponden a objetos físicos que pueden manipularse "directamente" de manera realista (Hobber; Berkman, 2011). Si bien existen una enorme variedad de gestos táctiles posibles de implementar y utilizar en la plataforma Android⁹⁸, en este desarrollo concreto solo se utilizarán solo cuatro de los gestos centrales propuestos por Wroblewski (2011), específicamente, los ya mencionados, *tap*, *drag*, *flick* y de ser necesario *press*, que consiste en tocar la pantalla por un período prolongado de tiempo.

Además de los patrones transversales a toda la aplicación, cada pantalla posee patrones únicos dictados por sus elementos visuales y funcionalidades. Con respecto a la *home screen*, en el diseño de esta pantalla influirá el patrón de composición pantalla de inicio. La pantalla de inicio de un sistema, en este caso una aplicación móvil, debe ofrecer un conjunto predeterminado de información y acciones una vez que este haya iniciado. Habitualmente, la pantalla de inicio posee una serie de íconos con todas las opciones disponibles en la aplicación, generalmente mostradas en *grid*. Un segundo tipo de patrón presente en la *home screen* es el patrón para mostrar información, *grid*. Los patrones para mostrar información basan su funcionamiento en componentes cuya única tarea es presentar conjuntos de información ordenados para que los usuarios puedan comprenderlos y actuar en consecuencia. Respecto al patrón *grid*, este permite mostrar una matriz de elementos seleccionables, similares en aspecto y tamaño. En este caso específico de la *home*

⁹⁸ Google. *Gesture*. <<https://material.io/guidelines/patterns/gestures.html>>. [Consulta: 10/12/2017].

screen de la aplicación móvil sobre fauna chilena, se utilizará un *grid* con los grupos de animales disponibles. Cada elemento del *grid* corresponderá a un botón con un ícono o imagen que represente a un determinado grupo de animales, además se acompañará esta representación icónica con una etiqueta textual que indique el nombre del grupo.



Fig. 20. Wireframe de la home screen.

Al acceder a un determinado grupo de animales, se desplegará una pantalla con un listado de todas las especies disponibles en ese grupo. Siguiendo esta misma lógica también es posible, en caso de ser necesario, implementar una pantalla que contenga un listado con todas las especies de la aplicación. En relación con el diseño este tipo de pantallas, dos patrones para mostrar información que influirán directamente en su construcción, son la lista vertical y la lista de miniaturas. La utilización de listas verticales es un método clave de organización y presentación de información en dispositivos móviles. Por una parte, la lista vertical permite mostrar información, basada en texto, de la manera más simple y eficiente posible, considerando que el texto es “horizontalmente ineficiente” ya que ocupa la mayor parte de su espacio en el eje horizontal (Hobber; Berkman, 2011). Respecto a la interacción con las listas, normalmente la selección de uno de sus elementos implica ver los detalles de este, ya sea en la propia lista o en una nueva pantalla. Si los elementos de la lista exceden el *viewport*, es necesaria la implementación de un patrón *scroll* que permita visualizarlos y posibilite al usuario conocer su posición dentro de la lista. En este sentido, es necesario considerar que, al llegar al final de la lista, el desplazamiento de la misma debe detenerse. En relación con la presentación de la lista, se recomienda que las etiquetas de texto estén alineadas a la izquierda para

facilitar el escaneo visual de los elementos por parte del usuario. En caso de presentar información adicional mediante una segunda etiqueta de texto, esta debe ir debajo de la etiqueta principal y ser más pequeña y/o tener un contraste menor, con el fin de destacar la etiqueta de texto principal. Finalmente es necesario que las divisiones entre los elementos de la lista sean evidentes, por ejemplo, utilizando líneas divisorias o agregando espacio suficiente entre un elemento y otro. Por otra parte, la lista de miniaturas se basa en la lista vertical pero además incorpora algún tipo de información gráfica adicional (miniatura) para ayudar al usuario a comprender los elementos dentro del conjunto de datos. Normalmente, dentro de un elemento de la lista, la miniatura es el componente ubicado más a la izquierda. En el caso de la aplicación móvil sobre fauna chilena, se modificará la miniatura de los elementos de la lista, de tal manera que se pueda incorporar como fondo, una fotografía de la especie, con el objetivo de facilitar al usuario la identificación visual del animal. Con este mismo fin, se agregará también, en cada elemento de la lista, una etiqueta de texto principal con el nombre común de la especie y debajo de ésta, una etiqueta secundaria con el nombre científico del animal. Finalmente, cabe señalar que la lista de especies se ordenará alfabéticamente según el nombre común de cada animal.

Un patrón que comúnmente se utiliza en conjunto con las listas es el patrón de control de información, “buscar dentro”. Los patrones de control de información permiten al usuario utilizar funciones que posibilitan un acceso rápido a la información necesitada. Lo anterior, considerando que el tamaño del *viewport*, en los dispositivos móviles, limita la cantidad de información que se puede presentar al usuario en un momento determinado. El patrón “buscar adentro”, permite, mediante una caja de búsqueda, consultar y encontrar rápidamente información concreta que el usuario sabe que existe dentro de la aplicación. Respecto a la interacción con la caja de búsqueda, el usuario podrá ingresar caracteres en un cuadro de texto, mediante el teclado virtual o usando el micrófono del dispositivo móvil, y luego haciendo *tap* en el botón de búsqueda obtener resultados acorde a la búsqueda realizada. Respecto a la presentación de la caja de búsqueda, normalmente se ubica en la parte superior de la pantalla, debajo de la barra de títulos, o en la propia barra, reemplazando al título cuando la funcionalidad de búsqueda se inicia, como se observa en las GUI de Google⁹⁹. Cabe señalar que Google diferencia entre dos tipos de búsqueda dentro de una aplicación, *persistent search* y *expandable search*. *Persistent search* debe implementarse cuando el enfoque principal de la aplicación o de una pantalla de esta sea precisamente la búsqueda, mientras que *expandable search* debe utilizarse en el caso contrario, y normalmente se encuentra oculta en la barra de títulos hasta que el usuario lo requiera.

Un tercer patrón presente en las pantallas del listado de especies, y estrechamente relacionado con la funcionalidad de búsqueda, es el patrón para revelar más información, resultados devueltos. En este contexto, revelar más información significa que la información no debe ocultarse detrás de un enlace u otra acción, sino que debe estar disponible de inmediato (Hoover; Berkman, 2011).

⁹⁹ Google. Search. <<https://material.io/guidelines/patterns/search.html>>. [Consulta: 10/12/2017].

Respecto del patrón resultados devueltos, como su nombre lo indica, permite mostrar al usuario un conjunto de datos, a partir de una solicitud explícita de este. En este sentido, con el objetivo de agilizar, tanto el proceso de búsqueda como la obtención de resultados, la funcionalidad de búsqueda de especies de la aplicación móvil sobre fauna chilena, acotará automáticamente los resultados de la lista de especies a medida que el usuario ingrese su consulta en la caja de búsqueda. Lo anterior considerando además que el tipo más común de resultados devueltos es el de lista vertical (Hooper; Berkman, 2011). Además, tomando en consideración los patrones de búsqueda de Google, se incorporará en la caja de búsqueda un botón “x” que permita al usuario borrar la consulta ingresada.

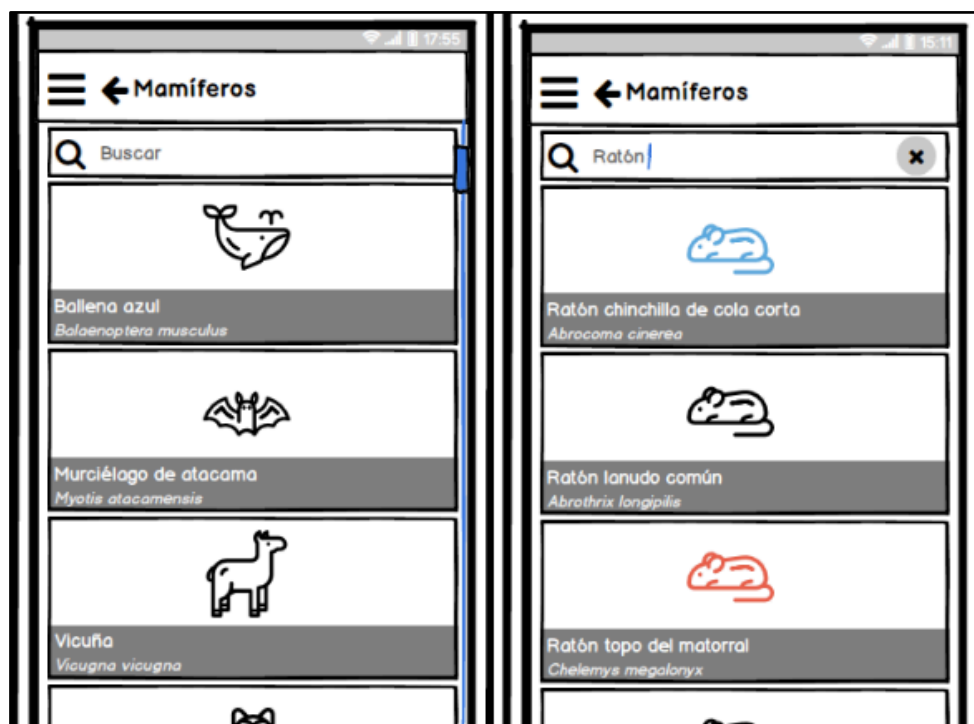


Fig. 21. De izquierda a derecha, *wireframes* del listado de especies y de la funcionalidad de búsqueda.

Una vez que el usuario escoja un elemento de la lista, se abrirá una pantalla con información acerca de la especie elegida. Esta ficha informativa sobre el animal se organizará utilizando el patrón de acceso lateral *tabs*. Los patrones de acceso lateral permiten presentar elementos de información, que están en un mismo nivel de AI, de manera lateral. Uno de los principales beneficios del acceso lateral es que limita el número de niveles de información que un usuario debe explorar para acceder a la información de prioridad. En el caso concreto del patrón *tabs*, este se aplica cuando es necesario proporcionar acceso a un número limitado de elementos, normalmente entre tres y ocho, que están en el mismo nivel de AI. Las *tabs* corresponden a una metáfora de las carpetas de archivos del mundo real, por lo que para llevar a cabo su implementación es necesario seguir los mismos principios aplicados a estos elementos. En este sentido, todas las *tabs* de una pantalla, deben ser visibles a la vez; es necesario indicar al usuario cuando una *tab* esta selecciona y/o directamente abrirla (solo una *tab* debe estar abierta a la vez); finalmente, se debe etiquetar correctamente cada

tab, de manera tal que la etiqueta textual exprese claramente su contenido, es posible también utilizar íconos que apoyen el concepto de cada etiqueta. Respecto a la interacción del usuario con las *tabs* en dispositivos móviles, para seleccionar una, el usuario puede hacer *tap* sobre la etiqueta de texto de esta o hacer *left/right flick* en la pantalla para moverse entre ellas. En relación con este último punto, no es correcto combinar *tabs* con contenido que también admita deslizamiento horizontal (Google, 2017). En cuanto a la presentación de las *tabs*, se sugiere ordenarlas en una fila horizontal, ya que resultan más entendibles para el usuario y además es la propuesta de Google para sus interfaces¹⁰⁰. Otra opción valorada para el diseño de la ficha informativa fue el uso de botones de navegación al final de cada pantalla, finalmente se desestimó esta opción, principalmente debido a que, en pantallas en que el contenido excede el tamaño del *viewport*, los botones quedarían ocultos a la vista del usuario, lo que implicaría que este deba desplazarse hasta el final de la pantalla para visualizarlos.

