



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Adaptabilidad sobre permisos y hardware de dispositivos móviles para favorecer la recolección de datos en Ciencia Ciudadana

AUTORES: Federico Di Claudio

DIRECTOR: Mg. Alejandra Beatriz Lliteras y Dr. Julián Grigera

CODIRECTOR:

ASESOR PROFESIONAL:

CARRERA: Licenciatura en Sistemas

Resumen

La Ciencia Ciudadana es una forma de colaboración del público en general (voluntarios) en proyectos científicos, muchas veces realizando tareas de recolección de datos. Las aplicaciones móviles juegan hoy un papel importante en esta tarea, debiendo permitir tomar la mayor cantidad de datos posibles sin descuidar la experiencia de usuario. En esta tesina, se presenta, en primera instancia, la extensión de un componente para la creación de widgets adaptables a restricciones de hardware y falta de permisos. Dicha extensión es utilizada para confeccionar una nueva versión de "Resuelvo Explorando, aplicación móvil de recolección de datos, para integrar nuevos widgets adaptables en las tareas de recolección.

Palabras Clave

Ciencia Ciudadana; Recolección de Datos; Adaptabilidad; Aplicaciones Móviles; Permisos; Restricciones de Hardware; Experiencia de Usuario

Conclusiones

Se desarrollo un componente de software para la creación de widgets adaptables ante restricciones de hardware y falta de permisos. Su adopción en la aplicación móvil "Resuelvo Explorando", permitió que esta muestre diferentes alternativas para realizar tareas de recolección de datos, permitiendo a los voluntarios completar siempre dichas tareas y dándole una mejor experiencia de usuario. El caso de estudio planteado, mostró como los potenciales usuarios valoraron que la aplicación se adapte a ellos y no ellos a las aplicaciones.

Trabajos Realizados

Se presenta, inicialmente, un relevamiento exploratorio de artefactos de software de Ciencia Ciudadana y se analiza su comportamiento ante restricciones de hardware y permisos. Luego, se presenta una extensión de una componente de software para la creación de widgets adaptables ante restricciones de hardware y falta de permisos en dispositivos móviles. Dicho componente se utilizó para crear diferentes widgets que luego fueron adoptadas en una nueva versión de la herramienta "Resuelvo Explorando" para diferentes tareas de recolección de datos. Finalmente, se realizaron pruebas con usuarios para comprobar la mejora en la realización de tareas.

Trabajos Futuros

Se propone a futuro analizar la incorporación de nuevos permisos y restricciones de hardware a el componente para la creación de nuevos widgets adaptables. A corto plazo, se buscará realizar más pruebas con potenciales usuarios sobre la aplicación "Resuelvo Explorando". Además, se contemplará la creación de una API para almacenar los datos recolectados y la posibilidad de facilitar la resolución colaborativa.

Fecha de la presentación: Marzo 2021

ÍNDICE

1	Introducción.....	4
1.1	Motivación.....	6
1.2	Investigación propuesta	6
1.3	Desarrollos propuestos	7
1.4	Resultados esperados	8
1.5	Estructura del documento	9
2	Ciencia Ciudadana	11
2.1	Principios de la Ciencia Ciudadana.....	13
2.2	Etapas de aplicación de la Ciencia Ciudadana	14
2.3	Enfoques de proyectos.....	16
2.4	Desafíos en el tratamiento de los datos	18
2.5	Ventajas de la Ciencia Ciudadana	19
3	Trabajo relacionado.....	22
3.1	Trabajo relacionado en adaptación de aplicaciones.....	22
3.1.1	Computación basada en el contexto.....	22
3.1.2	Adaptabilidad de aplicaciones.....	24
3.1.3	Manejo de permisos	25
3.1.4	Conclusión	26
3.2	Relevamiento de artefactos de Ciencia Ciudadana	28
3.2.1	Etapa 1: Relevamiento de artículos.....	28
3.2.2	Etapa 2: Relevamiento de artefactos de software	29
3.2.3	Análisis de los artefactos de software.....	32
3.2.4	Resultados obtenidos	39
4	Propuesta de desarrollo	42
4.1	Arquitectura de la solución	43
4.2	Tecnologías empleadas.....	45
4.3	Componente para la creación de widgets adaptables.....	47
4.3.1	Permisos	47

4.3.2	Restricciones de hardware	50
4.3.3	Uso del componente y ventajas	52
4.4	Widgets adaptables para la recolección de datos	54
4.4.1	Widget adaptable para la obtención de una imagen	55
4.4.2	Widget adaptable para la obtención de una ubicación	57
4.4.3	Widget adaptable para la obtención de un video	60
4.4.4	Widget adaptable para la obtención de una grabación de audio	62
4.5	“Resuelvo Explorando”: Aplicación de recolección de datos para Ciencia Ciudadana	64
4.5.1	Widgets adaptables en “Resuelvo Explorando”	66
4.5.2	Reutilización de componentes en “Resuelvo Explorando”	68
5	Caso de estudio	70
5.1	Estructura del caso de estudio	71
5.2	Prueba con usuarios	81
5.3	Conclusiones obtenidas y Discusión	83
6	Conclusiones y trabajos futuros	85
	Referencias	89
A.	Anexo A – Instructivo para el uso del componente	96
A.1.	Agregar componente a un proyecto	96
A.2.	Uso del componente	97
A.3.	Permisos y restricciones de hardware	99

Agradecimientos

Quiero agradecer a mi familia, por haber estado siempre presente y por haber ayudado y motivado a culminar esta carrera. Gracias principalmente a ellos que todo esto fue posible.

También agradecer a mis amigos, por haberme dado su apoyo y comprensión en todos los ámbitos de la vida desde hace ya mucho tiempo.

Por último y no menos importante, a mis directores, Ale y Juli, porque fueron una ayuda y una guía constante durante el desarrollo de esta tesina.

1 INTRODUCCIÓN

La ciencia frecuentemente requiere que se tomen muestras. Por ejemplo: un biólogo toma muestras del ecosistema que lo rodea, un astrónomo toma fotografías de los cuerpos del universo, un hidrólogo toma muestras de ríos o lagos. Estas muestras pueden tener diferentes características: ser pocas o miles, en un área pequeña o en continentes enteros y durante periodos de tiempo cortos o largos, incluso años o décadas; dichas características dependen del alcance del proyecto en cuestión. Para proyectos a gran escala donde se requiera grandes cantidades de datos en una variedad de ubicaciones y hábitats durante un largo periodo de tiempo, recopilar tales cantidades de muestras puede ser muy costoso en tiempo y dinero si está solo a cargo de los responsables del proyecto (Cohn, 2008; Rowbotham et al., 2019). Por ello, casi cualquier proyecto con dichas características sólo es posible llevarlo adelante por medio de la Ciencia Ciudadana (Silvertown, 2009).

La Ciencia Ciudadana es una forma de colaboración que involucra a miembros del público general en proyectos de investigación científica que abordan problemas del mundo real (Cohn, 2008). Estos miembros del público son llamados normalmente voluntarios, participantes o científicos ciudadanos y participan como asistentes de campo en estudios científicos (Cohn, 2008). Es decir, son estos quienes se encargan de realizar las tareas encomendadas por los científicos a cargo del proyecto. Esta forma de colaboración, además de reducir los costos de la recolección de muestras para lograr avances científicos, tiene otros beneficios, como el aprendizaje de los voluntarios con respecto a la ciencia y la temática del mismo proyecto (Louv & Fitzpatrick, 2012) o el permitir monitorear procesos en amplias escalas geográficas e, incluso, en terrenos privados o de difícil acceso que sería complicado o, incluso, imposible de monitorear de manera tradicional (Dickinson et al., 2010). En la actualidad existen a nivel mundial miles de proyectos activos de Ciencia Ciudadana, que abarcan disciplinas tan diversas como astronomía, biodiversidad, ecología, ciencias ambientales, psicología, turismo etc. (Rowbotham et al., 2019).

Si bien, como se mencionó anteriormente, los voluntarios se encargan de recolectar muestras en la mayoría de los casos, las formas de participación varían entre proyectos. Los responsables del mismo pueden solicitar tareas basadas en reconocimiento, clasificación, resolución de problemas u otros tipos (Wiggins & Crowston, 2011). En el segundo capítulo de este trabajo, se presentará una clasificación de los tipos de proyectos de Ciencia Ciudadana.

Hoy, las nuevas tecnologías y los recursos que esta nos proporciona, como internet y los teléfonos móviles, con sus diversos sensores, nos permiten nuevas y más amplias formas de llevar a cabo proyectos que involucren Ciencia Ciudadana provocando un gran incremento en el uso de esta (Bonney et al., 2009; Silvertown, 2009), además de facilidades para interactuar con los voluntarios y que estos conozcan, se interesen y se unan a los proyectos.

Una de estas tecnologías para Ciencia Ciudadana es la aplicación “Resuelvo Explorando” (Di Claudio et al., 2020), la cual permite la recolección de datos desde un dispositivo móvil. La aplicación es configurable y reutilizable para diferentes proyectos de Ciencia Ciudadana, permitiendo recolectar muestras en diferentes formatos como texto, preguntas de opción múltiple, foto, audio, ubicación geográfica, entre otros.

“Resuelvo Explorando” (Di Claudio et al., 2020), además de permitir la recolección de muestras, considera ciertas restricciones y permisos disponibles en el dispositivo móvil para favorecer la experiencia de usuario y asegurar la toma de las muestras en diferentes escenarios. Lo anterior se logra con la adaptación de la aplicación ante ciertas condiciones, como ser restricciones de hardware del dispositivo en que se ejecuta, como la conectividad o el bajo porcentaje de batería y ante la falta de permisos concedidos, como el acceso a la ubicación mediante GPS o la cámara. De esta forma, se le muestra al voluntario (mediante la aplicación) una u otra forma de recolectar una muestra según el resultado de la evaluación de estas condiciones. Por ejemplo, al tomar una muestra de tipo foto, si el usuario de la aplicación (el voluntario) concede el permiso de acceso a la cámara en su dispositivo, este podrá tomar la foto usando la cámara; en caso contrario, la aplicación “Resuelvo Explorando” preguntará al voluntario si prefiere conceder el permiso de acceso al almacenamiento para recuperar una foto que haya sido tomada con anterioridad; si este permiso tampoco es concedido, una tercera alternativa será brindada al voluntario. Esta tercera alternativa, no requiere de ningún tipo de permiso ni valida ninguna restricción, consiste en mostrar un cuadro de texto donde el voluntario puede describir textualmente la muestra encontrada.

Para la implementación de esta adaptación en la aplicación “Resuelvo Explorando”, se desarrolló un componente de software que facilita la creación de widgets adaptables ante la evaluación de restricciones de hardware, como si el dispositivo móvil posee conectividad y de qué tipo o el porcentaje actual de batería, y ante falta de permisos, como el acceso al almacenamiento o a la ubicación. Además, permite manejar de manera sencilla la solicitud o verificación de estos en el dispositivo móvil.

A través de la implementación de estos widgets adaptables y la adopción de estos en aplicaciones de recolección de datos, se intenta lograr una mejor experiencia de usuario y una mayor cantidad de muestras obtenidas, al permitir a los voluntarios más alternativas y saltar posibles barreras como lo son sus preferencias de privacidad o la falta o rotura de algunos de los sensores del dispositivo, que no le permitirían tomar una muestra.

En esta tesina se presenta una ampliación de la versión de “Resuelvo Explorando” presentada en (Di Claudio et al., 2020). La mencionada extensión contempla una mejora en el código público existente, una extensión de la componente para la creación de nuevos widgets adaptables contemplando tipos de tareas que un no lo consideraban en la versión citada y, además, se propone la adopción de estos nuevos widgets en “Resuelvo Explorando”.

1.1 MOTIVACIÓN

El uso de los teléfonos inteligentes se ha visto incrementado en el último tiempo, hasta el punto que hoy la mayoría de nosotros tiene acceso a uno de ellos. Estos dispositivos cuentan con gran cantidad de sensores que permiten recolectar diferentes datos del mundo real. Como se mencionó anteriormente, es posible que voluntarios recolecten datos mediante estos dispositivos para proyectos de Ciencia Ciudadana utilizando dichos sensores. Se nos presenta así una forma mucho más simple para los voluntarios que los métodos anteriores, donde utilizaban distintos artefactos de propiedad de los organizadores del proyecto, como en (Cohn, 2008), y donde además debían aprender a usarlos. Se identifica así la necesidad de desarrollar un artefacto de software que permita a los voluntarios ver que datos son necesarios y el soporte para obtenerlos y comunicarlos a los encargados del proyecto de Ciencia Ciudadana al cual colabora.

Pero este incremento del uso de teléfonos inteligentes ha hecho de la privacidad una problemática creciente en los últimos tiempos (Andriotis et al., 2017; Li et al., 2015; Wijesekera et al., 2015), por lo no podemos dejar de lado esta cuestión. Cada vez más los usuarios se interesan por los requerimientos que una aplicación obtiene para su funcionamiento.

Por otro lado, no todos los potenciales voluntarios tienen acceso a las más nuevas tecnologías que se requieren para aportar ciertos datos, como por ejemplo el uso de GPS para aportar la ubicación, ya sea porque su teléfono inteligente no lo posee o porque no funciona. Esta limitación puede provocar perder un voluntario en nuestro proyecto o que este no recolecte los datos solicitados. Sin embargo, estos datos podrían aportarse por otros medios, siguiendo con el ejemplo anterior de proporcionar una ubicación, esta podría obtenerse de igual forma mediante una ubicación manual en un mapa en lugar de utilizar el GPS o, simplemente, mediante un texto explicativo.