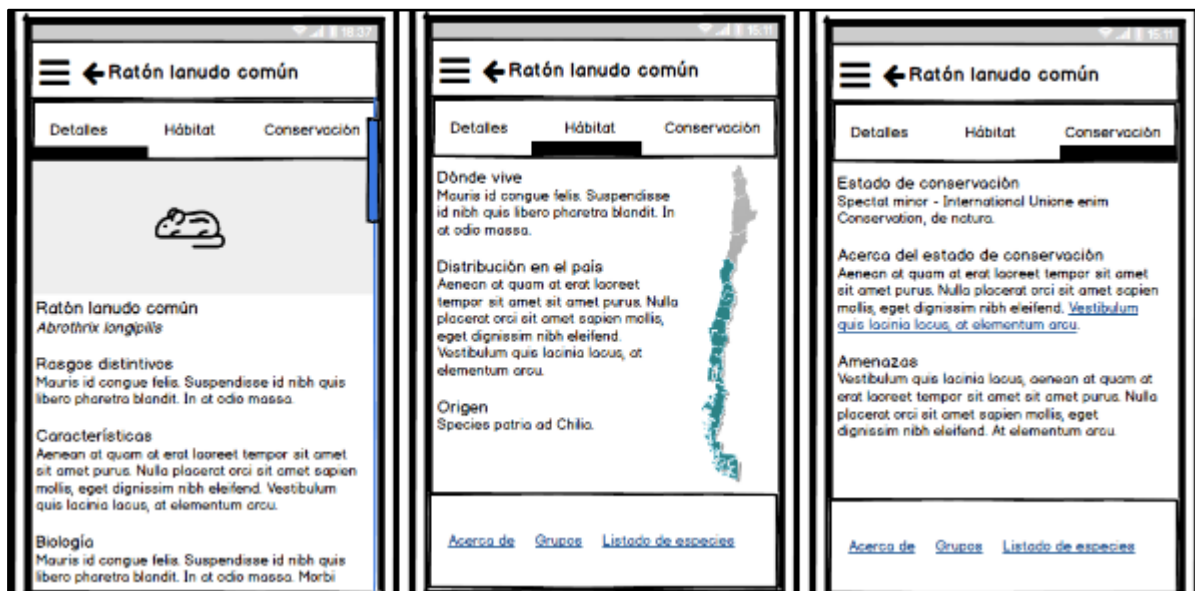


Fig. 22. Wireframe de la ficha informativa de las especies.

Respecto a los diseños GUI presentados, si bien los *wireframes* permiten transmitir ideas en cuanto al diseño de una pantalla no resultan tan útiles para comunicar flujos de procesos dinámicos (Laubheimer, 2016). Es por lo anterior y con la intención de plasmar la interacción del usuario con la aplicación sobre fauna chilena que, se presenta a continuación un diagrama con el principal flujo de trabajo de la aplicación, desde que el usuario ingresa, hasta que logra visualizar la ficha informativa de un determinado animal. Para lograr lo anterior se utilizará un *wireflow*. Los *wireflows* son diagramas que permite documentar las interacciones de los usuarios con una aplicación móvil mostrando secuencias de varios *wireframes*. El uso de diseños de pantalla, en lugar de símbolos de diagrama de flujo abstracto, mantiene el foco en el producto con el que los usuarios interactuarán (Laubheimer, 2016).

¹⁰⁰ Google. *Tabs*. <<https://material.io/guidelines/components/tabs.html>>. [Consulta: 10/12/2017].

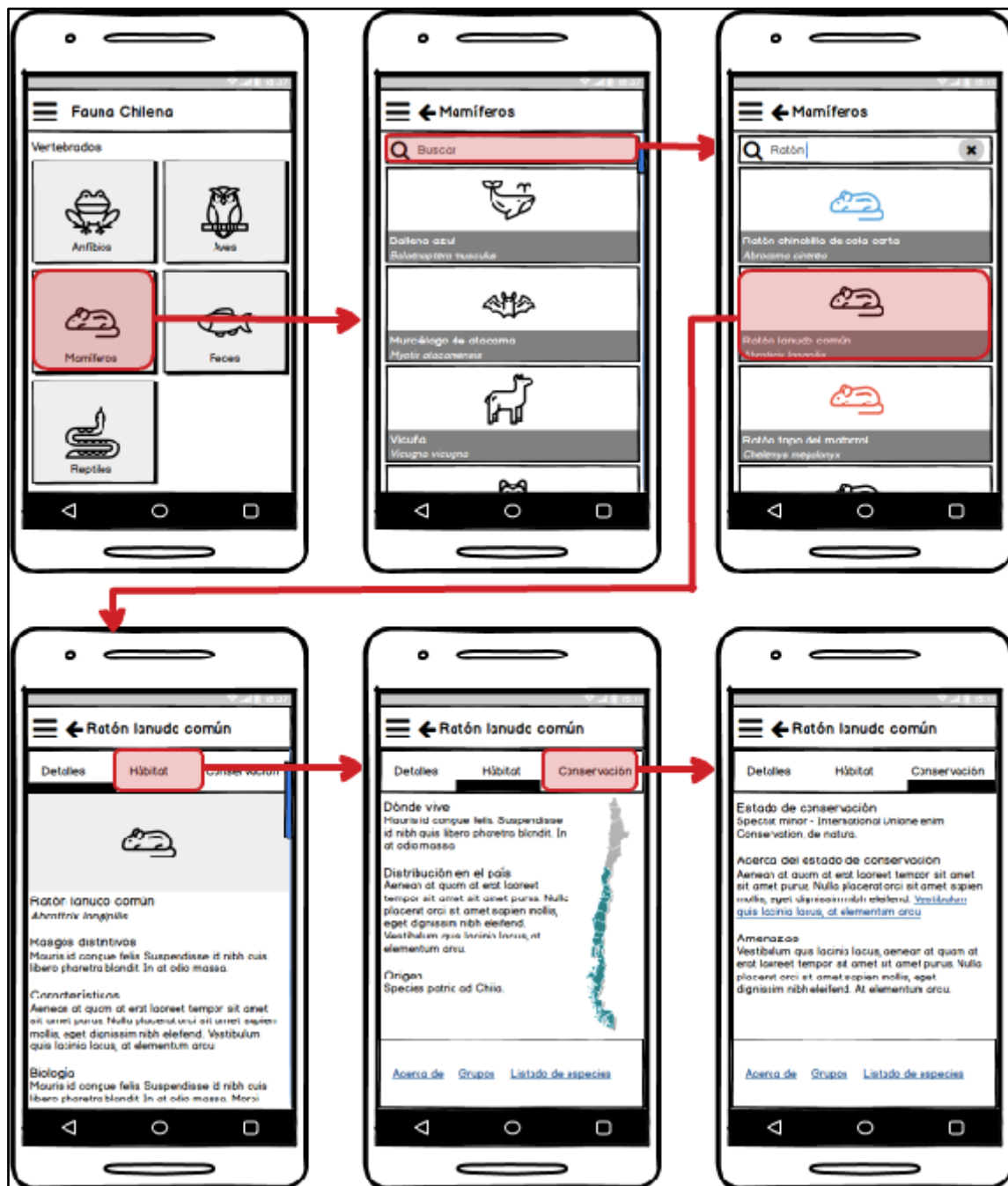


Fig. 23. Wireflow que muestra la principal tarea de la aplicación sobre fauna chilena.

Finalmente, con respecto a esta propuesta de interfaces GUI, parece necesario mencionar que, si bien estas fueron diseñadas en base recomendaciones ampliamente difundidas, y luego de estudiar iniciativas móviles similares, no es posible asegurar el nivel de usabilidad de la aplicación móvil sin antes realizar algún tipo de evaluación por parte de los usuarios. Es por esto que se recomienda, previa implementación, la realización de pruebas de usabilidad, con el fin de confirmar el cumplimiento de las necesidades o expectativas de los usuarios finales.

6.2. Implementación

6.2.1. Elementos transversales al desarrollo

Tal y como se definió en el apartado Decisión de desarrollo, se construirá una aplicación híbrida de prototipo, enfocada en Android, utilizando para ello, el *framework* JQuery Mobile. Como se mencionó con anterioridad, la construcción de aplicaciones híbridas generalmente se lleva a cabo utilizando las tecnologías propias del desarrollo web (HTML5, CSS3 y JS). Con respecto a la creación de cada una de las pantallas de la aplicación móvil, existen elementos y componentes comunes utilizados para su desarrollo. A continuación, se describen estos elementos.

Un primer elemento común al desarrollo de todas las pantallas de la aplicación móvil, son las etiquetas estructurales/semánticas de HTML5¹⁰¹. Estos elementos HTML permiten indicar de manera inequívoca las diferentes secciones de un documento web y al mismo tiempo permiten dar formato estructural a éste. De esta manera, por medio de estas definiciones es posible que los procesadores HTML, como buscadores web y motores de búsqueda, presenten documentos y aplicaciones de manera consistente en diferentes contextos (W3C, 2017). Las etiquetas HTML5 más relevantes utilizadas para la construcción de las pantallas de la aplicación son <header> para definir el encabezado, <nav> para especificar las opciones de navegación de la aplicación, <main> para determinar el contenido principal, <aside> para indicar cuál es el contenido secundario, <footer> para definir el pie de página y <sección> para identificar distintas secciones temáticas dentro del contenido principal y secundario de las pantallas.

Un segundo elemento transversal al desarrollo de cualquier aplicación móvil con JQuery Mobile, son los atributos personalizados de HTML5 (en inglés *custom data attribute*). Los *custom data attributes* están destinados a almacenar datos personalizados privados en una página o aplicación, en el caso de que no haya atributos o elementos más apropiados (W3C, 2017). En otras palabras, los *custom data attributes* permiten almacenar información adicional personalizada sobre los elementos de un documento HTML. En el caso específico de JQuery Mobile¹⁰², los *custom data attributes* de HTML5 se utilizan para permitir inicializar y configurar diferentes *widgets*¹⁰³ basados en elementos HTML (JQuery Mobile, 2017). Mediante los *widgets* JQuery Mobile, es posible maquetar las distintas secciones de una GUI para móviles. Por ejemplo, mediante la utilización de atributos *data-role* de JQuery Mobile, es posible definir propiedades específicas para una sección o un elemento concreto dentro de un documento HTML, mediante la asignación de valores como “*header*”, “*main*” o “*footer*”. Realizar esto implica la incorporación automática de un estilo gráfico determinado para

¹⁰¹ W3C. *Semantics*. <<https://www.w3.org/TR/html51/dom.html#elements-semantics>>. [Consulta: 17/12/2017].

¹⁰² JQuery Mobile. *Data attributes*. <<http://api.jquerymobile.com/data-attribute/>>. [Consulta: 17/12/2017].

¹⁰³ JQuery Mobile. *Category: Widgets*. <<https://api.jquerymobile.com/category/widgets/>>. [Consulta: 17/12/2017].

cada elemento, según el valor asignado. Lo anterior, ocurre debido a que los valores de los atributos *data-role* se encuentran asociados a estilos CSS predefinidos para implementaciones JQuery Mobile.



```
HTML
1 <div data-role="header">
2   <h1>Page Title</h1>
3 </div>
```

Page Title

Fig. 24. Código fuente que muestra un elemento `<div>` con valor “header” para el atributo *data-role*, y resultado visual de este código. Fuente: JQuery Mobile 1.4.5 demos (2017).

Respecto del atributo *data-role*, un valor fundamental que este puede tomar, en documentos creados mediante el *framework* JQuery Mobile, es el valor “page”. La asignación de este valor permite crear páginas dentro del `<body>` de un documento HTML. La página¹⁰⁴ es la unidad de interacción principal en JQuery Mobile y se usa para agrupar contenido en vistas lógicas (JQuery Mobile, 2017). Cabe señalar que, también es posible construir un documento HTML que contenga varias páginas (estructura multipágina), que se diferenciarán entre sí mediante el atributo `id`. El atributo `id` es un identificador único (UID por su sigla en inglés) que no debe repetirse en todo el documento HTML y cuyo propósito es identificar un determinado elemento para posteriormente vincularlo, con scripts u hojas de estilo (Mozilla, 2017). Finalmente, es necesario mencionar que es posible para los usuarios navegar entre las diferentes páginas de un documento JQuery Mobile, mediante AJAX, sin necesidad de volver a recargar el contenido en pantalla.

Con el objetivo de llevar a cabo un desarrollo híbrido, cuyo *look and feel* se asemeje lo más posible al de aplicaciones nativas Android, se utilizará para esta implementación `nativeDroid2`¹⁰⁵. `NativeDroid2` es un tema de JQuery Mobile, cuyo diseño, tanto visual como de interacción, está inspirado en el Material Design de Google. Además de esto, `nativeDroid2` posee una serie de características que facilitan la creación de aplicaciones móviles Android, como por ejemplo, una vasta cantidad de íconos basados en Material Design¹⁰⁶, fragmentos de código reutilizables e incluso una versión demostrativa del tema¹⁰⁷, todo esto, accesible desde su sitio web de documentación¹⁰⁸. Una

¹⁰⁴ JQuery Mobile. *Pages*. <<http://demos.jquerymobile.com/1.4.5/pages/>>. [Consulta: 17/12/2017].

¹⁰⁵ Godesign Webpublishing. *nativeDroid2*. <<http://natedroid.godesign.ch/material/>>. [Consulta: 17/12/2017].

¹⁰⁶ Kupletsky, Sergey. *Material design iconic font*. <<http://zavoloklom.github.io/material-design-iconic-font/icons.html>>. [Consulta: 17/12/2017].

¹⁰⁷ Godesign Webpublishing. *nativeDroid2 demo*. <<http://nd2.godesign.ch/>>. [Consulta: 17/12/2017].

característica destacable de nativeDroid2 es el *widget* nd2-include, el cual permite incluir o incrustar contenido en un documento HTML, desde otro archivo local. Esto agiliza la realización de modificaciones de código fuente, ya que los cambios necesarios se efectúan una sola vez y en un solo lugar (en el archivo incrustado) y no en cada documento HTML. Nd2-include resulta ideal para integrar componentes que estarán presentes en todas las pantallas de una aplicación. En el caso específico de esta implementación, tanto el *hamburger menu* como el *footer* se incluirán en las diferentes pantallas de la aplicación utilizando el *widget* nd2-include.

```
<!-- home screen -->
<div data-role="page" id="homescreen">
  <!-- hamburger menu -->
  <nd2-include data-src="hamburger.menu.html">
    <span class="sr-only">Menú</span>
  </nd2-include>
  <!-- encabezado home screen -->
  <header ...6 lines /><!-- /header -->
  <!-- contenido home screen -->
  <main ...33 lines /><!-- /main -->
  <!-- pie de página -->
  <nd2-include data-src="footer.html"></nd2-include>
</div><!-- /page -->
```

Fig. 25. Código fuente: utilización de nd2-include para incorporar el *hamburger menu* y el *footer* en la *home screen*.

6.2.2. Pantallas de la aplicación

Considerando los patrones de diseño descritos en el apartado de Diseño, se detallan a continuación los elementos JQuery Mobile, utilizados para la implementación de las pantallas principales de la aplicación sobre fauna. En base al patrón de composición, menú de navegación desplegable, se creó un *hamburger menu* con las opciones de navegación de la aplicación. Para la construcción de este menú fue clave el uso de los *custom data attributes* *data-role* y *data-display*. Al primero se le asignó un valor “*panel*” para indicar que se trata de un panel JQuery Mobile¹⁰⁹ y al segundo un valor “*overlay*” para lograr que el *hamburger menu* se superponga al contenido de la pantalla. Con respecto a las opciones de navegación del *hamburger menú*, se creó una lista de enlaces, usando el atributo *data-role* con valor “*listview*”¹¹⁰, en combinación con las etiquetas HTML y , las que permiten crear listas y sus correspondientes elementos. Semánticamente hablando, las opciones de navegación del *hamburger menú* se encuentran definidas mediante la etiqueta <nav> de HTML5, la que permite determinar la sección que contiene enlaces de navegación dentro de un documento HTML. Con respecto a lo anterior, cabe señalar que, no todos los enlaces de un documento deben

¹⁰⁸ Godesign Webpublishing. *nativeDroid2 documentation*. <<http://natedroid.scripter.click/>>. [Consulta: 17/12/2017].

¹⁰⁹ JQuery Mobile. *Panel*. <<http://demos.jquerymobile.com/1.4.5/panel/>>. [Consulta: 17/12/2017].

¹¹⁰ JQuery Mobile. *Listview*. <<http://demos.jquerymobile.com/1.4.5/listview/>>. [Consulta: 17/12/2017].

estar en un elemento <nav>, el cual está destinado solo al bloque principal de enlaces de navegación; por lo general, el elemento <footer> a menudo tiene una lista de enlaces que no necesitan estar en un elemento <nav> (Mozilla, 2017).

```
<!-- hamburger menu -->
<div id="leftpanel" data-role="panel" data-display="overlay"
  data-position-fixed="true" >
  <!-- Perfil estilo Material de nativeDroid2 -->
  <div ...16 lines /><!-- /nd2-sidepanel-profile -->
  <!-- opciones de navegación -->
  <nav id="navoptions" role="navigation">
    <ul data-role="listview" data-inset="false">
      <li>
        <a href="about.html" data-ajax='false'>
          <span class='zmdi zmdi-info-outline'></span> Acerca de
        </a>
      </li>
      <li>
        <a href="index.html" data-ajax='false'>
          <span class='zmdi zmdi-apps'></span> Grupos
        </a>
      </li>
      <li>
        <a href="species.html" data-ajax='false'>
          <span class='zmdi zmdi-view-agenda'></span> Especies
        </a>
      </li>
    </ul>
    <hr class="inset">
  </nav><!-- /nav -->
</div><!-- /panel -->
```

Fig. 26. Código fuente del *hamburger menu*.

Respecto al funcionamiento del *hamburger menu*, por una parte, para detectar las interacciones que despliegan el panel de navegación, se utilizaron los eventos JQuery Mobile, *tap*¹¹¹ para detectar si el usuario hace *tap* sobre el botón del menú y *swiperight*¹¹² para cuando el usuario realiza un *right drag* sobre la pantalla. Por otra parte, para detectar el *left drag* que ocultará el panel de navegación se usó el evento *swipeleft*¹¹³. Con respecto a las interacciones que el usuario lleve a cabo con las opciones de navegación del *hamburger menu*, éstas se detectarán mediante el evento *tap*. Cabe señalar que este evento JQuery Mobile corresponde a un componente de interacción esencial y se encuentra presente en todas las pantallas de la aplicación móvil.

¹¹¹ JQuery Mobile. *tap*. <<https://api.jquerymobile.com/tap/>>. [Consulta: 17/12/2017].

¹¹² JQuery Mobile. *swiperight*. <<https://api.jquerymobile.com/swiperight/>>. [Consulta: 17/12/2017].

¹¹³ JQuery Mobile. *swipeleft*. <<https://api.jquerymobile.com/swipeleft/>>. [Consulta: 17/12/2017].

Finalmente, como ya se mencionó, con el fin de otorgar a los usuarios un acceso rápido y permanente al sistema de navegación de la aplicación móvil, las opciones de navegación del *hamburger menu* se replicaron en el *footer*. A diferencia del *hamburger menu*, los enlaces de navegación del *footer* se crearon en base a elementos `<div>` y al uso del sistema Flexbox Grid¹¹⁴ incorporado en nativeDroid2, como alternativa a la *grid* propio de JQuery Mobile. Flexbox Grid funciona de manera similar al sistema *grid* de Bootstrap¹¹⁵ y se basa en uso de clases CSS que permiten dividir la pantalla hasta en doce columnas. El *footer* se estructuró de esta manera con el fin de darle una apariencia similar a la sugerida por Google para las opciones de navegación inferior¹¹⁶ y además, considerando la posibilidad de agregar más de cinco opciones de navegación, lo cual no podría realizarse usando el *grid* propio de JQuery Mobile. Finalmente, cabe mencionar que, el *footer* está definido semánticamente mediante la etiqueta `<footer>` que permite precisar el pie de página de un documento HTML.

```

<!-- pie de página-->
<footer data-role="footer" data-position="fixed">
  <div class="row_center-xs">
    <div class="col-xs-4">
      <div class="box">
        <a href="about.html" class="ui-btn ui-btn-icon-block">
          <i class='zmdi zmdi-info-outline zmd-2x'></i>Acerca de
        </a>
      </div>
    </div>
    <div class="col-xs-4">
      <div class="box">
        <a href="index.html" class="ui-btn ui-btn-icon-block">
          <i class='zmdi zmdi-apps zmd-2x'></i>Grupos
        </a>
      </div>
    </div>
    <div class="col-xs-4">
      <div class="box">
        <a href="species.html" class="ui-btn ui-btn-icon-block">
          <i class='zmdi zmdi-view-agenda zmd-2x'></i>Especies
        </a>
      </div>
    </div>
  </div><!-- /row -->
</footer><!-- /footer -->

```

Fig. 27. Código fuente del *footer*.

¹¹⁴ @dam. Flexbox grid. <<http://flexboxgrid.com/>>. [Consulta: 17/12/2017].

¹¹⁵ Bootstrap. Grid system. <<https://getbootstrap.com/docs/4.0/layout/grid/>>. [Consulta: 17/12/2017].

¹¹⁶ Google. Bottom navigation. <<https://material.io/guidelines/components/bottom-navigation.html>>. [Consulta: 17/12/2017].

Por último, es necesario precisar que siguiendo las recomendaciones de navegación inferior y *navigation drawers* de Google, se agregaron íconos en los elementos de las opciones de navegación, tanto del *hamburger menu* como del *footer*, con el fin de facilitar a los usuarios la comprensión de las acciones asociadas a las opciones de navegación. A continuación, se presenta el resultado final de la construcción del *hamburger menu* y el *footer* para la aplicación móvil sobre fauna chilena.

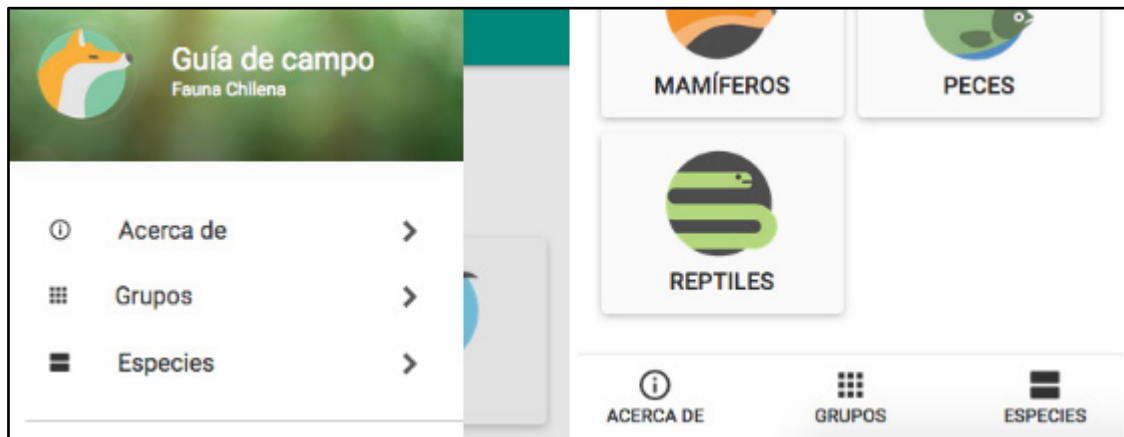


Fig. 28. De izquierda a derecha, *hamburger menu* y *footer* de la aplicación.

Una primera pantalla fundamental de la aplicación móvil sobre fauna chilena es la *home screen*, en la cual se muestran los diferentes grupos de animales disponibles. Como ya se mencionó, en su construcción influirá el patrón de composición pantalla de inicio, por lo que esta debe ofrecer un conjunto predeterminado de opciones una vez que la aplicación móvil haya iniciado. Con base en lo anterior, para crearla se utilizó una etiqueta `<div>` para definir cada uno de los grupos de animales, en este caso concreto cinco etiquetas, para los grupos, anfibios, aves, mamíferos, peces y reptiles. Cabe señalar que el elemento `<div>` permite definir bloques de contenido dentro de los documentos HTML. Como ya se mencionó, esta pantalla se basa en el patrón *grid*, el cual permite mostrar una matriz de elementos seleccionables, por lo que, con el objetivo de mostrar los grupos de manera estructurada, se utilizaron los *grids* de JQuery Mobile. En términos simples, un *grid* permite colocar elementos en columnas una al lado de la otra (Chetan, 2012). Es posible especificar hasta cinco columnas dentro de un *grid* según el valor definido para el atributo `class` (`ui-block-a/b/c/d/e`). En este caso específico, dentro del elemento `<main>` que contiene los `<div>` de cada grupo de animales, se definió el atributo `class="ui-grid-a"` que permite dividir la pantalla en dos columnas de igual ancho (50%/50%). Para utilizar estas columnas, el primer `<div>` posee un atributo `class="ui-block-a"`, el segundo un atributo `class="ui-block-b"`, el tercero un atributo `class="ui-block-a"` y así de manera sucesiva según sea necesario. Dentro de cada elemento `<div>` se encuentra una etiqueta `<a>`. El elemento HTML *anchor* `<a>` posibilita crear un enlace a otras páginas de Internet, archivos o ubicaciones dentro del mismo documento HTML, direcciones de correo, o cualquier otra URL (Mozilla, 2017). En este caso específico, se utiliza el elemento *anchor* para definir enlaces textuales a las páginas que contienen el listado de especies de cada grupo. Con el objetivo de otorgar a los enlaces un *look and feel* similar al de los botones usados en las GUI de Google, se utiliza la clase CSS

ui-btn-raised de nativeDroid (*default button raised*)¹¹⁷. Dentro de cada botón se incorpora una representación gráfica (ícono) del grupo de especies correspondiente, para facilitar al usuario la comprensión de la pantalla destino. Para incorporar los íconos se utilizó la etiqueta ``, elemento que permite incrustar imágenes en los documentos HTML.

```
<!-- contenido home screen -->
<main role="main" class="ui-content ui-grid-a" data-inset="false">
  <h2>Vertebrados</h2>
  <div class="ui-block-a">
    <a href="amphibians.html" class="ui-btn ui-btn-raised"
      data-role="button">
      
      <span>Anfibios</span>
    </a>
  </div>
  <div class="ui-block-b">
    <a href="birds.html" class="ui-btn ui-btn-raised"
      data-role="button">
      
      <span>Aves</span>
    </a>
  </div>
</main>
```

Fig. 29. Código fuente: elementos de la *home screen*.