Finalmente, proporcionar una buena experiencia de usuario es un requerimiento muy importante en las aplicaciones de hoy en día y que está directamente relacionado con los puntos descritos en los dos párrafos anteriores. Un usuario obtendrá una mejor experiencia de usuario si puede elegir qué permisos dar a una aplicación o de qué forma aportar un dato concreto.

1.2 INVESTIGACIÓN PROPUESTA

En este trabajo se propone, por un lado, realizar una investigación sobre la adaptación de artefactos de software tanto a restricciones de hardware propias de los dispositivos como así también a los permisos concedidos por usuarios a dicho artefacto, condiciones que son contempladas en el desarrollo de la aplicación “Resuelvo Explorando” presentada en esta tesina. De esta forma, se pretende analizar de qué manera distintos artículos abordan esta problemática y las distintas estrategias de adaptación de artefactos ante las limitaciones ya mencionadas. Se describirán cada uno de los trabajos relacionados encontrados y se analizarán y tomarán conclusiones sobre los mismos.

Por otro lado, se propone realizar un relevamiento exploratorio de bibliografía en busca de artefactos de software reutilizados o resignificados que hayan sido utilizados por la comunidad de Ciencia Ciudadana para la recolección de datos. Se describirá cada una de ellas y se las analizarán para detectar cómo estos artefactos de software se comportan ante restricciones de hardware, como, por ejemplo, la falta de conectividad a internet o el bajo nivel de batería, y ante la falta de permisos requeridos para sus funciones, como el acceso al GPS ante una tarea que requiere obtener la ubicación del dispositivo. Finalmente se describirán las conclusiones obtenidas del análisis.

1.3 DESARROLLOS PROPUESTOS

Se tomará como base de esta tesina el trabajo presentado en (Di Claudio et al., 2020). En dicho trabajo se presenta un componente para la creación de widgets adaptables, que muestren una u otra interfaz al usuario según las restricciones de hardware del dispositivo, como la conectividad y el tipo de la misma, el porcentaje de batería y la condición de activación del modo ahorro de batería, y la falta de permisos otorgados por el usuario, como el acceso a cámara, ubicación, almacenamiento, etc. Un widget es una pequeña aplicación con interfaz y funcionalidad propia¹ que a su vez puede ser integrado en aplicaciones más grandes.

Dicho componente fue utilizado en la aplicación de recolección de datos para Ciencia Ciudadana “Resuelvo Explorando”, para la implementación de widgets adaptables para la realización de una tarea de toma de fotografía y obtención de la ubicación. La aplicación muestra, ante dichas tareas, diferentes formas de cargar un mismo dato según la evaluación de las restricciones y permisos ya mencionados.

En este trabajo, a raíz de lo ya presentado en (Di Claudio et al., 2020) y del análisis de los relevamientos bibliográficos mencionados anteriormente, se propone una mejora y extensión de los componentes de dicho trabajo.

Inicialmente se realizará una extensión de la componente para la creación de widgets adaptables² para que se incluyan nuevos permisos no contemplados, como el acceso al micrófono, y nuevas restricciones de hardware, como, por ejemplo, la condición del modo ahorro de energía del dispositivo, que pudiesen limitar el uso de una aplicación móvil. A partir de la extensión de dicho componente se presentará la inclusión de esta nueva versión en “Resuelvo Explorando”. Adicionalmente, considerando el código existente de la componente y de los widgets³ se analizará la necesidad de refactoring y mejoras sobre el mismo.

Luego se realizará la implementación de nuevos widgets adaptables para la recolección de datos de tipo video y audio, y para la lectura tanto de actividades como de tareas, mediante el

¹ <https://www.w3.org/TR/widgets/>

² <https://github.com/cientopolis/mutableWidget>

³ <https://github.com/cientopolis/MOLE-mobile-app>

componente desarrollado para la creación de los mismos. Finalmente, los nuevos widgets creados se adoptarán en una evolución de la aplicación móvil “Resuelvo Explorando” para de esta forma presente la adaptación en nuevos escenarios de uso que no fueron contemplados en la anterior versión. Además, se analizarán y realizarán cambios en los widgets ya incluidos en la aplicación, contemplando la nueva versión del componente desarrollado.

Finalmente, con el objetivo de probar la expansión generada, se ampliará el caso de estudio presentado en (Di Claudio et al., 2020), donde se utiliza “Resuelvo Explorando” para el registro de donaciones al Banco de Alimentos de La Plata⁴. Este nuevo caso de prueba incluirá el uso de los nuevos widgets adaptables y se realizarán pruebas con potenciales usuarios para evaluar tanto el correcto funcionamiento de la herramienta como la mejora en la experiencia de usuario de los mismos. Para instanciar dicho caso se utilizará la herramienta web “MoLE” desarrollada con anterioridad (Lliteras et al., 2019).

En resumen, en esta tesina se llevarán a cabo los siguientes desarrollos:

- Se realizará una extensión de la componente para la creación de widgets adaptables para incluir nuevas restricciones de hardware y permisos.
- Se desarrollarán nuevos widgets adaptables, utilizando la nueva versión de la componente para la creación de las mismas, para tomar datos de video y audio y para la lectura de actividades y tareas.
- Se adoptarán los nuevos widgets adaptables en una nueva evolución de la aplicación “Resuelvo Explorando”
- Se expandirá el caso de estudio y se realizarán pruebas con potenciales usuarios para evaluar el correcto funcionamiento y la experiencia de usuario.

1.4 RESULTADOS ESPERADOS

Se espera obtener, por un lado, una ampliación del componente de software presentado en (Di Claudio et al., 2020) para la creación de widgets adaptables, contemplando nuevas restricciones de hardware y nuevos permisos que puedan afectar el comportamiento de una aplicación.

Por otro lado, al finalizar este trabajo se obtendrá una ampliación de la aplicación “Resuelvo Explorando” presentada en (Di Claudio et al., 2020) para la recolección de datos pensada en el ámbito de la Ciencia Ciudadana, para que disponga de nuevos widgets adaptables empleados para casos de uso que no fueron contemplados en su primera versión. Adicionalmente se obtendrá una mejora de los widgets adaptables ya presentados empleando la nueva versión del componente, además de mejoras en su código.

Finalmente, y mediante la herramienta de instanciación de proyectos “MoLE” (Lliteras et al., 2019), se instanciará un nuevo caso de estudio, extensión del propuesto con anterioridad en

⁴ <https://bancoalimentario.org.ar/>

(Di Claudio et al., 2020). Se obtendrá mediante este caso los resultados de una prueba de conceptos, usando la misma para comprender los resultados finales del desarrollo realizado y comprender los beneficios al usuario tanto en la utilización de la aplicación como en la experiencia de usuario en la misma.

1.5 ESTRUCTURA DEL DOCUMENTO

A continuación, se realiza una breve descripción de cada uno de los siguientes capítulos que componen esta tesina.

En el Capítulo 2, *Ciencia Ciudadana*, se desarrollan los conceptos generales de la Ciencia Ciudadana, recalando su importancia en los proyectos de ciencia actuales y los aspectos concretos que motivaron el desarrollo de este trabajo. Se describe el rol e importancia de los voluntarios en los proyectos, así como los diferentes tipos de proyectos de Ciencia Ciudadana que existen, haciendo especial énfasis en los de recolección de datos, objetivo principal de este trabajo. Por último, se detallan los beneficios de emplear Ciencia Ciudadana, así como sus limitaciones y dificultades.

En el Capítulo 3, *Trabajos Relacionados*, se describe inicialmente el estado del arte de la adaptación en artefactos de software, enumerando diversos trabajos que refieren a este tema y obteniendo conclusiones sobre los mismos. Por otro lado, se describe el relevamiento exploratorio de artefactos de software de Ciencia Ciudadana realizado, detallando las diferentes etapas del relevamiento, los artefactos que fueron resultado del relevamiento y, mediante el análisis propuesto sobre los mismos, se desarrolla la conclusión obtenida.

En el Capítulo 4, *Propuesta de Desarrollo*, se describe inicialmente la arquitectura de la solución propuesta del ecosistema que conforma las herramientas de Ciencia Ciudadana “MoLE” y “Resuelvo Explorando”. Luego, se describe el desarrollo y funcionamiento del componente para la creación de widgets adaptables desarrollado, así como también las ventajas de su implementación. Después, se describen cada uno de los widgets adaptables desarrollados en el marco de esta tesina, describiendo las decisiones tomadas en la implementación de cada uno. Por último, se describe la aplicación de recolección de datos “Resuelvo Explorando” y la incorporación de estos widgets adaptables a la misma.

En el Capítulo 5, *Caso de Estudio*, se detalla la identificación e instanciación del caso de estudio propuesto y la prueba de conceptos realizada. Por último, se muestran y discuten los resultados obtenidos en dicha prueba.

En el Capítulo 6, *Conclusiones y Trabajos Futuros*, se presentan las conclusiones obtenidas con respecto al componente para la creación de widgets adaptables y a la expansión obtenida de la aplicación móvil “Resuelvo Explorando” que incorpora dicho componente. Por otro lado, se proponen algunas ideas sobre trabajos futuros para los desarrollos realizados.

Finalmente, el *Anexo A*, se trata de un instructivo para el uso del componente para la creación de widgets adaptables presentado en este el trabajo, el cual se encuentra libre para el uso de quien lo necesite.

2 CIENCIA CIUDADANA

La Ciencia Ciudadana se define como una forma de colaboración en investigación científica que involucra miembros del público, sin conocimiento científico, en proyectos de investigación científica que aborden problemas del mundo real (Cohn, 2008). Este modo de colaboración entre científicos y el público no se trata de algo nacido en los últimos tiempos, ha sido utilizado durante mucho tiempo como una forma sencilla de recopilar datos e información relevante para científicos, responsables políticos, etc. y contribuir así a los proyectos en los que estos estén involucrados (Kobori et al., 2016).

Los miembros del público que participan en estos proyectos son normalmente llamados participantes, voluntarios o científicos ciudadanos y trabajan, generalmente, como asistentes de campo realizando las tareas encomendadas por los científicos a cargo del proyecto (Cohn, 2008).

Estos proyectos se tratan, generalmente, de estudios a gran escala de patrones, es decir, se requiere que se recopilen grandes cantidades de datos en una gran variedad de ubicaciones y hábitats y durante largos periodos de tiempo, esto puede ser años o incluso décadas (Bonney et al., 2009). Casi cualquier proyecto con dichas características sólo puede ser exitoso con la ayuda de la Ciencia Ciudadana (Silvertown, 2009), ya que, con la ayuda de voluntarios, podemos monitorear procesos en amplias escalas geográficas e, incluso, en terrenos privados que serían difíciles de monitorear de manera tradicional (Dickinson et al., 2010). Además, al incluir un mayor número de voluntarios, obtendremos los mismos resultados en un menor tiempo. Obtenemos todas estas ventajas en contraposición con los métodos tradicionales, donde los pocos científicos se encargan de completar sus tareas, lo que puede resultar, dependiendo del alcance del proyecto, que este sea inviable con respecto a tiempos o costos.

Esta metodología logra obtener una gran variedad de resultados que incluye, además de avances en la investigación científica, aportes al aprendizaje de los voluntarios, el seguimiento del comportamiento ambiental, entre otros, que varían según el énfasis del proyecto (Louv & Fitzpatrick, 2012).

Por otro lado, los voluntarios o participantes normalmente no realizan tareas de análisis de datos o escriben artículos específicos (aunque más adelante detallaremos algunos casos donde esto ocurre), pero son esenciales para reunir la información en la que estos se basan (Cohn, 2008).

Mencionaremos algunos motivos específicos por lo que reclutar voluntarios resulta un recurso de gran utilidad. El primero es que permiten recolectar grandes cantidades de datos en un menor tiempo y en áreas más grandes sin demasiadas dificultades (Cohn, 2008), solo basta con reclutar a la cantidad que se considere necesarios y de las locaciones adecuadas. Otro motivo importante es que permite a personas fuera del ámbito de la ciencia auténtica participación en proyectos de investigación lo que puede despertar el interés, la motivación y el aprendizaje de

estos (Kobori et al., 2016). Además, los voluntarios podrían realizar aportes valiosos de su área de interés (Pescott et al., 2015; Silvertown, 2009). Por último, y posiblemente el principal motivo, es el dinero, llevar a cabo proyectos de investigación de gran envergadura requiere una inversión muy grande que muchas veces hace que no sea posible llevarlo a cabo; sin embargo, dicha inversión puede ser evitada en gran medida mediante esta metodología. Los voluntarios en muchos casos son una fuente de trabajo gratuita para los científicos en términos de dinero, aportando habilidades, poder computacional, por ejemplo (Rowbotham et al., 2019). A cambio, obtienen una experiencia enriquecedora.

Las formas de participación de estos voluntarios varían entre proyectos. Aunque siempre son dadas por los responsables del mismo pueden tratarse de contribuir con datos de acuerdo a un protocolo establecido o con el resultado de tareas basadas en reconocimiento, clasificación o resolución de problemas u otros (Wiggins & Crowston, 2011). Para ello, quienes idean los proyectos de investigación deben escribir protocolos teniendo en cuenta los voluntarios y cuidando no cometer el error de pedirle a estos que recopilen datos que sean demasiado complejos o detallados (Cohn, 2008).