Finalmente, cabe señalar que la utilización de íconos, en desmedro de otros elementos gráficos como fotografías, se justifica considerando el enfoque didáctico y educativo que se le podría dar a esta aplicación luego de su implementación. A continuación, se presenta el resultado final de la construcción de la pantalla *home screen*.

¹¹⁷ Godesign Webpublishing. *Buttons*. <<http://nd2.godesign.ch/examples/elements/buttons.html>>. [Consulta: 20/12/2017].

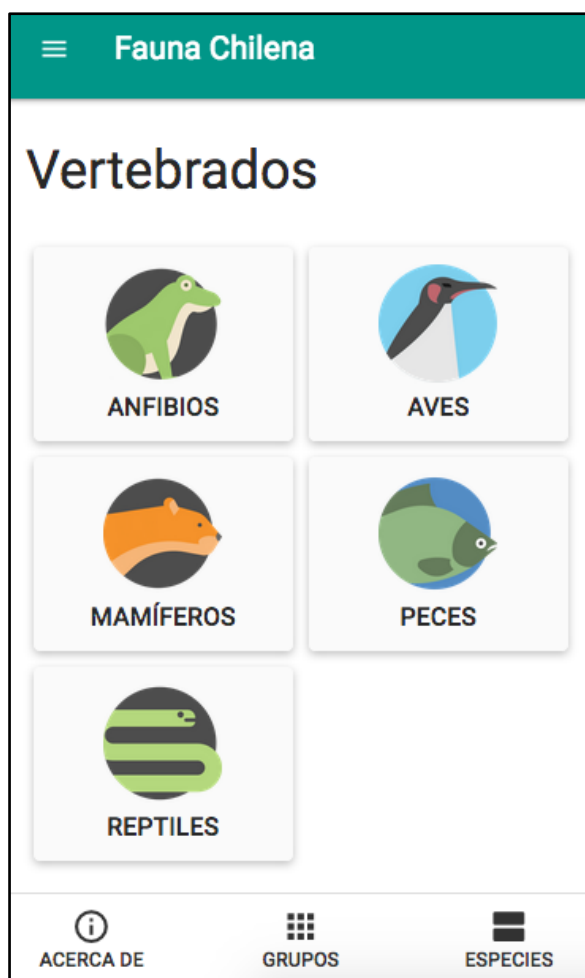


Fig. 30. Home screen de la aplicación.

Una segunda pantalla esencial de la aplicación móvil sobre animales chilenos es aquella pantalla que contiene el listado de especies, la cual se desplegará cuando el usuario escoja alguno de los grupos de la *home screen*. Como se explicó antes, en su construcción se basa en los patrones lista vertical y lista de miniaturas, por lo que, para construir este listado de elementos, al igual que en el caso del *hamburger menu*, se utilizó el atributo *data-role* de JQuery Mobile con valor "*listview*", en conjunto con las etiquetas HTML `` y ``. Cada elemento `` de la lista está compuesto por una etiqueta `<a>` enlazada a una pantalla con información detallada sobre cada animal. A su vez, cada elemento `<a>` está conformado por una imagen del animal, insertada mediante la etiqueta ``, cuyo fin es permitir que el usuario comprenda de qué especie se trata; un encabezado de nivel 2 con el nombre común del animal; y un párrafo con su nombre científico. Ambos componentes textuales, se alinearon a la izquierda con el fin de facilitar el escaneo visual por parte de los usuarios. Es preciso señalar que, se personalizó el estilo gráfico de la clase CSS *listview* de JQuery MOBILE, de forma tal que las imágenes de los animales ocupasen todo el ancho y todo el alto de cada elemento ``. Con el fin de permitir al usuario encontrar rápidamente una determinada especie, se implementó debajo de la barra de títulos de esta pantalla, una caja de búsqueda de tipo *persistent search* en base al patrón "buscar dentro". El pilar fundamental de este sistema de búsqueda corresponde al atributo

data-filter, propio del *widget filterable*¹¹⁸. Este atributo permite filtrar elementos secundarios dentro de los distintos *widgets* de JQuery Mobile como por ejemplo, dentro de un *listview*, de un tabla, en un conjunto de botones *controlgroup*, etc. Lo anterior, permite acotar de manera automática los elementos de la lista de especies a medida que el usuario ingresa su consulta. Por defecto, el filtro en sí, se lleva cabo sobre el contenido textual de los componentes JQuery Mobile, por lo que en este caso específico el filtro se aplicará sobre el encabezado de nivel 2 con el nombre común del animal y sobre el párrafo con el nombre científico. Además de lo anterior, por medio del atributo *data-filtertext* del *widget filterable*, es posible asociar determinados *keywords* a los elementos del *listview*, lo cual permite en este caso, por ejemplo, generar relaciones asociativas entre una especie animal y sus características o comportamientos particulares, relaciones de equivalencia para especies con más de un nombre común, o agregar como opción de filtrado cualquier otro metadato resultante del proceso de indización. Sin embargo, es necesario mencionar que al utilizar el atributo *data-filtertext*, el filtro ignora completamente el contenido textual de los elementos del *listview*, por lo que, cuando se utilizó este atributo en el listado de especies, tanto el nombre común como el nombre científico del animal fueron agregados como *keywords* para que fueran considerados en la búsqueda. Respecto a la apariencia de la caja de búsqueda, usando el atributo *data-filter-placeholder* se personalizó el texto de esta, definiéndolo simplemente como “Buscar”, siguiendo el diseño de las cajas de búsqueda de aplicaciones móviles de Google.

```

<!-- lista de mamíferos -->
<ul id="mammalslist" data-role="listview" data-filter="true"
  data-filter-placeholder="Buscar" data-inset="true">
  <li data-filtertext="Huemul Hippocamelus bisulcus ciervo sur andino">
    <a href="huemul.html">
      
      <h2>Huemul</h2>
      <p>Hippocamelus bisulcus</p>
    </a>
  </li>
  <li data-filtertext="Monito de monte Chimaihuen Dromiciops gliroides">
    <a href="monito.monte.html">
      
      <h2>Monito de monte, Chimaihuen</h2>
      <p>Dromiciops gliroides</p>
    </a>
  </li>
</ul>

```

Fig. 31. Código fuente del *listview* de especies.

¹¹⁸ JQuery Mobile. *Filterable*. <<http://demos.jquerymobile.com/1.4.5/filterable/>>. [Consulta: 25/12/2017].

Como ya se mencionó en el apartado de Diseño, con el fin de que el usuario pueda borrar la consulta ingresada se integró un botón “x”. Finalmente, cabe señalar que cuando los elementos del listado exceden el *viewport*, es posible para el usuario hacer *scroll* para visualizar el listado completo. A continuación, se presenta el resultado final de la pantalla con el listado de especies, en este caso específico el listado de mamíferos.



Fig. 32. Pantalla listado de mamíferos de la aplicación.

Una tercera pantalla elemental para la aplicación sobre animales chilenos es la ficha informativa que se abrirá al escoger un elemento del listado de especies. Como se explicó con anterioridad, esta pantalla se basa en el patrón *tab*, por lo que para implementarla se utilizó el *widget tab*¹¹⁹ de nativeDroid2. Este *widget* posibilita la creación de *tabs* mediante la combinación de dos atributos *data-role*. Por una parte, el atributo *data-role="nd2tabs"* hace posible precisar el nombre de las *tabs* del encabezado, al definirlo dentro de un elemento ``.

¹¹⁹ Godesign Webpublishing. *Tabs*. <<http://natedroid.scripiter.click/tabs/>>. [Consulta: 25/12/2017].


```

<!-- tabs con información sobre la especie -->
<ul data-role="nd2tabs" data-swipe="true">
  <li data-tab="detalles" data-tab-active="true">Detalles</li>
  <li data-tab="habitat">Hábitat</li>
  <li data-tab="conservacion">Conservación</li>
</ul>

```

Fig. 33. Código fuente: definición de tabs mediante *nd2tabs*.

Por otra parte, el atributo *data-role="nd2tab"* permite crear el contenido de una cada una de las *tabs* definidas mediante *nd2tabs*. Cabe señalar que, el vínculo entre las *tabs* del encabezado y el contenido de las mismas se lleva a cabo por medio del atributo *data-tab="nombre-tab"*, incorporándose en tanto en los elementos ** del encabezado como en los elementos HTML que engloban el contenido de la correspondiente *tab*, habitualmente bloques *<div>*. Finalmente, es preciso mencionar que, con el fin de diferenciar las diferentes secciones de contenido incluidas dentro de cada *tab*, se utilizó la clase CSS *Cards*¹²⁰ de *nativeDroid2*. Esta clase permite definir un estilo gráfico similar al del componente *Card*¹²¹, recomendado por Google para mostrar contenido compuesto por diferentes elementos.

```

<!-- tab detalles -->
<section data-role="nd2tab" data-tab="detalles">
  <!-- tarjeta con fotografía -->
  <div class="nd2-card">
    <!-- fotografía -->
    <div class="card-media">
      
    </div>
    <!-- nombre común y nombre científico -->
    <div class="card-title has-supporting-text">
      <h2 class="card-primary-title">Ratón lanudo común</h2>
      <p class="card-subtitle">Abrothrix longipilis</p>
    </div>
  </div><!-- .nd2-card -->
  <!-- rasgos distintivos -->
  <div class="nd2-card">
    <div class="card-title has-supporting-text">
      <h2 class="card-primary-title">Rasgos distintivos</h2>
    </div>
    <div class="card-supporting-text has-action has-title">
      <p>El ratón lanudo común es un roedor que pesa entre 30
    </div>
  </div><!-- .nd2-card -->

```

Fig. 34. Código fuente: *tab* con detalles generales de la especie.

¹²⁰ Godesign Webpublishing. *Cards*. <<http://natedroid.scripster.click/cards/>>. [Consulta: 25/12/2017].

¹²¹ Google. *Cards*. <<https://material.io/guidelines/components/cards.html>>. [Consulta: 25/12/2017].

A continuación, se presenta el resultado final de la ficha informativa de especies, en este caso específico sobre el animal Ratón lanudo común.

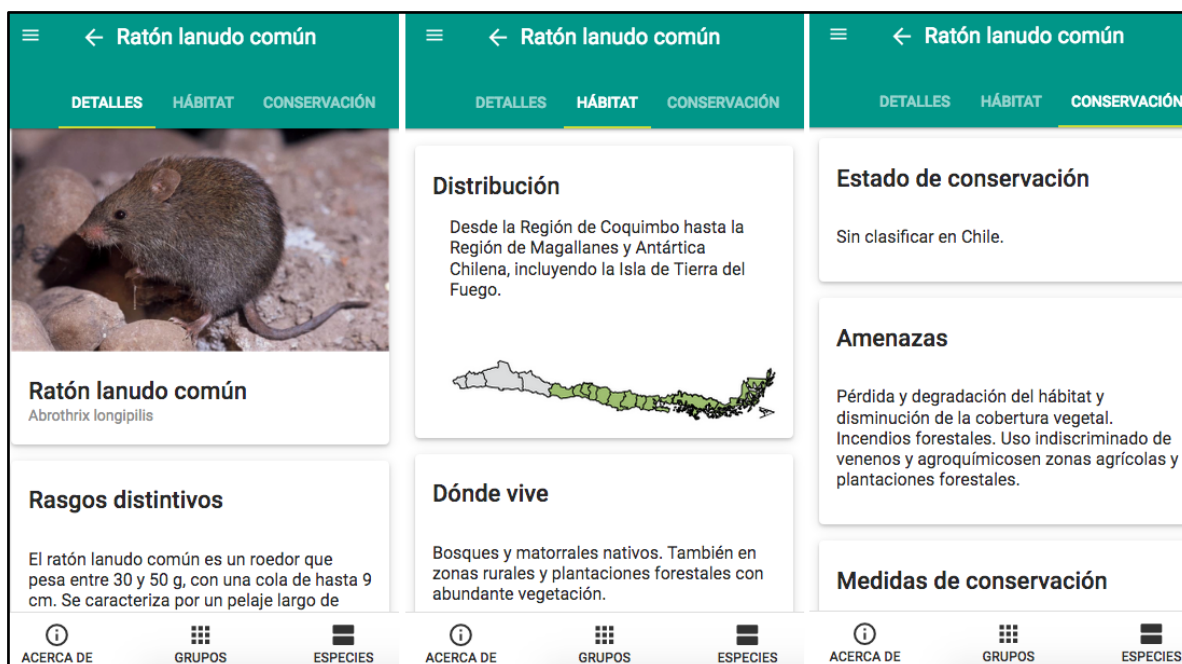


Fig. 35. Pantalla ficha informativa especie.