En muchos casos, los científicos además requieren que los voluntarios usen equipamiento y técnicas sofisticadas para realizar sus tareas, como por ejemplo para monitorear la calidad del aire (Wing et al., 2008), documentar cómo las plantas crecen, florecen y mueren u observar cómo los pájaros y otros animales se comportan en un área (Cohn, 2008). Es importante instruir a los voluntarios para estos requerimientos, teniendo en cuenta que muchas veces no poseen los conocimientos necesarios en ciencia o en el manejo de dichas herramientas.

Una vez que los voluntarios completan sus tareas, presentan los resultados obtenidos, generalmente mediante formularios, imágenes, videos, etc. a los investigadores a cargo del proyecto. Dichos datos necesitan ser registrados para verificar su correctitud y calidad, dado que los voluntarios no son expertos en el área, podrían cometer errores. En la cuarta sección del segundo capítulo se ampliará sobre estas cuestiones.

Mencionamos ya que la Ciencia Ciudadana no se trata de una práctica nueva. Los primeros registros de participación pública en proyectos de ciencia datan de la década de 1880, cuando un grupo de fareros recolectaron datos de aves que surcaban la zona aledaña a sus respectivos faros (Droege, 2007). Sin embargo, es considerado como el primer proyecto de Ciencia Ciudadana como tal, dado sus características más cercanas a la definición dada, al conteo anual de aves navideñas de la National Audubon Society⁵ que comenzó en 1900 y se sigue realizando cada año (Cohn, 2008). En la actualidad, existen a nivel mundial miles de proyectos activos que recurren a la Ciencia Ciudadana y estos abarcan disciplinas tan diversas como la astronomía, biodiversidad, ecología y ciencias ambientales, psicología, turismo, etc. (Rowbotham et al., 2019).

⁵ <https://www.audubon.org/es>

Este incremento en el uso de la Ciencia Ciudadana como metodología para llevar adelante proyectos de ciencia, es en parte una consecuencia directa de los nuevos avances en tecnología. Las nuevas herramientas como internet y los dispositivos móviles (smartphones, tablets, etc.) y el acceso a herramientas de abiertas como Google Maps⁶ dieron un impulso importante al uso de esta práctica (Bonney et al., 2009; Silvertown, 2009). Esto debido a que recopilar datos mediante teléfonos inteligentes tiene sus ventajas, es más rápido, fácil, preciso y está al alcance de casi cualquier persona (Vercayie & Herremans, 2014), lo que también ha permitido una participación más amplia. El observador, no necesita copiar las observaciones en su cuaderno de campo, como se hacía antiguamente, o, ni siquiera traspasarlas al sitio web (Vercayie & Herremans, 2014). Los datos recopilados con el mismo teléfono inteligente a través de sus múltiples sensores (cámara, GPS, micrófonos, etc.) se envían directamente desde el mismo con la utilización de un software adecuado.

Estos nuevos métodos han llevado a la Ciencia Ciudadana a un nuevo nivel más avanzado. Se puede ver estos avances en, por ejemplo, el monitoreo de accidentes de tránsito (Vercayie & Herremans, 2014), el seguimiento de enfermedades infecciosas (Hines & Sibbald, 2015), la recolección de imágenes de flores (Boho et al., 2020), así como también otras diversas áreas. Incluso, algunas plataformas como iNaturalist (Horn et al., 2018), pretenden recolectar muestras con el propósito de que futuros proyectos las utilicen.

Sin embargo, hoy, muchos de los artefactos de software que se desarrollan para mediar con los voluntarios son creados específicamente para la recolección de datos de un proyecto en específico, por lo que existe un faltante de aplicaciones configurables a nuestro proyecto que nos permitan recolectar los datos necesarios para estos.

2.1 PRINCIPIOS DE LA CIENCIA CIUDADANA

Con el fin de entender mejor la Ciencia Ciudadana e impulsar nuevos proyectos, la European Citizen Science Association (ECSA)⁷, organismo encargado de estudiar y fomentar la Ciencia Ciudadana en Europa, expuso diez principios, que luego se reconocerían a nivel internacional, considerados como buenas prácticas en la elaboración de proyectos (Robinson et al., 2018):

1. Los proyectos de Ciencia Ciudadana involucran activamente a los ciudadanos en tareas científicas generando nuevo conocimiento y/o una mayor comprensión.
2. Los proyectos de Ciencia Ciudadana producen un resultado científico. Puede ser respondiendo una pregunta de investigación o informando de acciones de conservación, decisiones de gestión o en política ambiental.

⁶ <https://www.google.com/maps>

⁷ ecsacitizen-science.net/

3. Tanto los científicos profesionales como los voluntarios se benefician de la participación. Esto incluye desde los resultados de la investigación hasta el disfrute y la satisfacción personal.
4. Los voluntarios pueden, si lo desean, participar en múltiples etapas del proceso científico. Esto se explicará con mayor profundidad en la siguiente sección.
5. Los participantes deben recibir información del proyecto en todo momento. Esto incluye cómo se utilizan sus datos y que resultados de investigación, políticos o sociales se esperan obtener.
6. La Ciencia Ciudadana representa un tipo de investigación como cualquier otro, con sus limitaciones y sesgos que hay que considerar y controlar. Se ampliará sobre esto más adelante.
7. Los datos y metadatos de proyectos de Ciencia Ciudadana deberían ser públicos y, a ser posible, los resultados deberían publicarse en un formato de acceso abierto. Este intercambio de datos puede darse durante o después de finalizado el proyecto, a menos que haya problemas de seguridad o privacidad que lo impidan.
8. Los voluntarios deben estar reconocidos en los resultados y publicaciones del proyecto.
9. Los programas de Ciencia Ciudadana deben evaluarse por su producción científica, la calidad de los datos, la experiencia de los participantes y el alcance del impacto social o político.
10. Los líderes de proyectos de Ciencia Ciudadana deben tener en cuenta tanto los aspectos legales y éticos como los derechos del autor, la propiedad intelectual, los acuerdos de intercambio de datos, la confidencialidad, la atribución y el impacto ambiental de sus actividades.

2.2 ETAPAS DE APLICACIÓN DE LA CIENCIA CIUDADANA

Si bien mencionamos a la realización de tareas de campo en cualquiera de sus formas (toma de muestras, reconocimiento, clasificación, etc.) como la principal tarea de los voluntarios que participan en un proyecto de Ciencia Ciudadana, esto no siempre es necesariamente así. Como bien se nos indica en (Robinson et al., 2018), uno de los diez principios ya numerados destaca que los voluntarios pueden participar en cualquier etapa del proceso científico, lo que abre nuevas posibilidades para ellos.

En (Rowbotham et al., 2019) se realiza una clasificación de los proyectos de Ciencia Ciudadana en términos de tres modelos de compromiso de los voluntarios con el desarrollo del proyecto: de contribución, colaborativo y co-creado. Dichos modelos se detallan con más precisión a continuación.

En el modelo de recolección, el cual se trata del modelo que más se adecua a la definición de Ciencia Ciudadana dada, así como también es el más usado, los científicos definen las preguntas de investigación y los voluntarios contribuyen con los datos. Generalmente requieren de una gran cantidad de participantes y una mínima contribución de cada uno. El proyecto

“eBird” (Sullivan et al., 2009) consiste en la observación y documentación del comportamiento de distintas especies de aves por parte de los voluntarios, lo que hace de este proyecto un claro ejemplo del tipo recolección.

En el modelo colaborativo, los científicos diseñan el proyecto y los voluntarios se involucran en la definición de la pregunta de investigación, el análisis de los datos, la evaluación de los métodos de recolección y la determinación de conclusiones. Es decir, en este modelo, los voluntarios comprender no sólo la recolección, sino que, además, son parte del desarrollo del proyecto y la toma de decisiones dentro de este. Como ejemplo de este modelo de proyecto podemos mencionar AirVisual⁸, que construyó una base de datos que contiene información sobre la calidad del aire y permite monitorearla, y el proyecto descrito en (Hines & Sibbald, 2015), en el que voluntarios llevan adelante una vigilancia del avance de las enfermedades infecciosas en Canadá.

Por último, en el modelo co-creado, voluntarios y científicos, trabajan juntos en todos los pasos del proceso científico. De hecho, en muchos casos, los voluntarios son quienes idean los proyectos y solo piden ayuda en cuestiones específicas a científicos. Un ejemplo de este tipo de proyectos es el descrito en (Wing et al., 2008), donde un conjunto de residentes cercanos a una industria de cerdos, en cooperación con una universidad, realizó un estudio sobre los efectos a la salud que causaba dicha industria. El compromiso de los participantes y el buen uso de la ciencia se vio reflejado en el resultado: una ordenanza para controlar la industria porcina de la región.

Estos últimos dos métodos, donde el voluntario tiene mucha más participación, son comunes en comunidades que desarrollan sus propios proyectos para abordar problemáticas comunes que le conciernen, como el estado del aire, el agua o el suelo, por ejemplo, y dar lugar a nuevas perspectivas, voces y soluciones a responsables que puedan tomar decisiones que afecten la problemática.

Además, cabe mencionar, que estos dos tipos de proyectos, son los que más se adecuan a la definición inicial de Ciencia Ciudadana, descrita en (Irwin, 1995), y donde se plantea al objetivo de la misma como una forma de democratizar la ciencia, que el público general lleve adelante proyectos que sean de su interés o necesidad.

Por otro lado, (Bonney, Cooper, et al., 2016; Bonney, Phillips, et al., 2016) destacan aquellos proyectos de ciencia en donde no existe participación alguna de científicos durante su desarrollo y los llama “proyectos comunitarios”.

Si bien el objetivo de este trabajo consiste en una herramienta de recolección de datos, centrándose más en el modelo de colaboración, no se quiere dejar pasar por alto que una herramienta configurable, abierta y con una gran variedad de tipos de datos posibles a recolectar

⁸ <https://www.iqair.com/about-us>

es necesaria en la etapa de realización de tareas de cualquier modelo de proyecto. Por lo que, si bien nos concentraremos en el modelo de contribución, contempla cualquiera de ellos.

2.3 ENFOQUES DE PROYECTOS

Los proyectos de Ciencia Ciudadana pueden adoptar una variedad de enfoques, tener diferentes objetivos e involucrar muchas disciplinas de la ciencia (Kobori et al., 2016). En (Wiggins & Crowston, 2011) se identifican cinco tipos de proyectos de Ciencia Ciudadana según cual sea el objetivo del mismo: acción, conservación, investigación o recolección, educación y virtual. A continuación, se explica cada uno de ellos.

Los proyectos de tipo acción se caracterizan por emplear un enfoque de participación activa por parte de los participantes, generalmente realizando algún tipo de actividad. Este tipo de proyectos está generalmente relacionado con preocupaciones locales, por lo que, siguiendo con la tipología descrita en la sección anterior, se tratan de proyectos comunitarios, co-creados o participativos en mayor medida. Muchas veces los participantes realizan procedimientos rigurosos, lo que requiere una capacitación adecuada. El uso de tecnologías en este tipo de proyectos no ha aportado grandes beneficios, hoy su implementación es mínima. Podemos mencionar como ejemplo de este tipo el proyecto Sherman's Creek Conservation Association⁹, el cual se formó para proteger un arroyo local y logró organizar monitoreo de cuencas hidrográficas y eventos de limpieza de arroyos, tareas que fueron realizadas por voluntarios.

En los proyectos de conservación, los cuales están principalmente orientados al área ecológica, los voluntarios realizan tareas de conservación de los recursos naturales de un área específica. Emplear voluntarios para este tipo de proyectos, no sólo resulta práctico debido a la gran cantidad de tareas en qué consisten estos proyectos, sino que, además, promueven y divulgan la conciencia sobre el tópico en estos. El uso de tecnologías ha impulsado la divulgación de este tipo de proyectos. Un ejemplo de este tipo de proyectos es el descrito por (Wing et al., 2008), ya mencionado en la sección anterior, donde se realiza un estudio sobre los efectos a la salud que causaba una industria de cerdos de la región.

Los proyectos de investigación o recolección se centran principalmente en la recolección de muestras o datos del entorno siendo este el tipo de proyectos que más se adecua a la definición de Ciencia Ciudadana dada al inicio de este capítulo. Al contrario de los otros tipos ya mencionados, estos proyectos no siempre limitan su área de aplicación a zonas regionales, sino que varían pudiendo alcanzarse proyectos a nivel internacional, lo que conlleva un desafío mayor. A menudo estos proyectos contribuyen al aprendizaje de los voluntarios ya sea por la estructura de las tareas que realizan o por los elementos empleados para la recolección si estos fueran necesarios. Se requieren una gran cantidad de voluntarios, por lo que resulta un desafío reclutarlos, sobre todo si la distribución debe ser grande y equitativa. Las tecnologías en este tipo de proyectos juegan un papel muy importante, el uso de internet y los teléfonos inteligentes hace

⁹ <https://www.facebook.com/ShermansCreekConservationAssociation/>

que la recolección y recopilación de muestras sea una tarea mucho más sencilla. El ampliamente conocido y ya mencionado proyecto “eBird” (Sullivan et al., 2009), en el que voluntarios observan y documentan poblaciones de aves, es un ejemplo claro de un proyecto de investigación o recolección.

Los proyectos orientados a la educación tienen como objetivo primario la educación y divulgación, no estando el énfasis en la realización de tareas como en los otros tipos de proyectos. Se busca generalmente, ampliar su alcance lo mayor posible. Los tópicos van desde temas de carácter regional e incluso concientización y desarrollo de habilidades de investigación científica. La mayoría suele tener organizaciones asociadas que financian y divulgan el proyecto. El uso de la tecnología juega un papel importante, tanto para la divulgación como para cumplir los objetivos educativos. Un ejemplo es The Fossil Finders Investigation¹⁰, que reúne profesores, estudiantes e investigadores de la Institucion Paleontologica en Ithaca, Nueva York¹¹ e implementa recursos curriculares y talleres que se enfocan en la investigación de fósiles, dirigidos a escuelas primarias y secundarias de la región. El formato propuesto, basado en la investigación supervisada, permite a los estudiantes formar y probar sus propias hipótesis, utilizando datos compilados de su propio trabajo y contribuciones de otras aulas.

Por último, en los proyectos de tipo virtual, a diferencia de los otros tipos, no poseen elementos físicos de ningún tipo, todas las actividades están mediadas por las tecnologías. Si bien, muchas veces comparten los objetivos con otros tipos, sus características son lo bastante distintas a las demás como para considerarlo una categoría aparte. Su principal desafío es diseñar correctamente las tareas de participación en línea, de forma que se cumplan los objetivos, así como también mantener la motivación de los voluntarios, por ello, se suele recurrir a técnicas como la implementación de juegos o la gamificación. Los proyectos virtuales proceden, en su mayoría, de los campos de la astronomía o la paleontología. Como ejemplos podemos mencionar los proyectos llevados adelante en plataformas de Ciencia Ciudadana, como Zooniverse¹² (Simpson et al., 2014) o INaturalist¹³ (Horn et al., 2018).

Cabe destacar que estas categorías no son excluyentes entre sí, es decir un proyecto podría tomar características e implementar estrategias de varias categorías a la vez. Así es el caso de, por ejemplo, el mencionado proyecto Sherman's Creek Conservation Association, que además de ser de tipo acción también lo es de tipo conservación y, además, como implemento un programa de educación ambiental, esto también lo hace de tipo educacional.

En este trabajo nos centramos en la implementación de una herramienta de recolección de datos, por lo que los proyectos de investigación o recolección son los más cercanos a su uso, aunque no se quiere dejar de mencionar que es posible incluir características de tipos de

¹⁰ <https://scistarter.org/fossil-finders>

¹¹ <https://www.priweb.org/>

¹² <https://www.zooniverse.org/>

¹³ <https://www.inaturalist.org/>

proyectos como de conservación, acción o, sobre todo, virtual, por medio de una resignificación de las tareas de la aplicación. Esto dependerá de las consignas que el jefe de proyecto establezca sobre cada tarea y que serán designadas a los voluntarios.

2.4 DESAFÍOS EN EL TRATAMIENTO DE LOS DATOS

Uno de los principios ya mencionados por (Robinson et al., 2018) menciona que, al implementar un proyecto de Ciencia Ciudadana, existen limitaciones y sesgos que hay que considerar y controlar. Esto debido a que los voluntarios que participan en estos proyectos son personas no involucradas en la ciencia y que tienen poco o ningún conocimiento al respecto y que normalmente participan por interés personal, como aficionados. Es por esto que los datos provenientes de proyectos de Ciencia Ciudadana son ampliamente discutidos por los expertos, existiendo grupos que incluso no los toman como evidencia científica cierta, ya que a menudo se llevan a cabo sin un protocolo de campo estandarizado y sin un diseño que garantice la representatividad geográfica de los sitios muestreados (Van Strien et al., 2013). Sin embargo, diversos estudios han demostrado que los resultados obtenidos a través de proyectos que involucran Ciencia Ciudadana, con la implementación de ciertos protocolos y un buen diseño previo, son realistas y de calidad, siendo similares a los obtenidos por profesionales (Crall et al., 2011; Schmeller et al., 2009).

Entonces, es cierto que los voluntarios no tienen el mismo nivel de conocimiento ni experiencia que los científicos y podrían cometer errores presentando datos incorrectos, pero para subsanar esto podemos destacar tres puntos importantes a implementar (Bonney et al., 2014) como son: implementar un entrenamiento previo, el ya mencionado uso de protocolos estandarizados y probados, y el análisis posterior de los datos por parte de los científicos. Si estas tres condiciones se aplican apropiadamente, los voluntarios podrán recolectar datos de la misma calidad que sí lo haría un experto (Bonney et al., 2014). A continuación, se detallan con mayor profundidad estos puntos.

El primero de los puntos destacados es el entrenamiento previo de los voluntarios, implementarlo en un proyecto les permite a los participantes conocer la metodología y tomar conocimiento sobre los datos obtener o acciones a realizar y, a la vez obtener una mayor confianza y motivación. De hecho, este punto es considerado como esencial y se incluye en la guía básica para generar un proyecto de Ciencia Ciudadana descrito en (Bonney et al., 2009). En el estudio realizado en (Crall et al., 2011), se pone como ejemplo de la importancia del entrenamiento el uso de GPS, esencial en el proyecto en cuestión, donde solo un bajo porcentaje de los voluntarios tenía conocimientos y experiencia sobre el uso del mismo, sin embargo, luego de la capacitación la mayoría pudo usarlo correctamente. Hoy, cualquier teléfono inteligente posee una multitud de sensores y lo maneja de manera automática, por lo que, siguiendo con el ejemplo del GPS, tener una aplicación que obtenga la ubicación GPS evita el entrenamiento para obtener una ubicación.

La utilización de protocolos estandarizados es el segundo elemento a aplicar para mejorar la calidad de los datos obtenidos. Estos deben ser *testeados* por profesionales y con voluntarios bajo condiciones realistas para determinar su correctitud (Delaney et al., 2008), se busca con esto limitar a los voluntarios en sus decisiones y así obtener datos más concretos y precisos. Por otro lado, (Droege, 2007) propone limitar los formularios a completar y las restricciones sobre los valores de este y, de esta forma, ayudar a un mejor resultado y facilitar el posterior análisis. Estas restricciones pueden ser fácilmente manejadas con limitaciones en la aplicación móvil, algo que tendremos en cuenta en este trabajo.

Por último, y a pesar de la aplicación de los primeros puntos, pueden existir errores en el conjunto de datos obtenido por los voluntarios, sobre todo si el proyecto posee una gran cantidad de participantes. Es por ello que los datos deben ser analizados por los científicos o expertos a cargo antes de utilizarse, eliminando aquellos que resulten dudosos y puedan repercutir en la correctitud del trabajo final. Se deben establecer con anterioridad patrones que permitan identificar y descartar errores de manera eficaz y eficiente, como bien indica el paso siete de la guía descrita en (Bonney et al., 2009). Esta etapa no será contemplada en este trabajo, poniendo el foco de atención solo en la recolección de datos y en mejorar esta al usuario de la aplicación.

Finalmente, cabe destacar que los científicos no deben pensar que por que existan datos erróneos en el conjunto o por que los voluntarios no sean remunerados por su trabajo, el conjunto de datos no sirva o tenga una menor calidad, en (Droege, 2007) se asegura que la información seguirá siendo útil y tendrá validez. De todas formas, no siempre se busca que el conjunto de datos sea exacto, el nivel de precisión dependerá de la pregunta de investigación que se esté examinando (Crall et al., 2011), por lo que se cuenta con un sesgo de error aceptable que nos puede permitir saltarnos algunos de los puntos mencionados con anterioridad.

2.5 VENTAJAS DE LA CIENCIA CIUDADANA

Al implementar Ciencia Ciudadana en un proyecto científico se obtienen múltiples ventajas que van desde la propia posibilidad de realización del proyecto hasta el aprendizaje y experiencia que obtienen los participantes.

Decimos que muchas veces la Ciencia Ciudadana hace posibles proyectos que de otra manera no lo serían ya que el empleo de muchos participantes permite recopilar grandes cantidades de datos en una variedad de ubicaciones y hábitats y durante largos periodos de tiempo (Bonney et al., 2009), incluso podríamos monitorear procesos en amplias escalas geográficas (Dickinson et al., 2010). En otras ocasiones, los voluntarios no son necesarios para cubrir estas características de amplitud, sino que para llegar a lugares que de manera tradicional no sería posible, como el acceso a terrenos privados (Dickinson et al., 2010).

Otra cuestión importante por la que se suele implementar Ciencia Ciudadana es porque los voluntarios generalmente no reciben una remuneración, lo que ayuda a llevar adelante proyectos que no tienen un respaldo económico importante (Rowbotham et al., 2019).

Sin embargo, uno de los mayores potenciales a explotar por la Ciencia Ciudadana es su capacidad educacional. (Irwin, 1995) dio la primera definición de Ciencia Ciudadana y definió como su principal objetivo hacer de esta una forma democrática de la ciencia, es decir, que los ciudadanos conformen un papel importante y tengan voz dentro de un proyecto de ciencia. Para esto es sumamente necesario explotar la capacidad educacional.

En (Bonney, Phillips, et al., 2016) se plantean algunas preguntas para realizarse dentro de un proyecto de ciencia sobre si los voluntarios realmente están aprendiendo. Estas preguntas pueden comprender el concepto de PUS, que engloba la comprensión de la ciencia (y sus actividades) por parte del público (Bonney, Phillips, et al., 2016).

Es sabido que la Ciencia Ciudadana es una gran oportunidad para incrementar el conocimiento del público tanto por la ciencia como por los tópicos del proyecto, pero no es sencillo evaluar este incremento, es decir, responder a las preguntas propuestas en (Bonney, Phillips, et al., 2016). Además, existen muy pocos proyectos que miden los resultados en cuanto al impacto social que generan y en lo que sus voluntarios aprendieron. Esto debido a que muchas veces los investigadores a cargo no están interesados realmente en provocar estos factores ni los voluntarios en tomarlos.

Sin embargo, algunos proyectos que sí han medido, o bien han notado ampliamente, un aprendizaje en los voluntarios, aunque este varía según la etapa de aplicación de Ciencia Ciudadana (Bonney, Phillips, et al., 2016). A continuación, se detallan algunos de estos proyectos.

En (Brossard et al., 2005), proyecto de recolección donde voluntarios registraron la cantidad de pájaros por especie en un área concreta, se observó que los participantes incrementaron su conocimiento sobre biología y aves. Por otro lado, en (Koke et al., 2007), se presenta los resultados de un proyecto de tipo colaborativo, organizado por el Museo de Historia Natural de Florida¹⁴, donde los participantes sin ningún interés en la ciencia, sino más bien en realizar actividades al aire libre, recolectaron información sobre mariposas; más de la mitad de ellos reportó un incremento y sus conocimientos e interés por la ciencia una vez terminada la tarea. Según estos proyectos y la evaluación realizada en (Price & Lee, 2013), los voluntarios demuestran un incremento en el conocimiento de la ciencia mayor en los proyectos colaborativos que en los de solo recolección; no así existe esta diferencia con respecto al conocimiento sobre los tópicos del proyecto.

Por último, en los proyectos co-creados o comunitarios el incremento en los conocimientos es aún mayor. Retomando lo presentado en (Wing et al., 2008), donde una comunidad lleva adelante un estudio sobre los efectos en la salud que causaba una industria de cerdos, se

¹⁴ <https://www.floridamuseum.ufl.edu/>

reconoció en los participantes un incremento en el conocimiento y la comprensión de la salud humana y la calidad del aire, así como también, una mayor conciencia de los problemas y soluciones con respecto a la salud ambiental del área.

Podemos concluir que en algunas ocasiones la ciencia ciudadana conduce al conocimiento del proceso y la naturaleza de la ciencia y sobre las temáticas que un proyecto aborda. La evidencia de esto es algo que está creciendo, así como también la evidencia de que la ciencia ciudadana puede influir en la vida de los participantes de alguna manera (Bonney, Phillips, et al., 2016).

3 TRABAJO RELACIONADO

En este capítulo se presenta, en primera instancia, un conjunto de trabajos relacionados que abordan la adaptabilidad en artefactos de software. Se hará especial énfasis en la adaptación ante restricciones de hardware y/o falta de permisos. Luego, se detalla un relevamiento exploratorio realizado con el fin de obtener un conjunto de artefactos de software que hayan sido reutilizados o resignificados por la comunidad en proyectos de Ciencia Ciudadana y se analizarán en ellos cómo se adaptan ante las mismas restricciones.

3.1 TRABAJO RELACIONADO EN ADAPTACIÓN DE APLICACIONES

En esta sección, se presentan diferentes trabajos relacionados, resultados de una búsqueda de literatura sobre adaptación de artefactos de software (no necesariamente de Ciencia Ciudadana). La búsqueda se realizó haciendo especial énfasis en la adaptación sobre restricciones de hardware de los dispositivos y ante la falta de permisos otorgados por el usuario, objeto de esta tesina.

Los trabajos que se presentan a continuación fueron divididos en tres campos: i) la adaptación dentro del campo de la computación basada en el contexto o “Context-aware Computing”, ii) la adaptación de aplicaciones fuera del ámbito que comprenden los primeros trabajos, y iii) metodologías e implementaciones para saltar barreras que se pueden presentar al no conceder el usuario un permiso de aplicación con el objetivo de mejorar la experiencia de usuario.