Cabe señalar que, para llevar a cabo esta prueba de concepto, los datos incluidos en las fichas informativas fueron obtenidos desde el inventario nacional de especies de Chile¹²².

Finalmente, un detalle relevante respecto a las fichas informativas de cada animal es que corresponden a páginas estáticas, es decir por cada especie incluida en la aplicación existe un documento HTML con la información correspondiente. Lo anterior, repercute directamente, tanto en tiempo como el coste de desarrollo de una versión definitiva de la aplicación, es decir, mientras mayor sea el número de especies incluidas, mayor será el tiempo y el coste de la implementación. Por último, otro factor estrechamente ligado a la cantidad de fichas informativas es el tamaño de la aplicación, el cual aumentará mientras más fichas existan. Sin embargo, y luego de analizar proyectos similares, se recomienda que almacenamiento de datos se lleve a cabo de manera local, para prescindir de una conexión permanente a Internet y favorecer así la movilidad del usuario, por ejemplo, cuando esté en terreno y no disponga de una conexión a la Red estable.

¹²² Ministerio del Medio Ambiente. *Inventario nacional de especies de Chile*. <<http://especies.mma.gob.cl/CNMWeb/Web/WebCiudadana/Default.aspx>>. [Consulta: 25/12/2017].

6.3. Optimización en tiendas de aplicaciones

6.3.1. Definición de ASO

Como ya se explicó anteriormente, una vez que una aplicación móvil híbrida se encuentra construida, es necesario someterla a un proceso de revisión y aprobación, para poder publicarla en una determinada tienda de aplicaciones. Considerando la enorme cantidad de aplicaciones sobre temáticas similares, es necesario implementar algún tipo de estrategia que permita que la aplicación aparezca en los primeros puestos de las SERP de una determinada tienda de aplicaciones. Este tipo de estrategias de posicionamiento se viene aplicando en sitios web desde hace bastante tiempo y se conoce como optimización en motores de búsqueda (SEO, por su sigla en inglés). El SEO corresponde a un conjunto de técnicas para optimizar el contenido de un sitio web, con el objetivo de alcanzar el mejor posicionamiento posible en motores de búsqueda de Internet, como por ejemplo Google o Bing¹²³. En otras palabras, SEO consiste en ayudar a los motores de búsqueda a entender y presentar el contenido de un sitio web (Google, 2017). En definitiva, el SEO hace posible mejorar la visibilidad de un sitio web, para lograr que este aparezca entre los primeros resultados de la SERP de un motor de búsqueda, para determinadas consultas hechas por los usuarios, lo cual se traduce en un mayor número de usuarios visitantes. En el caso de las aplicaciones móviles, para lograr posicionarlas en los resultados de las SERP de tiendas de aplicaciones se utiliza la optimización en tiendas de aplicaciones (ASO, por su sigla en inglés). ASO corresponde al proceso de optimización de una aplicación o juego con el objetivo de maximizar su visibilidad en Apple App Store, Google Play Store u otra tienda de aplicaciones, incrementar el tráfico hacia la ficha de la aplicación y mejorar la tasa de conversión (en inglés *conversion rate*¹²⁴) para generar el máximo volumen de instalaciones (PickASO; TheTool; Tribal Worldwide; IAB Spain, 2017).

6.3.2. Factores ASO

Para que ASO funcione es necesario trabajar una serie de factores que influyen directamente en el posicionamiento de aplicaciones en las tiendas, es decir, factores que impactan tanto la visibilidad como el *conversion rate* de una aplicación móvil. Estos factores se dividen en internos y externos. Por una parte, los factores internos se denominan factores ASO *On-Metadata* y corresponden a aquellos que dependen directamente de los desarrolladores, ya que pueden ser editados desde las consolas de desarrollador, iTunes Connect¹²⁵ y Google Play Console¹²⁶ (PickASO; TheTool; Tribal

¹²³ Microsoft. *Bing*. <<https://www.bing.com/>>. [Consulta: 25/12/2017].

¹²⁴ En ASO, el *conversion rate* corresponde al porcentaje de usuarios que visita la ficha de una aplicación y termina descargando la aplicación.

¹²⁵ Apple. *iTunes Connect*. <<https://www.apple.com/es/itunes/working-itunes/sell-content/connect/>>. [Consulta: 25/12/2017].

¹²⁶ Google. *Google Play Console*. <<https://developer.android.com/distribute/console/index.html?hl=es>>. [Consulta: 25/12/2017].

Worldwide; IAB Spain, 2017). A continuación, se describen los factores ASO *On-Metadatos* que influyen en el posicionamiento de las aplicaciones subidas Google Play.

Un primer factor es el título de la aplicación móvil, el cual debe contener las *keywords* mediante las que el público objetivo del proyecto pueda encontrar la aplicación, además se sugiere que este sea llamativo. El título de la aplicación puede contener como máximo hasta 50 caracteres.

Un segundo factor *On-Metadatos* es la descripción larga, la que, al igual que el título, debe incorporar las *keywords*, pero además debe incluir argumentos que convencan al usuario de descargar la aplicación. La descripción larga puede tener hasta 4.000 caracteres. Un tercer factor *On-Metadatos* es la descripción corta mediante la cual es posible describir la aplicación usando un máximo de 80 caracteres.

Otro factor *On-Metadatos* corresponde al nombre del desarrollador de la aplicación, el cual también posiciona a nivel de búsqueda en las tiendas. Además, Google favorece la visibilidad de aquellas aplicaciones cuyos desarrolladores tengan mayor antigüedad en Google Play.

La categoría es también otro factor *On-Metadatos* que, si bien no influye directamente a nivel de posicionamiento, sí es importante para los usuarios que navegan por las diferentes categorías de aplicaciones de Google Play. Cabe mencionar que en Google Play es posible escoger solo una categoría para una aplicación móvil, a diferencia de App Store donde se pueden elegir dos categorías, una principal y una secundaria.

Además de los factores *On-Metadatos* textuales descritos, existen factores *On-Metadatos* visuales que permiten al usuario comprender el objetivo y el funcionamiento de una aplicación, al observarlos en su ficha. Un primer factor visual es el ícono. El ícono debe mostrar de manera clara cuál es la función de la aplicación y además debe ser visualmente atractivo con el fin de destacar entre los demás íconos de las aplicaciones que aparezcan en los resultados de búsqueda de Google Play.

Un segundo factor *On-Metadatos* visual corresponde a la imagen destacada, esta resulta muy relevante ya que se ubica en la parte superior de la ficha de la aplicación. Al igual que el ícono, debe ser visualmente atractiva, para convencer al usuario de descargar la aplicación. Un tercer factor visual son las capturas de pantalla (en inglés *screenshots*). Las *screenshots* deben centrarse en aspectos relacionados con las principales funcionalidades de la aplicación, de manera que se transmita al usuario todas las ventajas de esta, pudiendo incluir explicaciones de funcionamiento. Cabe mencionar que Google Play permite incorporar hasta cinco *screenshots* por aplicación.

Un último factor *On-Metadatos* visual corresponde al video. Este elemento visual debe ser descriptivo acerca de las funcionalidades de la aplicación, aunque también es posible otorgarle un enfoque publicitario. Google Play permite agregar un video a la ficha de la aplicación, mediante la plataforma

YouTube. Finalmente, es preciso mencionar que, en el caso de Google Play Store, la totalidad de los factores *On-Metadata* son visibles en la ficha de cualquier aplicación móvil.

Por otra parte, los factores externos se conocen como factores *ASO Off-Metadata* y son aquellos que los desarrolladores no pueden controlar. Lo anterior debido a que dependen casi en su totalidad de la respuesta del mercado al producto lanzado, y del presupuesto de marketing disponible para un determinado proyecto (PickASO; TheTool; Tribal Worldwide; IAB Spain, 2017).

Los factores *Off-Metadata* que influyen en el posicionamiento de las aplicaciones alojadas en Google Play son: el volumen de instalaciones, es decir, mientras más instalaciones tenga una aplicación mejor posicionará; la velocidad de las instalaciones, vale decir, un mayor volumen de instalaciones en un tiempo reducido se traduce en un mejor posicionamiento; las valoraciones, que corresponden a la calificación otorgada por los usuarios para una determinada aplicación, mediante el sistema de estrellas de Google, donde una estrella es la peor valoración posible y cinco estrellas es la mejor; la nota media (en inglés *rating*), que corresponde al promedio de valoraciones recibidas por una aplicación; los comentarios hechos por los usuarios respecto a la aplicación, los que además influyen a nivel de búsqueda, ya que las *keywords* que aparecen en estos, son consideradas por Google al momento de posicionar una aplicación móvil; las menciones en sitios web de terceros son un factor permiten dar a conocer la aplicación, lo cual podría traducirse en nuevas descargas; y finalmente el compromiso o implicación de los usuarios con la aplicación (en inglés *engagement*), es decir un número alto de usuarios activos en la aplicación (usuarios que acceden de manera recurrente) y un bajo número de desinstalaciones, influirá en el posicionamiento en Google Play.

6.3.3. Propuesta ficha para Google Play

A partir de los factores *On-Metadata* descritos en los párrafos anteriores, se presenta a continuación una propuesta de ficha para Google Play para la aplicación móvil sobre fauna chilena. Lo anterior, mediante un *wireframe* de fidelidad media.

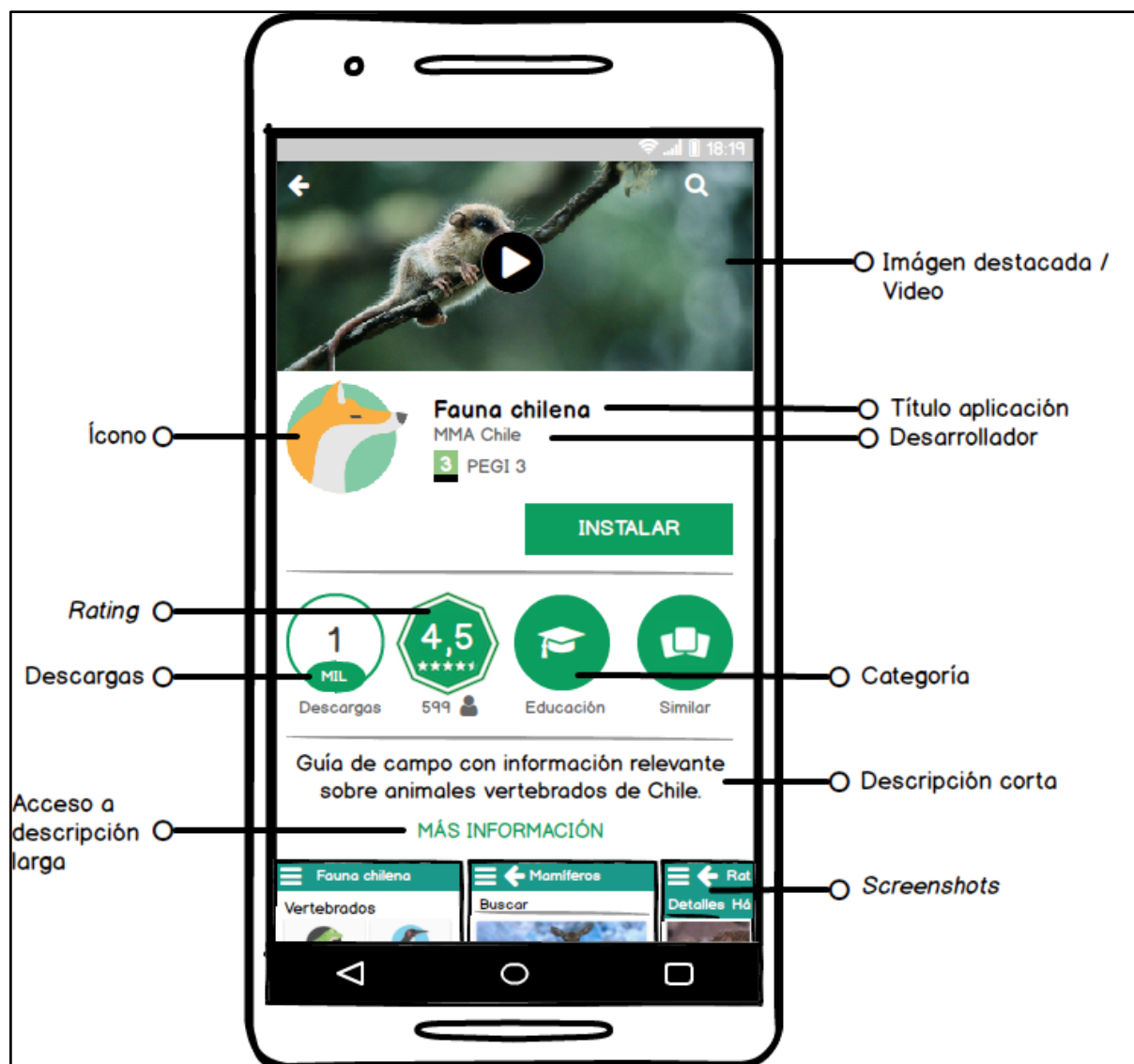


Fig. 36. Wireframe de la ficha de la aplicación Fauna chilena para Google Play.