Por último, se presenta una conclusión con respecto a los trabajos descritos y en relación al presentado en esta tesina.

3.1.1 Computación basada en el contexto

Actualmente, es deseable que las aplicaciones y los sistemas móviles adapten funcionalidades a su contexto de ejecución. Se entiende por contexto, según (Dey, 2001), a cualquier información que se puede utilizar para conocer la situación de un usuario con respecto su interacción con el sistema. Esta información puede ser, por ejemplo, la ubicación del usuario, su estado de conectividad, el porcentaje de batería restante o, incluso, la orientación del dispositivo. Esta capacidad de software para detectar, reaccionar y adaptarse al contexto se ha denominado comúnmente en la informática computación basada en el contexto o sensible al contexto, más conocida por su nombre en inglés, “context-aware computing” (Schilit & Theimer, 1994).

Estos cambios de contexto pueden afectar la calidad del sistema y, por tanto, la satisfacción del usuario, por lo tanto, las aplicaciones móviles pueden beneficiarse del conocimiento del contexto y adaptarse a dichos cambios para mejorar la experiencia de usuario (Abusair et al., 2017).

Las aplicaciones sensibles al contexto suelen presentar varios métodos y conjuntos de parámetros variables, cada uno de los cuales se activa cuando el sistema se encuentra en un conjunto específico de condiciones (Elmalaki et al., 2015).

Muchos de los trabajos actuales sobre computación basada en el contexto que se pueden encontrar, están dirigidos al campo del internet de las cosas (IoT). La informática sensible al contexto ha demostrado tener un gran éxito en la comprensión de datos obtenidos por sensores y en la creación de funcionalidades adaptables al resultado de estas (Perera et al., 2013). El trabajo presentado en (Perera et al., 2013) analiza la conciencia del contexto desde un panorama de IoT, demostrando su éxito en dicho campo de investigación.

La computación ubicua y la inteligencia ambiental son otras ramas que presentan gran cantidad de trabajos que utilizan conceptos de computación basada en el contexto, ya que en estas disciplinas la correcta interacción con el usuario es un aspecto crítico (Varela et al., 2016). Estos sistemas están fuertemente relacionados con las personas y sus necesidades, proporcionándoles funcionalidades que les ayuden en su vida diaria cualquiera sea el contexto en que se encuentren (Varela et al., 2016).

A continuación, se detallan algunos trabajos relacionados, resultado del relevamiento realizado, más cercanos a lo contemplado en esta tesina.

En (Abusair et al., 2017), se propone un enfoque de aplicaciones sensibles al contexto basadas en servicios. El enfoque propuesto se basa en determinar la mejor adaptación de la aplicación a aplicar bajo un contexto dado por los recursos de software disponibles, es decir, los servicios disponibles, y los recursos de hardware del dispositivo. Para esto se definen situaciones de contexto donde cada una consiste en un conjunto de variables de contexto. Cada una de estas variables especifica el estado del sistema para esa situación de contexto, por ejemplo, el nivel de batería, el estado de conectividad, el permiso de GPS, etc. De esta manera el sistema determina el contexto actual y, por tanto, el servicio a llamar. La disponibilidad del servicio también forma parte del contexto. Se busca con este enfoque tanto la calidad del software como también la satisfacción del usuario, a través del comportamiento basado en el contexto y la disponibilidad constante desde el punto de vista del usuario. El enfoque requiere definir múltiples situaciones de contexto y, por tanto, gran cantidad de servicios.

En (Sboui et al., 2017) se presenta un enfoque dedicado al desarrollo de una familia de interfaces de usuario sensibles al contexto y adaptables en tiempo de ejecución. El contexto es definido por la plataforma de ejecución (el dispositivo físico y las condiciones de hardware de este), el entorno en el que se utiliza la aplicación (por ejemplo, la ubicación del usuario) y las preferencias del usuario. La información contextual es evaluada y se determina, en tiempo de ejecución, para mostrar una customización de una misma interfaz o widget que puede variar en dos aspectos: presentación y función. La presentación se refiere a la personalización de la interfaz o widget como la fuente, el tamaño de fuente, los colores u otros. La función puede variar, esto se logra con un mismo componente que disponga de varios métodos funcionales. Si bien la

interfaz varía en aspectos claves, como son la funcionalidad de sus elementos, se nos presenta como limitación el adaptar los elementos de una misma interfaz a distintas funcionalidades, por lo que mostrar una alternativa totalmente diferente a una misma tarea de usuario resulta difícil.

En (Elmalaki et al., 2015) se presenta CareDroid, un framework para Android que busca facilitar el desarrollo de aplicaciones sensibles al contexto y hacerlas más eficientes. Para esto desacopla la funcionalidad de la aplicación del monitoreo y manejo del contexto. Así, los desarrolladores escriben sus métodos funcionales y los asocian, mediante un archivo XML, a un contexto específico. CareDroid intercepta constantemente los diversos flujos de sensores del dispositivo para determinar el contexto actual del dispositivo móvil en que la aplicación se ejecuta y de esta forma, limita o permite la ejecución de los métodos definidos por el desarrollador. El caso de estudio muestra una cámara que se adapta al contexto de diferentes maneras, por ejemplo, si el dispositivo dispone de poca batería, no permite encender el flash. La definición del archivo de contexto es un trabajo arduo y una desventaja, al tratarse de un XML con diversas propiedades que definir. Si disponemos de una aplicación grande, con mucha funcionalidad y múltiples contextos, este puede volverse inmanejable. Por otro lado, este framework no contempla los permisos otorgados a la aplicación y la adaptación ante la falta de estos.

Por último, en (Milic & Stojanovic, 2017) se presenta un framework para detectar el contexto del usuario en un teléfono móvil, basándose en sensores integrados en el dispositivo (como GPS, acelerómetros, temperatura, detector de movimiento, etc.) y en sus preferencias de usuario para realizar una u otra acción. El objetivo del trabajo es reducir el procesamiento de los servidores remotos. De esta forma, si el dispositivo se encuentra en cierto contexto puede decidir procesar o no los datos localmente. El caso de estudio que se presenta es una aplicación de recopilación de datos médicos que detecta las condiciones de salud del paciente e interactúa con un servidor o no dependiendo la urgencia de este. Si bien la adaptación proporcionada cumple con los objetivos del trabajo, este no contempla posibles decisiones del usuario al utilizar la aplicación, como el otorgamiento o no de los permisos necesarios para por ejemplo la ubicación lo que puede repercutir en la experiencia de usuario.

3.1.2 Adaptabilidad de aplicaciones

Fuera del concepto de computación basada en el contexto fueron encontrados dos trabajos que refieren a aplicaciones adaptables. Cada uno se detalla a continuación.

En (Van Tonder & Wesson, 2008) se identifica como una problemática de los dispositivos móviles el mostrar sistemas basados en mapas, debido al tamaño de las pantallas y a la dificultad de los gestos de navegación. Para mejorar estos se propone un modelo para incorporar interfaces de usuario adaptables a las características individuales de los usuarios y las capacidades del dispositivo, como el tamaño de su pantalla, para mostrar, por ejemplo, más o menos información al instante. También se describe un prototipo de prueba de concepto, basado en este modelo, que demuestra la implementación exitosa del mismo. El sistema adapta la información (según la

ubicación y tiempo), la visualización (procesan los datos de interacción del usuario y almacenan las preferencias del mismo de forma persistente en el modelo de usuario) y los elementos de la interfaz (orden de menús, por ejemplo). No se tienen en cuenta con ningún tipo de adaptación ante restricciones de hardware del dispositivo o falta de permisos.

En (Mirri et al., 2014) se presenta mPASS (Mobile Pervasive Accessibility Social Sensing), un sistema consciente del contexto para proporcionar rutas y mapas peatonales personalizados. El sistema recopila información de diferentes fuentes como bases de datos de crowdsourcing o datos producidos por autores locales y organizaciones de discapacidad y muestra según las preferencias de usuario un camino peatonal u otro. Así, por ejemplo, alguien que posee alguna discapacidad se le mostrará un camino más accesible, mientras que para quien priorice la seguridad será un camino sobre el cual se tenga información de que sea más seguro, mientras quien desee ir más rápido se le mostrará el camino más corto. La aplicación, por otro lado, recopila datos sobre accesibilidad urbana mediante los sensores del dispositivo, aunque marca estos como de menor confiabilidad que los obtenidos mediante fuente más seguras. Así la adaptación de mPASS trabaja en tres dimensiones: la ubicación del usuario, el perfil de preferencias del mismo y la confiabilidad de la ruta que este desee. No contempla ninguna adaptación sobre restricciones de hardware, como el bajo nivel de batería, ni sobre la falta de permisos, por ejemplo, al solicitar la ubicación mediante GPS, algo requerido por la aplicación.

3.1.3 Manejo de permisos

A través del análisis de los trabajos anteriores, se notó que muy pocas de las estrategias mencionadas en ellos, se adaptan ante la falta de permisos otorgados por el usuario, aunque la mayoría si lo hace ante ciertas restricciones de hardware contempladas en esta tesina. Es por ello que se optó por realizar una nueva búsqueda de bibliografía con el fin de obtener trabajos donde se propongan estrategias para, ante la falta de permisos otorgados por el usuario, poder seguir utilizando las funcionalidades de la aplicación y de esta forma mejorar la experiencia de usuario.

La falta de permisos es una problemática de gran importancia a tener en cuenta a la hora de crear aplicaciones adaptables. Hoy los usuarios muestran una mayor preocupación por la privacidad, prefiriendo en muchos casos no otorgar ciertos permisos que solicitan las aplicaciones. Estudios como (Wijesekera et al., 2015) y (Andriotis et al., 2017) demuestran lo anteriormente mencionado. Esto debido a que muchas veces, las aplicaciones solicitan y acceden a recursos innecesariamente y en momentos donde el usuario no tiene conciencia de ello (Wijesekera et al., 2015).

Si el usuario no se siente cómodo con otorgar algún permiso, puede suceder que, o bien no podrá utilizar alguna funcionalidad de la aplicación o, en el peor de los casos, no podrá utilizar la aplicación en su totalidad, llevando a que el usuario decida desinstalar dicha aplicación de su dispositivo (Wijesekera et al., 2015). Mejorar la experiencia de usuario con respecto a este aspecto resulta de interés para esta tesina.

A continuación, se detallan algunos trabajos que proponen modificar el sistema de permisos para mejorar la experiencia de usuario frente a una aplicación a la que el usuario decide no otorgarle permisos que solicita.

En (Olejnuk et al., 2017), se presenta SmarPer (Smart Permissions) un mecanismo alternativo de permisos avanzados que pretende reemplazar la política estática (se conceden y perdura) de los permisos en los sistemas operativos por una dinámica que dependa del contexto (se otorgan y quitan de manera automática). SmartPer aprende sobre los permisos que fue otorgando o no el usuario, mediante técnicas de Machine Learning, y de esta forma predice y toma decisiones en tiempo de ejecución sobre si conceder ciertos permisos a la aplicación. El mecanismo SmarPer tiene como objetivo facilitar el manejo de permisos al usuario y proporcionarle una mejor experiencia a este en el uso de aplicaciones, sin embargo, al imitar las decisiones del usuario, deja de lado la privacidad de este, si el usuario toma malas decisiones, la inteligencia artificial también lo hará.

En (Gasparis et al., 2019) se presenta Fignant, que consiste en un conjunto de librerías para que desarrolladores puedan construir y utilizar capacidades de administración de permisos dinámicos de grano fino. Con permisos de grano fino se pretende que las aplicaciones ejecuten funcionalidades reducidas si el usuario no desea otorgar ciertos permisos completos. Fignant también permite que el desarrollador especifique permisos sensibles al contexto, a través de la definición de diferentes modos funcionales que pueden guardar las decisiones del usuario para imitarlas en otro momento. La declaración de los permisos necesarios es por función por lo que no se ve limitado el acceso a una interfaz sino su uso. El framework solo está disponible para Android, debido a las limitaciones de iOS con respecto al manejo de los permisos. Los autores han logrado con éxito separar Fignant de la versión de Android en la que corre, lo que hace que se puedan desarrollar y correr aplicaciones en cualquier plataforma. Si bien con la implementación de permisos de grano fino se logra una mejor administración de permisos al permitir al usuario una mayor versatilidad, el desarrollador debe crear los diferentes permisos y los modos funcionales lo que repercute en un mayor tiempo de desarrollo y complejidad. Además, al igual que ocurría con SmarPer, imitar las decisiones del usuario puede poner en riesgo su privacidad.

3.1.4 Conclusión

La búsqueda de trabajo relacionado con respecto a técnicas de adaptación ante restricciones de hardware y permisos resultó en diversos trabajos que fueron divididos en tres áreas: computación basada en el contexto, adaptación de aplicaciones y manejo de permisos.

La aplicación “Resuelvo Explorando” presentada en este trabajo es una aplicación sensible al contexto, que en este caso está dado por las condiciones de hardware del dispositivo como la conectividad y el porcentaje de batería restante, y los permisos otorgados a la aplicación, siendo estos los parámetros variables antes mencionados. Ante un contexto dado, “Resuelvo

Explorando” muestra una u otra interfaz que provee una u otra alternativa a una misma funcionalidad.

Los trabajos encontrados sobre Context-aware Computing o computación basada en el contexto, se tratan de enfoques o metodologías propuestas para la creación de aplicaciones, donde la adaptación de estas se basa en el contexto de ejecución. En la mayoría de los casos requiere un esfuerzo mayor por parte del desarrollador para su implementación. Por otro lado, cada trabajo varió con respecto a lo evalúa como contexto, donde la mayoría considera algunas de las restricciones de hardware que se contemplan en esta tesina. Sin embargo, muchos de los trabajos no evalúan dentro de su contexto decisiones de usuario como el otorgamiento o no de permisos de ejecución entre las preferencias de este. Ninguno de ellos tampoco, muestra una forma sencilla de definir interfaces y funcionalidades alternativas ante un contexto desfavorable, sino que lo que hacen es mostrar una u otra información o reducir la funcionalidad de la aplicación.

Los trabajos encontrados de aplicaciones adaptables fuera del campo de la computación basada en el contexto, refieren aplicaciones que se adaptan para mostrar una u otra interfaz, o bien una u otra funcionalidad, siendo que ninguna contempla ambas cosas. Estas aplicaciones consideran pocas restricciones de hardware para adaptarse y, aunque si muestran algún tipo de adaptación ante preferencias de usuario extraídos del perfil del mismo, dejan de lado la consideración del otorgamiento o no permisos a la aplicación.

Por último, y ante la falta de trabajos sobre adaptabilidad que contemplen permisos de usuario a las aplicaciones, se enumeraron algunos trabajos que intentan mejorar la experiencia de aquellos usuarios que no deseen conceder ciertos permisos a las aplicaciones. Ambos trabajos requieren de gran esfuerzo por parte de los desarrolladores y no permiten saltar las barreras como si la logramos saltar en nuestro trabajo, donde ante la falta de un permiso proponemos una alternativa a la función, pero con el mismo objetivo. Encontramos como falencias en los trabajos descritos, la vulneración de la privacidad, al recopilar e imitar decisiones del usuario o el acotamiento de funcionalidad como medio para afrontar la falta de permisos.

Como se pudo observar mediante los citados trabajos relacionados, y como bien se indica en (Soui et al., 2017), la mayoría de los trabajos existentes de adaptación se basan en la definición de reglas declarativas utilizando combinaciones de criterios de contexto (como el perfil de usuario, la ubicación, el dispositivo, etc.) y las características de la interfaz (colores, dimensión, etc.).

De esta forma podemos concluir que ningún trabajo de los citados, permite la instanciación de componentes adaptables a restricciones de hardware y falta de permisos, ni existen aplicaciones que presenten dicha adaptabilidad. En esta tesina se tiene como objetivo que el usuario, sin perder privacidad y pudiendo tomar decisiones con respecto a esta, encuentre toda la funcionalidad de la aplicación disponible mediante opciones alternativas que se presentan de

manera transparente al usuario. No fue posible encontrar trabajos que se correspondan con ese objetivo.

3.2 RELEVAMIENTO DE ARTEFACTOS DE CIENCIA CIUDADANA

En esta sección se presenta un relevamiento exploratorio realizado con el fin de determinar cómo los artefactos de software utilizados para la recolección de datos de distintos proyectos de Ciencia Ciudadana se adaptan antes restricciones de hardware, como la falta de acceso a internet o el bajo nivel de batería, y la falta de permisos, como el acceso a la cámara o al almacenamiento, según la problemática propuesta en este trabajo.

Se busca obtener un conjunto de artefactos de software de recolección de datos de Ciencia Ciudadana, que cumplan ciertos criterios propuestos con el fin de que dichos artefactos cumplan con los objetivos y características de la herramienta “Resuelvo Explorando” presentada en este trabajo.

La metodología empleada para llevar a cabo el relevamiento consistió en dos etapas: en primera instancia se reunió un conjunto de artículos que utilizan artefactos de software para la recolección de datos dentro de un proyecto de Ciencia Ciudadana o que presentan un artefacto para tal fin. En una segunda etapa se analizó los artefactos de software resultantes bajo ciertos criterios de inclusión. Como resultado, se obtuvieron un conjunto de artefactos de software configurables y adaptables a la recolección de datos en futuros proyectos de Ciencia Ciudadana, que cumplen los mismos objetivos que “Resuelvo Explorando” y se analizó en cada uno de estos su comportamiento ante restricciones de hardware y falta de permisos, antes mencionadas.

A continuación, se desarrolla con mayor detalle el relevamiento realizado y los resultados obtenidos.

3.2.1 Etapa 1: Relevamiento de artículos

Esta primera etapa del relevamiento consistió en encontrar artículos informáticos, escritos en inglés o español, que hayan sido publicados hasta octubre de 2020 (fecha en que fue realizado dicho relevamiento) y que presenten o bien una herramienta concreta de recolección de datos de Ciencia Ciudadana o un proyecto de Ciencia Ciudadana en el que se utilice un artefacto de software para el mismo fin. Se optó por no establecer una fecha mínima para la inclusión de artículos con el fin de incluir aquellos que expliquen el desarrollo y funcionamiento de la herramienta, sin importar que sean anteriores siempre que dicha herramienta siga en funcionamiento. Fueron incluidos artículos de conferencias, revistas y workshops y capítulos de libros. Cabe aclarar que no se trata de un estudio sistemático, sino que es exploratorio.

Para la búsqueda de artículos se utilizó el motor de búsqueda Google Scholar¹⁵ y se empleó como principales términos para dicha búsqueda: “Citizen Science” y “Data Collection”. A partir

¹⁵ <https://scholar.google.com/>

de ellos, se consideraron otros términos derivados como “Citizen Science Application”, “Citizen Science Tool”, “Citizen Science Platform”, “Citizen Science Mobile”, “Citizen Science Data”, “Data Collection Tool”, “Data Collection Platform”, “Observations to Science”, “Capturing Research Data”, “Data acquisition”, entre otros.

Se listan a continuación los criterios de inclusión y exclusión considerados para esta primera etapa del relevamiento:

- Se aceptaron artículos que presentan un artefacto de software para la recolección de datos de Ciencia Ciudadana
- Se aceptaron artículos que presentan un proyecto de Ciencia Ciudadana y que utilicen como medio de recolección de datos algún artefacto de software.
- Se aceptaron artículos publicados en cualquier fecha, con el fin de obtener los primeros artículos específicos de presentación de algunos artefactos y así obtener mayor información sobre su funcionamiento.
- Se aceptaron artículos de conferencias, revistas, workshops y capítulos de libros.
- Se excluyeron artículos que utilicen o presenten artefactos de software para otros tipos de tareas de Ciencia Ciudadana, como clasificación o resolución de problemas, es decir, que no permitan la recolección de datos.

Los artículos encontrados a través de la metodología descrita de esta primera etapa del relevamiento, pasaron a una segunda etapa del relevamiento, en la que se analizaron los artefactos que estos presentaron. El proceso de selección fue realizado por el autor de este trabajo y con asesoramiento de los directores.

3.2.2 Etapa 2: Relevamiento de artefactos de software

A partir de los artículos resultantes de la búsqueda realizada en la primera etapa, se procede a esta segunda etapa donde se analizan y aplican criterios de inclusión y exclusión sobre los artefactos de software que se describen en los artículos encontrados.

Además, en esta etapa se realizó una segunda búsqueda de artículos, con los mismos criterios y el mismo motor de búsqueda, por cada artefacto identificado. Se logró recaudar por cada artefacto de software un conjunto de artículos que emplean el mismo para sus objetivos. Así se logró obtener más información sobre el uso del artefacto de software en distintos proyectos de Ciencia Ciudadana.

Los criterios de aceptación fueron planteados para obtener un conjunto de artefactos que posean las mismas características generales y objetivos que la plataforma “Resuelvo Explorando” que se presenta en este trabajo. Como principal punto, el artefacto debe poder ser utilizable en otro proyecto, es decir, debe poder ser reutilizados o resignificado nuevamente para próximos proyectos de Ciencia Ciudadana. Estas prácticas permiten aumentar la productividad y reducir los tiempos para obtener una solución tecnológica, al no tener que desarrollar de cero una nueva herramienta (Walton & Maiden, 2019).

A continuación, se detallan los criterios de inclusión y exclusión aplicados sobre los artefactos de software reunidos:

- El artefacto de software debe permitir realizar la recolección de datos en un proyecto de Ciencia Ciudadana (no se descartan artefactos que no hayan sido diseñados para tal fin, pero puedan ser resignificados)
- El artefacto de software debe poder configurarse y emplearse en otro proyecto de Ciencia Ciudadana, por cualquier comunidad o persona que lleve adelante dicho proyecto, independientemente del dominio del mismo.
- El artefacto de software debe estar disponible para su uso y de manera gratuita (al menos de forma parcial).
- El artefacto de software debe poder ejecutarse desde un dispositivo móvil.

Al finalizar este trabajo obtenemos un conjunto de 14 artefactos de software que cumplen con los criterios establecidos. En la *Tabla 1* se incluyen los artefactos con sus respectivos artículos donde fue presentado y/o utilizado, resultado de ambas etapas del relevamiento realizado:

Tabla 1 – Artefactos de software resultantes del relevamiento realizado

Aplicación	Artículos
Sensr	(1) Kim, S., & Paulos, E. (2012). A subscription-based authoring tool for mobile citizen science campaigns. In CHI'12 Extended Abstracts on Human Factors in Computing Systems (pp. 2135-2140). (2) Kim, S., Mankoff, J., & Paulos, E. (2013, February). Sensr: evaluating a flexible framework for authoring mobile data-collection tools for citizen science. In Proceedings of the 2013 conference on Computer supported cooperative work (pp. 1453-1462).
iNaturalist	(1) Boone, M. E., & Basille, M. (2019). [UW458] Using iNaturalist to contribute your nature observations to science. EDIS, 2019(4), 5-5. (2) Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., ... & Belongie, S. (2018). The inaturalist species classification and detection dataset. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8769-8778).
CitSci.org	(1) Wang, Y., Kaplan, N., Newman, G., & Scarpino, R. (2015). CitSci.org: A new model for managing, documenting, and sharing citizen science data. PLoS Biol, 13(10), e1002280.
Zooniverse	(1) Simpson, R., Page, K. R., & De Roure, D. (2014, April). Zooniverse: observing the world's largest citizen science platform. In Proceedings of the 23rd international conference on world wide web (pp. 1049-1054). (2) Fortson, L., Masters, K., Nichol, R., Edmondson, E. M., Lintott, C., Raddick, J., & Wallin, J. (2012). Galaxy zoo. <i>Advances in machine learning and data mining for astronomy, 2012</i> , 213-236.

Aplicación	Artículos
KoBoToolBox	(1) Chau, M. M. (2020). Rapid Response to a Tree Seed Conservation Challenge in Hawai ‘i Through Crowdsourcing, Citizen Science, and Community Engagement. <i>Journal of Sustainable Forestry</i> , 1-19.
Epicollect	(1) Aanensen, D. M., Huntley, D. M., Feil, E. J., & Spratt, B. G. (2009). EpiCollect: linking smartphones to web applications for epidemiology, ecology and community data collection. <i>PloS one</i> , 4(9), e6968. (2) Heigl, F., & Zaller, J. G. (2014). Using a citizen science approach in higher education: a case study reporting roadkills in Austria. <i>Human Computation</i> , 1(2), 165-175.
ODK	(1) Brunette, W., Sundt, M., Dell, N., Chaudhri, R., Breit, N., & Borriello, G. (2013, February). Open data kit 2.0: expanding and refining information services for developing regions. In <i>Proceedings of the 14th workshop on mobile computing systems and applications</i> (pp. 1-6). (2) Davids, J. C., Rutten, M. M., Pandey, A., Devkota, N., Oyen, W. D. V., Prajapati, R., & Giesen, N. V. D. (2019). Citizen science flow—an assessment of simple streamflow measurement methods. <i>Hydrology and Earth System Sciences</i> , 23(2), 1045-1065.
REDCap	(1) Wright, A. (2016). REDCap: A tool for the electronic capture of research data. <i>Journal of Electronic Resources in Medical Libraries</i> , 13(4), 197-201. (2) Patridge, E. F., & Bardyn, T. P. (2018). Research electronic data capture (REDCap). <i>Journal of the Medical Library Association: JMLA</i> , 106(1), 142. (3) Schmitz, H., Howe, C. L., Armstrong, D. G., & Subbian, V. (2018). Leveraging mobile health applications for biomedical research and citizen science: a scoping review. <i>Journal of the American Medical Informatics Association</i> , 25(12), 1685-1695. (4) Santori, C., Keith, R. J., Whittington, C. M., Thompson, M. B., Van Dyke, J. U., & Spencer, R. J. (2021). Changes in participant behaviour and attitudes are associated with knowledge and skills gained by using a turtle conservation citizen science app. <i>People and Nature</i> .
Twitter	(1) Daume, S. (2016). Mining Twitter to monitor invasive alien species—An analytical framework and sample information topologies. <i>Ecological Informatics</i> , 31, 70-82. (2) Roberts, H. V. (2017). Using Twitter data in urban green space research. <i>Applied Geography</i> , 81, 13-20. (3) Daume, S., & Galaz, V. (2016). “Anyone Know What Species This Is?”—Twitter Conversations as Embryonic Citizen Science Communities. <i>PloS one</i> , 11(3), e0151387 (4) Hart, A. G., Carpenter, W. S., Hlustik-Smith, E., Reed, M., & Goodenough, A. E. (2018). Testing the potential of Twitter mining methods for data acquisition: Evaluating novel opportunities for ecological research in multiple taxa. <i>Methods in Ecology and Evolution</i> , 9(11), 2194-2205.