7. Planificación y presupuesto

Con el objetivo de hacer posible la planificación de un proyecto digital referente al desarrollo de una aplicación móvil sobre fauna chilena, se presenta a continuación un escenario ficticio en el que se detallan las principales etapas, tareas y recursos necesarios para llevar a cabo su implementación.

7.1. Escenario ficticio

Considerando la abundante variedad de especies animales existentes en Chile, y con el objetivo de generar conciencia en la ciudadanía, acerca de su protección y conservación, el MMA ha encargado a su División de Recursos Naturales y Biodiversidad, llevar a cabo un proyecto tecnológico con un enfoque divulgativo, mediante el cual se ofrezca a la población del país, información de carácter descriptivo sobre la fauna existente en el territorio. Luego de realizar un análisis acerca del contexto tecnológico actual del país, la División de Recursos Naturales y Biodiversidad ha optado por llevar a cabo el desarrollo de una aplicación móvil híbrida para dispositivos Android, la cual contenga descripciones y fotografías que permitan a los usuarios identificar las diferentes especies.

Esta primera implementación estará centrada en los vertebrados nativos del país, los que, ignorando las variedades de peces marinos existentes (1182 especies), suman aproximadamente 820 especies¹²⁷. Considerando el alcance de esta primera etapa del proyecto, de las 820 especies de vertebrados nativos, se seleccionarán aproximadamente 200 para crear las fichas informativas a incorporar en la versión de lanzamiento de la aplicación. Finalmente, con el fin de aprovechar la información existente en el MMA acerca de las especies animales del país, tanto el contenido textual como las fotografías que se incorporen en la aplicación móvil serán obtenidas del inventario nacional de especies de Chile.

7.2. Recursos humanos necesarios y etapas de implementación

7.2.1. Recursos humanos

Llevar a cabo el desarrollo de una aplicación móvil sobre fauna implica contar con un equipo humano cualificado en diferentes áreas. A continuación, se describen los perfiles profesionales necesarios para el desarrollo de la aplicación móvil señalada.

- **Jefe de proyecto:** encargado de identificar y planificar cada una de las fases del proyecto, estableciendo para ello metas y plazos. Además, es quién debe coordinar las acciones de los miembros del equipo humano involucrado, manteniendo una comunicación clara y oportuna

¹²⁷ Ministerio del Medio Ambiente. *Especies nativas*.

<<http://especies.mma.gob.cl/CNMWeb/Web/WebCiudadana/pagina.aspx?id=88&pagId=85>>. [Consulta: 29/12/2017].

con cada uno. El jefe de proyecto también debe relacionarse con el cliente, escuchando opiniones e interpretar sus necesidades con el fin de lograr plasmarlas de manera acertada en el proyecto. Finalmente, este profesional debe poseer conocimientos acabados sobre gestión de proyectos digitales y dominio transversal sobre las demás áreas involucradas.

- **Programador:** encargado de generar el código fuente de la aplicación móvil. En este caso específico es necesario que el programador tenga conocimientos generales sobre el desarrollo de aplicaciones móviles en general y conocimientos específicos sobre estándares web (HTML, CSS y JavaScript) y diferentes *frameworks* para el desarrollo de aplicaciones híbridas.
- **Diseñador gráfico:** encargado de llevar a cabo la maquetación y el diseño de las GUI de la aplicación, lo que incluye generar la guía de estilos CSS y cualquier elemento gráfico necesario para la implementación, como el ícono de la aplicación y los íconos de los grupos de animales. Es necesario que el diseñador gráfico tenga nociones avanzadas de CSS y HTML.
- **Experto UX:** encargado de validar la GUI de la aplicación móvil en términos de UX. Para lograr lo anterior, debe llevar a cabo estudios de usuarios que permitan evaluar si la aplicación cumple con las necesidades o expectativas de los usuarios finales. Debe poseer conocimientos sobre usabilidad y AI en general y conocimientos acabados sobre IxD y diseño UI en entornos móviles y debe tener experiencia en el desarrollo de pruebas de usabilidad.
- **Experto en fauna chilena:** encargado de validar el contenido informativo que se incorporará en la aplicación, es decir es el profesional a cargo de corroborar que los datos existentes en el inventario nacional de especies sean fiables. Debe ser un especialista en biología y poseer conocimientos específicos sobre los animales vertebrados que habitan el territorio chileno.

7.2.2. Fases de implementación

Fase 0. Reunión inicial de proyecto

- **Descripción y tareas relacionadas:** en esta reunión se definirá el alcance del proyecto y otros aspectos fundamentales como el objetivo general y las fases necesarias para llevar a cabo su implementación. Además, se precisará el rol y las responsabilidades de cada uno de los miembros del equipo de trabajo.
- **Perfiles implicados:** equipo completo.
- **Calendario:** 1 día.

Fase 1. Análisis de referentes y estudio de usuarios

- **Descripción y tareas relacionadas:** en esta fase se llevará a cabo un análisis de proyectos similares que hayan sido implementados por otras entidades a nivel global con el fin de identificar los principales referentes del tema y rescatar las buenas prácticas que hayan implementado. También se efectuará, en esta fase, un estudio de usuarios que permita detectar las necesidades del público objetivo del proyecto, para lograrlo se realizará una

serie de entrevistas. Ambas acciones, permitirán formalizar los objetivos específicos del proyecto. De acuerdo a lo anterior, se llevarán a cabo las siguientes tareas:

- Análisis de proyectos de referencia.
- Estudio de necesidades de los usuarios.
- **Entregables:**
 - Informe buenas prácticas de referentes.
 - Informe con datos resultantes del estudio de usuarios.
- **Perfiles implicados:** jefe de proyecto y experto UX.
- **Calendario:** 5 días.

Fase 2. Definición de requisitos de la aplicación

- **Descripción y tareas relacionadas:** esta fase del proyecto se centra en definir las funcionalidades de la aplicación móvil, a partir de los datos obtenidos en el estudio de usuarios. En esta fase también se definirán los principales flujos de interacción del usuario con la aplicación. Para lograr lo anterior, se llevarán a cabo las siguientes tareas:
 - Especificación de funcionalidades.
 - Diagramación de procesos de interacción de los usuarios con la aplicación.
- **Entregables:**
 - Informe de requisitos funcionales.
 - *Wireflows* de procesos de interacción principal.
- **Perfiles implicados:** programador, experto UX y jefe de proyecto.
- **Calendario:** 5 días.

Fase 3. Definición del contenido de la aplicación

- **Descripción y tareas relacionadas:** en esta fase del proyecto se revisarán y validarán las fichas informativas de las diferentes especies disponibles en el inventario nacional de especies, con el fin de incorporar esta información a la aplicación.
- **Entregables:** Informe de revisión y validación de las fichas de especies.
- **Perfiles implicados:** experto en fauna chilena y jefe de proyecto.
- **Calendario:** 10 días.

Fase 4. Definición de AI y diseño de la aplicación

- **Descripción y tareas relacionadas:** en esta fase del proyecto se determinará la AI de la aplicación móvil y se llevará a cabo el prototipado de la GUI y la definición del IxD, en base a patrones de diseño ampliamente difundidos y según el SO móvil para el que se desarrollará.
 - Definición de AI.
 - Diseño de pantallas.
- **Entregables:**
 - Mapa AI de la aplicación.

- *Wireframes* de las pantallas de la aplicación.
- **Perfiles implicados:** diseñador gráfico y experto UX.
- **Calendario:** 5 días.

Fase 5. Reunión de validación

- **Descripción y tareas relacionadas:** en esta reunión se llevará a cabo la validación de los entregables desarrollados hasta el momento, antes de comenzar con el desarrollo informático de la aplicación.
- **Perfiles implicados:** equipo completo.
- **Calendario:** 1 día.

Fase 6. Desarrollo y testeo técnico de la aplicación

- **Descripción y tareas relacionadas:** durante esta fase se llevará a cabo la programación de la aplicación en un entorno de pruebas. Además, en paralelo se realizarán las primeras pruebas de funcionamiento de la aplicación. Esta etapa conlleva las siguientes tareas:
 - Programación de código fuente.
 - Primeras pruebas de funcionamiento.
- **Entregables:** versión prototipo.
- **Perfiles implicados:** programador.
- **Calendario:** 15 días.

Fase 7. Test de usabilidad

- **Descripción y tareas relacionadas:** durante esta etapa se llevará a cabo una serie de test de usabilidad sobre la versión prototipo de la aplicación móvil. Llevar a cabo los test de usabilidad implica realizar las siguientes tareas:
 - Captación de usuarios.
 - Realización de test.
 - Análisis de resultados.
- **Entregables:** informe de usabilidad con observaciones y recomendaciones.
- **Perfiles implicados:** experto UX.
- **Calendario:** 5 días.

Fase 8. Corrección de incidencias

- **Descripción y tareas relacionadas:** durante esta fase del proyecto se aplicarán los cambios necesarios a la versión prototipo de la aplicación a partir de los resultados obtenidos en los test de usabilidad. Lo anterior, en caso de que se hayan detectado incidencias.
- **Entregables:** versión prototipo con incidencias corregidas.
- **Perfiles implicados:** programador y experto UX.
- **Calendario:** 2 días.

Fase 9. Diseño gráfico

- **Descripción y tareas relacionadas:** en esta fase del proyecto diseña la apariencia gráfica de la aplicación móvil por medio de la definición de la guía de estilos CSS y de los elementos gráficos necesarios para la GUI, ya sean íconos, botones, logos, etc. Durante esta fase se realizarán las siguientes tareas:
 - Creación de elementos gráficos.
 - Definición de guía de estilos CSS.
 - Aplicación de estilo gráfico a la versión prototipo.
- **Entregables:**
 - Conjunto de elementos gráficos de la GUI.
 - Guía de estilos CSS.
 - Versión final de la aplicación.
- **Perfiles implicados:** diseñador gráfico.
- **Calendario:** 7 días.

Fase 10. Implementación de la aplicación

- **Descripción y tareas relacionadas:** en esta fase se lleva a cabo el traspaso de la aplicación desde el entorno de pruebas a la tienda Google Play para que pase por el proceso de aprobación correspondiente. Si todo marcha correctamente la aplicación se encontrará disponible para descarga y podrá llevarse a cabo su lanzamiento.
 - Creación de ficha Google Play para la aplicación.
 - Subida de la aplicación a Google Play¹²⁸.
 - Corrección de incidencias detectadas por Google Play.
- **Perfiles implicados:** programador.
- **Calendario:** 3 días¹²⁹.

Fase 11. Reunión final de proyecto

- **Perfiles implicados:** equipo completo.
- **Calendario:** 1 día.

¹²⁸ Esta tarea se repetirá en caso de que Google Play detecte incidencias.

¹²⁹ Si bien el proceso de aprobación de aplicaciones de la tienda Google Play dura en promedio unas cuatro horas, se prevén tres días para llevar a cabo modificaciones, en caso de que sea necesario llevarlas a cabo y volver a pasar por el proceso de aprobación.

7.2.3. Cronograma

A continuación, se presenta un diagrama de Gantt¹³⁰, mediante el cual se expone el tiempo de dedicación previsto para llevar a cabo las diferentes tareas de cada fase del proyecto.

Fases	Fecha de inicio	Duración (días)	Fecha de término
Fase 0. Reunión inicial de proyecto	8-1-2018	1	8-1-2018
Fase 1. Análisis de referentes y estudio de usuarios	9-1-2018	5	13-1-2018
Tarea 1.1. Análisis de proyectos de referencia	9-1-2018	2	10-1-2018
Tarea 1.2. Estudio de necesidades de los usuarios	9-1-2018	5	13-1-2018
Fase 2. Definición de requerimientos de la aplicación	14-1-2018	5	18-1-2018
Tarea 2.1. Especificación de funcionalidades	14-1-2018	2	15-1-2018
Tarea 2.2. Diagramación de procesos de interacción	16-1-2018	3	18-1-2018
Fase 3. Definición del contenido de la aplicación	9-1-2018	10	18-1-2018
Tarea 3.1. Análisis y validación de información	9-1-2018	10	18-1-2018
Fase 4. Definición de AI y diseño de la aplicación	19-1-2018	5	23-1-2018
Tarea 4.1. Definición de AI	19-1-2018	2	20-1-2018
Tarea 4.2. Diseño de pantallas	21-1-2018	3	23-1-2018
Fase 5. Reunión de validación	23-1-2018	1	23-1-2018
Fase 6. Desarrollo y testeo técnico de la aplicación	24-1-2018	15	7-2-2018
Tarea 6.1. Programación de código fuente	24-1-2018	14	6-2-2018
Tarea 6.2. Primeras pruebas de funcionamiento	7-2-2018	1	7-2-2018
Fase 7. Test de usabilidad	8-2-2018	5	12-2-2018
Tarea 7.1. Captación de usuarios	8-2-2018	2	9-2-2018
Tarea 7.2. Realización de test	10-2-2018	1	10-2-2018
Tarea 7.3. Análisis de resultados	11-2-2018	2	12-2-2018
Fase 8. Corrección de incidencias	13-2-2018	2	14-2-2018
Tarea 8.1. Corrección de incidencias sobre prototipo	13-2-2018	2	14-2-2018
Fase 9. Diseño gráfico	15-2-2018	7	21-2-2018
Tarea 9.1. Creación de elementos gráficos	15-2-2018	2	16-2-2018
Tarea 9.2. Definición de guía de estilos CSS	17-2-2018	4	20-2-2018
Tarea 9.3. Aplicación de estilo gráfico al prototipo	21-2-2018	1	21-2-2018
Fase 10. Implementación de la aplicación	22-2-2018	3	24-2-2018
Tarea 10.1. Creación de ficha Google Play	22-2-2018	1	22-2-2018
Tarea 10.2. Subida de la aplicación a Google Play	23-2-2018	1	23-2-2018
Tarea 10.3. Corrección de incidencias Google Play	24-2-2018	1	24-2-2018
Fase 11. Reunión final de proyecto	25-2-2018	1	25-2-2018
Duración total del proyecto	8-1-2018	49	25-2-2018

Tabla 5. Diagrama de Gantt desarrollo aplicación fauna chilena MMA. Primera parte.

¹³⁰ Como ya se mencionó previamente, las fichas informativas corresponden a páginas estáticas, por lo que el cálculo de la duración de las tareas: 3.1 Análisis y validación de las especies del inventario nacional y 6.1: Programación del código fuente, de este diagrama de Gantt, se basó en las 200 fichas informativas consideradas para esta primera implementación. Es necesario recalcar que el tiempo de desarrollo puede variar según el número de especies con las que se lanzará la aplicación.

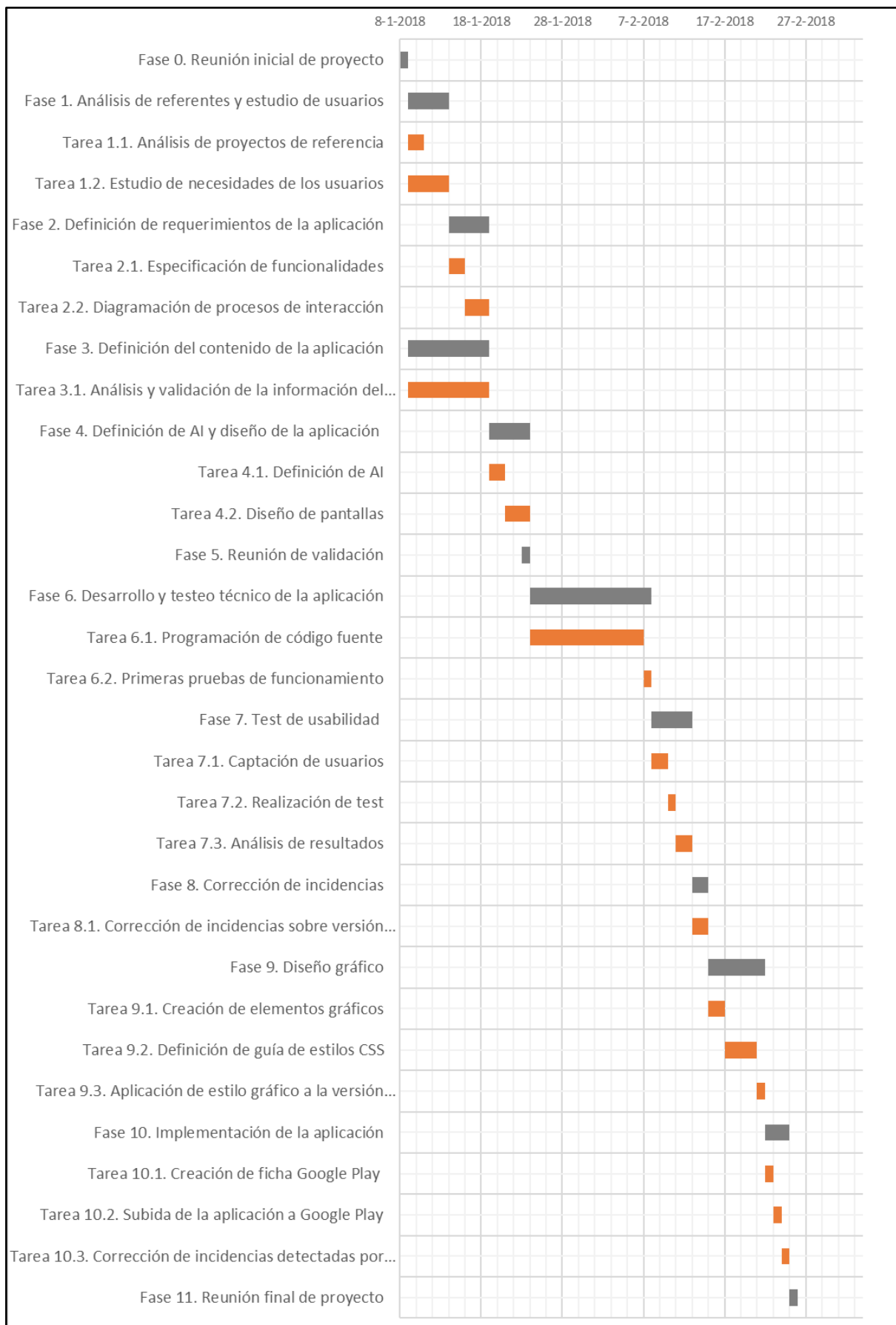


Fig. 37. Diagrama de Gantt desarrollo aplicación fauna chilena MMA. Segunda parte.

7.2.4. Presupuesto

A continuación, se detalla el número de horas que cada profesional deberá dedicar a las diferentes tareas del proyecto y se presenta además un presupuesto tentativo final.

Tareas	Jefe de proyecto	Programador	Diseñador gráfico	Experto UX	Experto fauna
Reunión inicial de proyecto	3	2	2	2	2
Tarea 1.1. Análisis de proyectos	16				
Tarea 1.2. Estudio de necesidades	8			40	
Tarea 2.1. Especificación de funcionalidades	4	16		16	
Tarea 2.2. Diagramación de procesos de interacción		8		24	
Tarea 3.1. Análisis y validación de la información del inventario nacional	4				80
Tarea 4.1. Definición de AI				16	
Tarea 4.2. Diseño de pantallas			24	8	
Reunión de revisión de avance	2	2	2	2	2
Tarea 6.1. Programación		112			
Tarea 6.2. Primeras pruebas de funcionamiento		8			
Tarea 7.1. Captación de usuarios				16	
Tarea 7.2. Realización de test				8	
Tarea 7.3. Análisis de resultados				16	
Tarea 8.1. Corrección de incidencias sobre prototipo		16		8	
Tarea 9.1. Creación de elementos gráficos			16		
Tarea 9.2. Definición de guía estilos			32		
Tarea 9.3. Aplicación de estilo gráfico a la versión prototipo			8		
Tarea 10.1. Creación de ficha Google Play		8			
Tarea 10.2. Subida de la aplicación a Google Play		4			
Tarea 10.3. Corrección de incidencias Google Play		8			
Reunión final de proyecto	1	1	1	1	1
Total horas	38	185	85	157	85
Coste por hora	30 €	15 €	15 €	20 €	30 €
Coste total	1.140 €	2.775 €	1.275 €	3.140 €	2.550 €

Tabla 6. Desglose de horas y coste monetario de cada perfil implicado.

Concepto	Coste por hora	Horas	Total
Gastos directos			
Recursos humanos			
Jefe de proyecto	30 €	38	1.140 €
Programador	15 €	185	2.775 €
Diseñador gráfico	15 €	85	1.275 €
Experto UX	20 €	157	3.140 €
Experto fauna chilena	30 €	85	2.550 €
Total			10.880 €
Contingencias		10%	1.088 €
Total recursos humanos			11.968 €
Recursos materiales			
Material fungible			100 €
Alquiler sala y equipo para el test de usuarios (1 día 8h.)			700 €
Recompensa estudios de usuarios			300 €
Total			1.100 €
Contingencias		10%	110 €
Total recursos materiales			1.210 €
Total gastos directos			13.178 €
Gastos indirectos			
Gastos varios (alquiler, ordenadores, luz, agua, Internet, etc.)			1.000 €
Total gastos indirectos			1.000 €
Total proyecto sin IVA			14.178 €
Total proyecto con IVA (21%)			17.155 €

Tabla 7. Presupuesto final del proyecto.

8. Conclusiones y trabajo futuro

8.1. Conclusiones

El trabajo realizado ha hecho posible obtener una visión global del contexto tecnológico actual respecto del uso de dispositivos móviles en Chile y en el mundo, permitiendo evidenciar la relevancia que estos han cobrado en la vida cotidiana de las personas. Prueba de ello es que hoy en día existen más usuarios de dispositivos móviles que personas con acceso a Internet. En este sentido, un factor que ha contribuido a la masificación de los dispositivos móviles es la disminución de la brecha entre sus prestaciones técnicas y las de ordenadores, principalmente debido al avance constante en la miniaturización de componentes electrónicos. Además, es probable que esta tendencia continúe hasta un punto de convergencia total entre dispositivo móvil y ordenador.

Respecto al desarrollo de aplicaciones móviles, se ha podido constatar que se trata de un sector emergente ya consolidado, cuyo crecimiento continuará durante los próximos años. La llegada de los teléfonos inteligentes y los nuevos modelos de desarrollo y comercialización de aplicaciones ha implicado un cambio de paradigma tecnológico. En este sentido, es preciso señalar que, desde los inicios de la informática el mercado del software ha estado restringido a las grandes compañías tecnológicas. Sin embargo, el contexto actual del desarrollo de software hace posible que cualquier persona o institución pueda llevar a cabo la implementación de una idea determinada por medio de la construcción de una aplicación móvil. Esto posibilita una competencia de igual a igual con los gigantes tecnológicos, ya sea desde las tiendas de aplicaciones o través de la Web.

En relación con las diferentes aproximaciones a la hora de distribuir contenidos a través de dispositivos móviles, cabe mencionar que, la inclinación por una u otra opción dependerá exclusivamente del contexto de desarrollo y las necesidades particulares de cada proyecto. Cabe señalar que, hace tan solo un par de años, desarrollar una aplicación móvil utilizando tecnologías web era sólo otro experimento más de programación, sin embargo, hoy en día la construcción de aplicaciones híbridas es una tendencia claramente al alza, principalmente debido a su tiempo y coste de desarrollo más reducido respecto al desarrollo nativo.

Con respecto a la revisión de la literatura científica referente al diseño de GUI para aplicaciones móviles, esta posibilitó desarrollar una primera versión de un prototipo funcional que se construyó, considerando los componentes de AI inherentes a cualquier entorno informacional, en base a los patrones de diseño propuestos por Hoober y Berkman (2011), y también siguiendo las guías del Material Design de Google, esto último considerando que se trata de una aplicación para Android. Respecto a los diferentes apartados del prototipo de aplicación móvil generado, este posee tres pantallas fundamentales: 1) una *home screen* con los diferentes grupos de especies disponibles; 2) una pantalla con un listado de especies según el grupo escogido en la *home screen*, en la que

además se ofrece un buscador que permite agilizar la recuperación de especies del listado; y 3) una ficha informativa con una fotografía e información básica sobre el animal seleccionado desde el listado de especies. Finalmente, cabe señalar que la aplicación cuenta con un sistema de navegación que permite al usuario moverse entre las diferentes pantallas, constituido por un componente de navegación oculta, en concreto un menú desplegable tipo *hamburger menu* y por un componente de navegación fijo, en el que se replican las opciones del menú desplegable en el *footer* de la página.