Aplicación	Artículos
AppInventor	(1) Ellul, C., Gupta, S., Haklay, M. M., & Bryson, K. (2013). A platform for location based app development for citizen science and community mapping. In <i>Progress in Location-Based Services</i> (pp. 71-90). Springer, Berlin, Heidelberg.
Google Forms	(1) Portas, A. M., Barnard, L., Scott, C., & Harrison, R. G. (2016). The National Eclipse Weather Experiment: use and evaluation of a citizen science tool for schools outreach. <i>Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences</i> , 374(2077), 20150223. (2) Hesley, D., Burdeno, D., Drury, C., Schopmeyer, S., & Lirman, D. (2017). Citizen science benefits coral reef restoration activities. <i>Journal for Nature Conservation</i> , 40, 94-99.
Google Sheets	(1) Britton, C. J., Level, A. V., & Gardner, M. A. (2013). Crowdsourcing: Divide the work and share the success. <i>Library Hi Tech News</i> . (2) Berland, A., Roman, L. A., & Vogt, J. (2019). Can Field Crews Telecommute? Varied Data Quality from Citizen Science Tree Inventories Conducted Using Street-Level Imagery. <i>Forests</i> , 10(4), 349.
Google Appsheet	(1) Appenfeller, L. R., Lloyd, S., & Szendrei, Z. (2020). Citizen science improves our understanding of the impact of soil management on wild pollinator abundance in agroecosystems. <i>PloS one</i> , 15(3), e0230007.
Google Docs	(1) Spicer, H., Nadolny, D., & Fraser, E. (2020). Going Squirrelly: Evaluating Educational Outcomes of a Curriculum-aligned Citizen Science Investigation of Non-native Squirrels. <i>Citizen Science: Theory and Practice</i> , 5(1).

3.2.3 Análisis de los artefactos de software

Hasta ahora, se obtuvo un conjunto de artefactos de software que han sido utilizados en proyectos de Ciencia Ciudadana para recolección de datos y que cumplen con características similares a “Resuelvo Explorando”, principalmente en ser configurables y usables en futuros proyectos. Se analizará para cada uno su comportamiento, por un lado, ante las restricciones de hardware previstas en este trabajo como: la falta de conectividad, el tipo de conectividad y el nivel de batería; y, por otro lado, ante la falta de permisos otorgables por el usuario al artefacto. Estos últimos son equivalentes en los sistemas operativos más usados, Android y iOS, y consisten en el acceso a recursos o sensores del dispositivo, como la ubicación mediante GPS, el acceso a la cámara, el micrófono, el almacenamiento interno (ya sea para leer o escribir), etc., que pueden poner en peligro la privacidad del usuario.

Para el análisis se leyeron de manera completa los artículos correspondientes a cada artefacto de software (listado en la *Tabla 1*), así como también se leyó la documentación en el sitio web de cada herramienta, en el caso de estar disponible. Además, se probó cada artefacto instanciando un proyecto sencillo de prueba en cada uno.

A continuación, se detalla cada una de los catorce artefactos de software producto del relevamiento realizado.

Sensr¹⁶, presentada en (Kim et al., 2013), se trata de una herramienta de creación que permite a potenciales usuarios no programadores instanciar aplicaciones de recopilación de datos. Aprovecha el hecho de que la mayoría de los proyectos de Ciencia Ciudadana tienen tareas similares sin importar el dominio. Posee, por un lado, un entorno virtual web en el cual los administradores del proyecto pueden crear una interfaz para un dispositivo móvil que consista en las tareas a realizar por el voluntario, llamando a esta “campana”. Por otro lado, posee una aplicación móvil que debe ser descargada por el voluntario y desde allí puede realizarse la carga de datos. Se limita a datos de tipo ubicación, foto, texto y preguntas de opción múltiple. Solo está disponible para dispositivos iOS, lo que restringe en gran medida la cantidad de potenciales voluntarios, ya que Android es el sistema operativo más utilizado en la actualidad. La única adaptación que presenta es ante la falta de conexión a internet, posibilitando retrasar la carga de datos hasta el momento en que se vuelve a obtener conexión. Ante la falta de permisos, el voluntario no puede realizar las tareas, por lo que Sensr no dispone de ninguna adaptación de este tipo.

INaturalist¹⁷, presentada en (Horn et al., 2018) y (Boone & Basille, 2019), es una plataforma donde voluntarios pueden subir “observaciones”, es decir, avistamientos de animales, insectos y plantas, en forma de foto, audio o texto y adjuntando una ubicación a la misma. Los datos pueden pertenecer a proyectos creados en la plataforma, lo que permite definir un formulario de datos necesarios para este para una observación de este, o los voluntarios pueden subir observaciones que realicen sin un fin destinado. Luego, estos datos pueden obtenerse de forma pública por quien lo desee para su proyecto de Ciencia Ciudadana. Las observaciones abarcan el dominio de los seres vivos, esto es, plantas, animales e insectos, por lo que se ve limitado el dominio de los proyectos a llevar adelante con este artefacto. Por otro lado, el artefacto utiliza técnicas de inteligencia artificial para la fácil identificación del objeto de observación, ayudando al voluntario en dicha tarea. INaturalist está disponible en aplicaciones web, Android y iOS. Permite, en una observación libre, la de carga de una observación a través de una foto, audio o solo una descripción, por lo que presenta allí la adaptación que se busca en cuanto a permisos. Además, permite retrasar la carga de datos hasta que se disponga de conexión a internet. Sin embargo, no presenta adaptación en cuanto al tipo de conexión o el nivel de batería, además, la adaptación en cuanto a permisos mencionada sólo es posible al realizar observaciones libres, dejando de lado la carga de datos a proyectos.

CitSci.org¹⁸, presentada en (Wang et al., 2015), es una plataforma que permite la creación y administración de proyectos de Ciencia Ciudadana de libre acceso sin limitar el dominio del mismo. Posee una interfaz web como también una aplicación para Android y iOS. Permite la carga de datos desde la web o la aplicación móvil y en distintos formatos configurables al proyecto: ubicación, descripciones en forma de texto, preguntas de opción múltiple y foto. Permite la

¹⁶ <http://www.sensr.org/>

¹⁷ <https://www.inaturalist.org/>

¹⁸ <https://www.citsci.org/>

interacción entre voluntarios a través del foro propio de cada proyecto y, además, permite desarrollar una wiki propia al este con información relevante para el voluntario sobre el proyecto. No presenta ningún tipo de adaptación ante las restricciones de hardware o falta de permisos, problemática contemplada en este trabajo.

Zooniverse¹⁹, presentada en (Simpson et al., 2014), es un portal de Ciencia Ciudadana para la administración de proyectos de cualquier dominio. Se trata de una de las primeras plataformas de Ciencia Ciudadana, por lo que resulta una de las más conocidas y ha llevado adelante múltiples proyectos de distintos dominios, entre ellos podemos destacar entre los más conocidos Galaxy Zoo (Fortson et al., 2012). Los voluntarios pueden unirse a un determinado proyecto y realizar tareas tanto de recolección de datos como de clasificación o identificación. Está disponible tanto en una interfaz web como en forma de aplicaciones para Android y iOS. Implementa foros de discusión en los proyectos y la posibilidad de mostrar guías de cómo tomar las muestras a los voluntarios que se hayan unido. Los datos pueden recolectarse en forma de imágenes, video, audio y ubicación geográfica. Zooniverse no presenta adaptación alguna ante restricciones de hardware o falta de permisos, de hecho, ni siquiera permite su uso de forma online, siendo uno de los más limitados en este aspecto.

KoboToolBox²⁰, es un artefacto de software que permite la recolección de datos, aunque que no ha sido específicamente diseñada para la administración de proyectos, sin embargo, como resultado del relevamiento se encontraron proyectos como (Chau, 2020) que han utilizado este artefacto, por lo que es posible una resignificación del mismo dentro del ámbito de la Ciencia Ciudadana. KoboToolBox posee una interfaz web donde el creador del proyecto diseña un formulario con los datos a recolectar, mientras que el voluntario desde su aplicación móvil llamada KoboCollect, puede descargarlo y completarlo. Esta aplicación móvil está disponible para Android. KoboToolBox dispone de soporte para muchos tipos de datos entre los que podemos encontrar foto, video, audio, texto y ubicación. En lo referente a la adaptación, la aplicación móvil KoboCollect permite retrasar la carga de datos si no se posee conexión a internet al momento de completar el formulario. Por otro lado, frente a la falta de permiso para acceder a la ubicación mediante GPS, la aplicación muestra como alternativa la posibilidad de dar una ubicación de forma manual.

EpiCollect²¹, es también otro artefacto de software de recolección de datos que ha sido resignificado para su uso en Ciencia Ciudadana. En el relevamiento se encontraron proyectos como (Aanensen et al., 2009) y (Heigl & Zaller, 2014), que son ejemplos de esta resignificación. EpiCollect permite la creación de proyectos de recolección donde los usuarios pueden unirse y aportar datos a formularios creados dentro de este. A diferencia de otros artefactos de software, EpiCollect, en sus últimas versiones, permite la creación de varios formularios dentro de un

¹⁹ <https://www.zooniverse.org/>

²⁰ <https://www.kobotoolbox.org/>

²¹ <https://five.epicollect.net/>

mismo proyecto, lo que permite una mayor flexibilidad. Si bien su interfaz para crear proyectos y formularios se encuentra solo en formato web, las entradas de datos son posibles tanto desde la misma aplicación web como en sus respectivas para Android y iOS. Los formularios consisten en múltiples entradas de diferentes tipos de datos entre los que se encuentran foto, video, audio, texto, ubicación, preguntas de opción múltiple, entre otros. Además de permitir su uso online en las aplicaciones móviles, permite retrasar la carga de datos si no se tiene conexión en estas aplicaciones. Este artefacto no presenta ninguna adaptación ante la falta de permisos.

ODK (Open Data Kit)²², presentada en (Brunette et al., 2013), es un artefacto de software que consiste en una suite de aplicaciones que nos permite, desde una interfaz web la creación de aplicaciones Android con cualquier fin. Como se muestra en (Davids et al., 2019), trabajo encontrado en el relevamiento realizado, ODK puede ser empleado para la creación de aplicaciones de recolección de datos de Ciencia Ciudadana, aunque en muchos casos requiere la colaboración de un programador con conocimiento para añadir características más complejas. Una vez creada una aplicación, esta se debe colocar al alcance de los voluntarios para que estos puedan obtenerla y participar en el proyecto. Al ser completamente configurable podemos destinar los datos cargados por los voluntarios a una base de datos de nuestra elección. Permite configurar la carga de datos en formato de tipo foto, ubicación, texto o podemos obtener la información de otros sensores del dispositivo, como el acelerómetro, aunque esto ya requiere mayores conocimientos de programación. Presenta la posibilidad de configurar que la carga de datos se retrase ante la falta de conexión, pero no presenta la posibilidad de crear, al menos de forma predeterminada, widgets de carga de datos que sean adaptables.

REDCap²³, presentada en (Wright, 2016) y (Patridge & Bardyn, 2018), es un artefacto de software originalmente pensada para el ámbito de la salud, sin embargo, como se menciona en (Schmitz et al., 2018) puede ser utilizada en el ámbito de la Ciencia Ciudadana para la recolección de datos, proyectos como (Santori et al., 2021) lo utilizan como solución. Es similar a la ya mencionada KoboToolBox, permitiendo la recolección de datos a través de la creación de formularios desde su interfaz web y pudiendo descargarse luego en su respectiva aplicación móvil, también sólo disponible en Android, para ser completadas por los voluntarios. Permite la recolección en algunos tipos de datos configurables, como texto, preguntas de opción múltiple, firma digital o da la posibilidad de subir archivos sin verificar el tipo concreto del mismo, lo que nos permite obtener archivos de cualquier tipo. Si bien permite que la carga de datos sin conexión, luego de completo el formulario, se retrase hasta obtener de nuevo conexión, no tiene en cuenta ninguna otra restricción de hardware del dispositivo ni sabe adaptarse ante la falta de permisos.

²² <https://opendatakit.org/>

²³ <https://www.project-redcap.org/>

Twitter²⁴, se trata de una red social mundialmente conocida, donde los usuarios publican textos cortos, opcionalmente acompañados de imágenes o un video y una ubicación. Se ha encontrado en el relevamiento que Twitter ha sido resignificada para su utilización en proyectos de Ciencia Ciudadana en distintas oportunidades y de diversas maneras. En casos como (Daume, 2016), (Daume & Galaz, 2016) y (Roberts, 2017), los responsables de proyectos han realizado una búsqueda de datos, mediante la API abierta de la que dispone Twitter, utilizando palabras clave o hashtags (etiquetas) que puedan contener los twits. Así obtuvieron datos de avistamientos de diferentes usuarios; estos cumplieron el rol de voluntarios sin tener noción de ello. En otros proyectos, como (Hart et al., 2018), se pidió a los voluntarios que compartan mediante twits las muestras encontradas y añadiendo un hashtag específico a dicha publicación para que luego pueda ser encontrada fácilmente a través de la búsqueda de ese hashtag. Twitter está disponible para múltiples plataformas incluyendo las móviles, con respectivas aplicaciones para los distintos sistemas operativos. Si bien ninguna aplicación de Twitter presenta widgets adaptables, nos da la posibilidad de subir fotos o videos directamente tomados con la cámara o desde el almacenamiento interno del dispositivo, siempre que hayamos concedido los permisos adecuados. Por otro lado, si realizamos un tweet sin conexión a internet, este completa su carga una vez recuperada la conexión a internet.