Finalmente, en relación con la propuesta de ficha de aplicación generada para su distribución en Google Play, cabe recalcar la importancia de generar una estrategia ASO que permita optimizar tanto los factores *On-Metadata* como los factores *Off-Metadata*, con el fin de maximizar la visibilidad de la aplicación.

8.2. Trabajo futuro

Con respecto al prototipo construido, si bien la idea siempre fue desarrollar precisamente eso: un molde que sirva como base para implementaciones similares, luego de realizar este trabajo queda en evidencia la necesidad de generar un proyecto de tecnología móvil que aporte a la conservación de las especies animales de Chile y, por consiguiente, la posibilidad latente de llevar a cabo una futura implementación de esta idea. En ese sentido, es necesario recalcar la importancia fundamental que tendría para el éxito del proyecto, el respaldo de alguna entidad vinculada con la conservación de la fauna chilena como, por ejemplo, los ya mencionados MMA y CONAF. Lo anterior, tanto para obtener los datos de las especies desde una fuente confiable y especializada, que permita evitar imprecisiones, como para sortear inconvenientes vinculados con la autoría tanto de la información textual como de las fotografías necesarias para desarrollar la aplicación.

En relación con un aspecto técnico sobre el prototipo desarrollado, considerando que, mientras más especies se incorporen en la aplicación mayor será su tamaño (MB), parece necesaria la implementación de un mecanismo que permita reemplazar el uso de páginas estáticas en la creación de las fichas informativas de las especies, por una única página dinámica que posibilite rescatar los datos necesarios desde algún fichero de texto. Considerando que se trabajó una aplicación híbrida con JQuery Mobile, JavaScript Object Notation (en inglés, JSON)¹³¹ parece presentarse como una alternativa válida.

Por último, en relación con las funcionalidades de la aplicación, una característica que resultaría interesante de incorporar en una versión futura es una funcionalidad que permita a los usuarios compartir las fichas informativas de las especies en servicios de red social como Facebook o WhatsApp por ejemplo, considerando la relevancia actual de este tipo de estructuras sociales.

¹³¹ JSON corresponde a un formato de texto plano usado para el intercambio de datos. Destaca porque es ligero y sencillo de escribir.

9. Bibliografía

3. Justificación

Chile (2009). “Ley 20380. Sobre protección de animales”. *Diario oficial de la República de Chile*. 03 octubre 2009, nº 6521-12. p 1-2. <<https://www.leychile.cl/Navegar?idNorma=1006858>>. [Consulta: 12 octubre 2017].

Ditrendia (2017). “Informe ditrendia: Mobile en España y en el Mundo 2017”. *Blog de marketing para entidades financiera y aseguradoras*. <<http://mktefa.ditrendia.es/hubfs/Ditrendia-Informe%20Mobile%20en%20Espa%C3%B1a%20y%20en%20el%20Mundo%202017.pdf>>. [Consulta: 11 octubre 2017].

IMS Corporate; comScore (2016). *IMS mobile in Latam study*. 2nd ed. <<https://www.ims corporate.com/news/Estudios-comScore/IMS-Mobile-Study-Septiembre2016.pdf>>. [Consulta: 11 octubre 2017].

4. Marco teórico

Apple (2017). “Cocoa Core Competencies”. En: *Guides and sample code*. <<https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/Cocoa.html>>. [Consulta: 4 noviembre 2017].

Antutu (2017). “Global Top 10 Best Performance Smartphones, August 2017”. *Antutu benchmark*. <<http://www.antutu.com/en/doc/111070.htm>>. [Consulta: 30 octubre 2017].

Ballard, Barbara (2007). *Designing the mobile user experience*. Chichester: John Wiley & Sons.

Budiu, Raluca (2013). “Mobile: native apps, web apps, and hybrid apps”. *Nielsen Norman Group. Articles*. <<https://www.nngroup.com/articles/mobile-native-apps/>>. [Consulta: 4 noviembre 2017].

Cass, Stephen (2017). “The 2017 top programming languages”. *IEEE Spectrum*. <<https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>>. [Consulta: 3 noviembre 2017].

Google (2017). *Driving installs and usage with Firebase App Indexing*. <<https://firebase.google.com/docs/app-indexing/partners/case-study-one-pager.pdf?hl=es>>. [Consulta: 30 octubre 2017].

Hégaret, Philippe Le; *et al.* (2004). “¿Qué es el modelo de objetos del documento?”. En: *W3C recommendation DOM3Core-es*. <<https://www.w3.org/2005/03/DOM3Core-es/introduccion.html>>. [Consulta: 15 noviembre 2017].

Kroski, Ellysa (2008). “What is the Mobile Web?”. *Library technology reports*. Vol. 44, no. 5. <<https://journals-ala-org.sire.ub.edu/index.php/ltr/article/view/4898/5909>>. [Consulta: 26 octubre 2017].

Levitas, Danielle (2017). “The App economy forecast: a \$6 trillion market in the making”. *Insights appannie.com*. <<https://www.appannie.com/en/insights/market-data/app-economy-forecast-6-trillion-market-making/>> [Consulta: 12 octubre 2017].

Marcotte, Ethan (2011). *Responsive web design*. New York: A Book Apart.

Mártil, Ignacio (2017). “¿Cómo funcionan las redes inalámbricas de telefonía móvil?”. *Blog public.es*. <<http://blogs.publico.es/ignacio-martil/2017/02/24/como-funcionan-las-redes-inalambricas-de-telefonía-movil/>>. [Consulta: 13 octubre 2017].

Moll, Cameron (2007). *Mobile web design*. Salt Lake City, UT: Lulu Press.

W3C (2007). *Guía breve de web móvil*. <<https://www.w3c.es/Divulgacion/GuiasBreves/WebMovil>>. [Consulta: 26 octubre 2017].

W3C (2008). “One Web”. En: *Mobile web best practices 1.0*. <<https://www.w3.org/TR/mobile-bp/#OneWeb>>. [Consulta: 26 octubre 2017].

W3C (2016). *Web Storage (second edition)*. <<https://www.w3.org/TR/webstorage/>>. [Consulta: 20 noviembre 2017].

W3C (2017). “CSS Tutorial”. En: *W3Schools*. <<https://www.w3schools.com/css/>>. [Consulta: 29 octubre 2017].

W3Schools (2017). “HTML5 Tutorial”. En: *W3Schools*. <<https://www.w3schools.com/html/default.asp>>. [Consulta: 29 octubre 2017].

5. Análisis de aplicaciones similares

Leafsnap (2017). "About Leafsnap". En: *Leafsnap.com*. <<http://leafsnap.com/about/>>. [Consulta: 20 noviembre 2017].

Museum Victoria (2014). "National field guide apps". En: *Museums Victoria*. <<https://museums victoria.com.au/apps/national-field-guide-apps/>>. [Consulta: 20 noviembre 2017].

SEO/BirdLife (2014). "Primera aplicación móvil para observar y conocer sobre el terreno las aves españolas". En: *SEO/BirdLife*. <<https://www.seo.org/2014/02/18/primera-aplicacion-movil-para-observar-y-conocer-sobre-el-terreno-las-aves-espanolas/>>. [Consulta: 20 noviembre 2017].

Siebert, Francisca (2017). "Casa de Bello crea la primera aplicación chilena para la identificación de aves". En: *Universidad de Chile. Noticias*. <<http://www.uchile.cl/noticias/138376/crean-la-primera-aplicacion-chilena-para-la-identificacion-de-aves>>. [Consulta: 20 noviembre 2017].

6.1. Diseño

Budiu, Raluca (2015). "Basic patterns for mobile navigation: a primer". *Nielsen Norman Group. Articles*. <<https://www.nngroup.com/articles/mobile-navigation-patterns/>>. [Consulta: 6 diciembre 2017].

Card, Stuart (1984). "Visual search of computer command menus". En: *Parc.com*. <<http://www2.parc.com/istl/groups/uir/publications/items/UIR-1984-08-Card-Attention-VisualSearch.pdf>>. [Consulta: 06 diciembre 2017].

Charlier, Martin (2015). "Interface and interaction design". En: *O'Really.com*. <<https://www.oreilly.com/ideas/interface-and-interaction-design>>. [Consulta: 06 diciembre 2017].

Google. (2017). *Material design*. <<https://material.io/>>. [Consulta: 08 diciembre 2017].

Hooper, Steven (2017). "Design for fingers, touch and people, part 1". *UX matters*. <<https://www.uxmatters.com/mt/archives/2017/03/design-for-fingers-touch-and-people-part-1.php>>. [Consulta: 01 diciembre 2017].

Hooper, Steven (2017). "Design for fingers, touch and people, part 2". *UX matters*. <<https://www.uxmatters.com/mt/archives/2017/05/design-for-fingers-touch-and-people-part-2.php>>. [Consulta: 01 diciembre 2017].

Hooper, Steven; Berkman, Eric (2011). *Designing mobile interfaces: patterns for interaction design*. 1st ed. Sebastopol: O'Reilly.

Laubheimer, Paige (2016). "Wireflows: a UX deliverable for workflows and apps". *Nielsen Norman Group. Articles*. <<https://www.nngroup.com/articles/wireflows/>>. [Consulta: 10 diciembre 2017].

Morville, Peter (2005). *Ambient findability*. 1st ed. Sebastopol: O'Reilly.

Morville, Peter; Rosenfeld, Louis (2006). *Information architecture*. 3ª ed. Sebastopol: O'Reilly.

Nielsen, Jakob (2012). "Usability 101: Introduction to Usability". *Nielsen Norman Group. Articles*. <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>>. [Consulta: 4 diciembre 2017].

Pérez-Montoro, Mario (2010). *Arquitectura de información en entornos web*. Gijón: Trea.

Pernice, Kara; Budiu, Raluca (2016). "Hamburger menus and hidden navigation hurt UX metrics". *Nielsen Norman Group. Articles*. <<https://www.nngroup.com/articles/hamburger-menus/>>. [Consulta: 6 diciembre 2017].

Rowland, Claire (2015). "What's different about user experience design for the Internet of Things?". En: *O'Really.com*. <<https://www.oreilly.com/learning/whats-different-about-user-experience-design-for-the-internet-of-things>>. [Consulta: 06 diciembre 2017].

Schade, Amy (2015). "Supporting mobile navigation in spite of a hamburger menu". *Nielsen Norman Group. Articles*. <<https://www.nngroup.com/articles/support-mobile-navigation/>>. [Consulta: 06 diciembre 2017].

Villamor, Craig; Willis, Dan; Wroblewski, Luke (2011). *Touch gesture: reference cards*. <<https://static.lukew.com/TouchGestureCards.pdf>>. [Consulta: 10 diciembre 2017].

6.2. Implementación

Chetan K., Jain (2012). *jQuery Mobile cookbook*. 1st ed. Birmingham, UK: Packt Publishing.

Godesign Webpublishing (2017). *nativeDroid2 documentation*. <<http://natedroid.scripter.click/>>. [Consulta: 17 diciembre 2017].

jQuery Mobile (2017). *jQuery Mobile 1.5 API documentation*. <<http://api.jquerymobile.com/>>. [Consulta: 15 diciembre 2017].

Mozilla (2017). *Lista de elementos HTML5*. <https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos>. [Consulta: 20 diciembre 2017].

Mozilla (2017). *Referencia de elementos HTML*. <<https://developer.mozilla.org/es/docs/Web/HTML/Elemento>>. [Consulta: 20 diciembre 2017].

W3C (2017). *HTML 5.1 2nd edition*. <<https://www.w3.org/TR/html51/>>. [Consulta: 15 diciembre 2017].

6.3. Optimización en tiendas de aplicaciones

Google (2017). *Guía de optimización en buscadores (SEO) para principiantes*. <<https://support.google.com/webmasters/answer/7451184?hl=es>>. [Consulta: 25 diciembre 2017].

Google (2011). *Guía para principiantes sobre optimización para motores de búsqueda*. <https://static.googleusercontent.com/media/www.google.com/en//intl/es/webmasters/docs/guia_optimizacion_motores_busqueda.pdf>. [Consulta: 25 diciembre 2017].

PickASO; TheTool; Tribal Worldwide; IAB Spain (2017). *App Store Optimization 2017*. <[http://www.tribalworldwide.es/wp-content/uploads/2017/10/Manual AS0 2017 Tribal Pickaso IAB.pdf](http://www.tribalworldwide.es/wp-content/uploads/2017/10/Manual_AS0_2017_Tribal_Pickaso_IAB.pdf)>. [Consulta: 25 diciembre 2017].