AppInventor²⁵, se trata de un artefacto de software que permite generar aplicaciones para Android. Es posible su resignificación para la recolección de datos en Ciencia Ciudadana, como bien se explica en (Ellul et al., 2013). El software de creación está disponible para Windows, Linux y Mac y permite desarrollar de forma sencilla aplicaciones que corran sobre el sistema operativo Android. Podemos incluir en estas aplicaciones formularios con los que recolectar datos debiendo establecer una base de datos propia como destino de los mismos, lo que puede acarrear costos adicionales. Los voluntarios deben obtener e instalar la aplicación en sus dispositivos. A diferencia de ODK, no requiere ningún conocimiento en programación, aunque AppInventor resulta más limitada en funciones y en los tipos de datos que el responsable del proyecto puede configurar pudiendo recolectar datos solo de tipo ubicación, foto y texto. Las aplicaciones resultantes no contemplan ningún tipo de adaptación ante restricciones de hardware o permisos.

Google Forms²⁶, es un artefacto de software para la creación de formularios en línea, y que ha sido resignificado en diferentes proyectos de Ciencia Ciudadana encontrados para recolectar datos que aporten voluntarios, algunos trabajos son (Portas et al., 2016) y (Hesley et al., 2017). Solo está disponible en forma de aplicación web, aunque presenta instalarse como aplicación en cualquier sistema operativo. Permite agregar campos de tipo texto, preguntas de opción múltiple y distintos tipos de archivos como fotos, videos, audios, documentos, entre otros a los formularios creados. Utiliza una planilla de Google Sheets para el almacenamiento de los datos.

²⁴ <https://twitter.com/>

²⁵ <https://appinventor.mit.edu/>

²⁶ <https://docs.google.com/forms/>

Los voluntarios solo necesitan recibir el link del formulario a completar para aportar datos. Al tratarse de una aplicación web, no presenta ningún tipo de adaptación ante restricciones de hardware o falta de permisos.

Google Sheets²⁷ se trata de un motor de planillas de cálculo. El relevamiento reveló que ha sido resignificada para la recolección de datos de Ciencia Ciudadana en proyectos como (Britton et al., 2013) y (Berland et al., 2019). Los responsables del proyecto crean una planilla, donde pueden agregar voluntarios a través de su respectiva cuenta de Google o pueden establecerla pública para que cualquiera, mediante solo un link, pueda editar dicha plantilla y así participar. Los voluntarios pueden editar el archivo agregando una fila por cada nueva entrada de datos. Como todos, tanto responsables como voluntarios, pueden manipular los datos directamente se debe tener cuidado de no modificar los ya existentes, esto podría generar un problema de pérdida de datos. Se nos permite recolectar datos de tipo texto, numérico, preguntas de opción múltiple, fecha, hora y solo algunos tipos de archivos, como foto o documentos, lo que limita la recolección de datos con esta herramienta. Google Sheets dispone tanto de una interfaz web como una aplicación para Android y iOS. No dispone de ninguna adaptación ante la falta de permisos que se necesiten para cargar algunos datos desde la aplicación móvil, como el de tipo foto, y, en cuanto a restricciones de hardware del dispositivo, solo nos posibilita realizar la carga de datos sin conexión con una posterior sincronización ante la recuperación de dicha conexión.

Google AppSheet²⁸ es un artefacto de software que permite el desarrollo de aplicaciones para Android y iOS y no requiere conocimientos en programación. Estas aplicaciones pueden mostrar o guardar datos de una forma sencilla en servicios de Cloud como Google Drive, Dropbox u Office365. En (Appenfeller et al., 2020) se emplea Google AppSheet para la creación de una aplicación de recolección de datos de Ciencia Ciudadana y donde se utiliza como base de datos una planilla de Google Sheets, lo que nos demuestra que es posible su resignificación en proyectos de Ciencia Ciudadana. Al usar otro artefacto para guardar los datos, con AppSheets nos encontramos con la limitación de solo a poder recolectar datos que sean compatibles a la base de datos que estemos utilizando. Además, si bien su uso básico es gratuito, requiere un pago en caso de utilizar algunas opciones avanzadas. Al igual que muchos artefactos de software vistos, nos permite retrasar la carga de datos en caso de no tener conexión, aunque no presenta ningún otro tipo de adaptación ante restricciones de hardware o falta de permisos, por lo que, si no concedemos permisos no podemos completar la carga de datos en caso de ser necesaria.

Por último, Google Docs²⁹, es un procesador de texto en la nube de libre utilización que puede resignificarse para su utilización en proyectos de Ciencia Ciudadana, en el relevamiento realizado se encontró el proyecto descrito en (Spicer et al., 2020), donde resignifican esta herramienta. Al igual que ocurría con Google Sheets, los encargados del proyecto pueden

²⁷ <https://docs.google.com/spreadsheets/>

²⁸ <https://www.appsheet.com/>

²⁹ <https://docs.google.com/document/>

compartir el documento con los voluntarios mediante la respectiva cuenta de Google de cada uno o pueden establecerlo como público y compartirla para que quien esté dispuesto pueda editarla y agregar datos a él, teniendo los mismos peligros de perder entradas de datos. Los datos posibles a recolectar pueden ser de tipo texto, fotos, videos, audios, entre otros. Está disponible a través de una aplicación web o puede descargarse su respectiva aplicación en dispositivos móviles con sistema operativo Android o iOS. Permite la edición del documento sin conexión, actualizándose su contenido en la nube una vez que vuelva a tenerse conexión a internet. No presenta ninguna adaptación ante la falta de permisos.

Los resultados del análisis pueden verse en la *Tabla 2*, donde se detalla por cada artefacto de software su comportamiento ante restricciones de hardware y permisos, además de la plataforma en la cual está disponible:

Tabla 2 – Comportamiento de los artefactos de software recolectados mediante el relevamiento ante restricciones de hardware y falta de permisos.

Aplicación	Plataforma	Comportamiento ante restricciones de hardware	Comportamiento ante falta de permisos
Sensr	Web, iOS	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	No.
INaturalist	Web, Android, iOS	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	Puede optarse para cargar uno u otro tipo de dato entre audio, foto, descripción, según el permiso otorgado
CitSci.Org	Web, Android, iOS	No.	No.
Zooniverse	Web, Android, iOS	No.	No.
KoboToolBox	Web, Android	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	Se permite una alternativa de tipo texto ante el pedido de una ubicación.
Epicollect	Web, Android, iOS	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	No.
ODK	Web, Android	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	No.
REDCap	Web, Android, iOS	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	No.

Aplicación	Plataforma	Comportamiento ante restricciones de hardware	Comportamiento ante falta de permisos
Twitter		Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	Se permite una alternativa al cargar una foto, desde el almacenamiento interno.
AppInventor	Android	No.	No.
Google Forms	Web	No.	No.
Google Sheets	Web, Android, iOS	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	No.
Google AppSheet	Android, iOS	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	No.
Google Docs	Web, Android, iOS	Posibilidad de retrasar la carga de datos si no se tiene acceso a internet	No

3.2.4 Resultados obtenidos

A partir del resultado del análisis presentado en la sección anterior, detallaremos en esta sección las conclusiones que se obtuvieron del mismo.

En la *Tabla 3* se resume que artefactos de software obtenidos del relevamiento realizado responden de alguna manera a las restricciones de hardware o falta de permisos propuestas para la implementación final de “Resuelvo Explorando” presentada en este trabajo, indicando dicho caso mediante una equis (X).

Tabla 3 – Resumen del comportamiento de los artefactos de software obtenidos del relevamiento frente a las restricciones de hardware y falta de permisos contemplados en “Resuelvo Explorando”

Artefacto	Restricciones de hardware				Falta de permisos			
	Conexión disponible	Tipo de conexión	Porcentaje de batería	Modo ahorro de energía	Cámara	Almacenamiento	Micrófono	Ubicación
Sensr	X							
INaturalist	X				X		X	
CitSci.org								
Zooniverse								
KoBoToolBox	X							X
EpiCollect	X							
ODK	X							
REDCap	X							
Twitter	X				X			
AppInventor								

Artefacto	Restricciones de hardware				Falta de permisos			
	Conexión disponible	Tipo de conexión	Porcentaje de batería	Modo ahorro de energía	Cámara	Almacenamiento	Micrófono	Ubicación
Google Forms								
Google Sheets	X							
Google Appsheet	X							
Google Docs	X							

En primer lugar, al observar cómo estos artefactos de software se adaptan ante restricciones de hardware podemos ver una tendencia clara, la mayoría de ellos, aunque no todos, tiene en cuenta la posibilidad de no poseer conexión a internet al momento realizar la carga de una entidad de datos, en cuyo caso, la acción se retrasa hasta el momento en que el dispositivo vuelve a tener conexión. Esto sin duda es algo necesario, en contraposición con las otras aplicaciones que en muchos casos ni siquiera permite su uso sin conexión, como es el caso de Zooniverse. Sin embargo, ninguno de estos artefactos tiene en cuenta el tipo de conexión que se dispone, es decir, si estamos conectados mediante WI-FI o datos móviles, y mucho menos que tipo de tecnología estamos empleando en los datos móviles (4G, 3G, 2G, etc.). Consideramos que esto debería tenerse en cuenta ya que, subir un tipo de dato multimedia, como una foto o video, puede consumir una carga de datos importante, que podría retrasarse hasta estar conectado a una conexión WI-FI, o, como alternativa, reemplazar la foto o video por un formato de datos más sencillo.

Otro aspecto que no ha sido tenido en cuenta por ningún artefacto de los analizados es el porcentaje de batería que posee el dispositivo al tomar la muestra. Es importante ya que, por ejemplo, obtener la ubicación o grabar un video son tareas que consumen una gran cantidad de batería y no sería satisfactorio para el usuario quedarse sin batería o con niveles muy bajos, sobre todo si se encuentra tomando muestras en lugares lejanos. Como alternativa, y para mayor conformidad con el usuario, podría verificarse, en lugar de un porcentaje de batería concreto, si el dispositivo tiene activado su modo ahorro de energía, presente hoy en cualquier dispositivo Android o iOS desde las últimas versiones. De esta forma, si se tiene un bajo nivel de batería, podría mostrarse una alternativa que consuma menos recursos a un tipo de dato que si consuma.

Con respecto a la adaptación de estos artefactos de software ante la falta de permisos, los resultados muestran que muy pocos artefactos de software de Ciencia Ciudadana tienen en cuenta esta problemática. Solo en algunos casos se permite una alternativa al tipo de datos que se solicita y solo para algunos tipos de datos disponibles en la aplicación. La más completa en

este aspecto es INaturalist³⁰, donde cada “observación”, como es llamada cada muestra en el artefacto, puede ser de diferentes tipos según se de acceso (conceda permiso) a un recurso u otro.

Podemos concluir entonces, que la adaptación en artefactos de Ciencia Ciudadana, basándonos en los obtenidos y analizados mediante el relevamiento realizado, es muy escasa o en algunos casos inexistente por completo. Con respecto, más específicamente, a las condiciones de hardware de cada dispositivo, la adaptación si es buena ante la falta de conexión a internet, pero no hay ninguna adaptación ante el tipo de conexión que existe o el nivel de batería que disponga el dispositivo. Ante la falta de permisos, el análisis refleja que la mayoría de los artefactos de software no tienen en cuenta esta cuestión.

Consideramos esto una problemática existente que puede no solo limitar la buena experiencia de usuario de un voluntario al utilizar el artefacto, sino también, puede reducir la cantidad de muestras obtenidas en un proyecto de Ciencia Ciudadana, ya que, por ejemplo, un usuario que no quiera entregar el permiso de cámara, no podrá subir información de la muestra observada si el artefacto no presenta ninguna alternativa. Además, estamos limitando el uso a dispositivos que dispongan en funcionamiento de todos sus sensores, lo cual también puede reducir la cantidad de potenciales voluntarios, algo que es de nuestro interés incrementar.

Así, con esta ampliación de la herramienta “Resuelvo Explorando” esperamos obtener un artefacto de software de recolección de datos de Ciencia Ciudadana que contemple estas restricciones, es decir, restricciones de hardware y falta de permisos y que se adapte mostrando una alternativa para obtener el mismo dato. De esta manera, se evitarán las problemáticas ya mencionadas.

³⁰ <https://www.inaturalist.org/>

4 PROPUESTA DE DESARROLLO

En este trabajo se presenta una evolución de “Resuelvo Explorando” (Di Claudio et al., 2020), una aplicación para dispositivos móviles con sistema operativo Android o iOS que permite, entre otros aspectos, la recolección de datos en proyectos de Ciencia Ciudadana.

“Resuelvo Explorando” forma parte de un ecosistema junto con “MoLE” (Llitas et al., 2019), una aplicación web para la instanciación de proyectos de recolección, donde es posible configurar (mediante formularios) los datos que deberán recolectar los voluntarios de dicho proyecto para tomar una muestra. Los formularios creados desde MoLE, se almacenan en una base de datos a través de una API que maneja la misma. Luego, “Resuelvo Explorando” se comunica con esta API para descargar los formularios creados y así permitir a los voluntarios recolectar los correspondientes datos.

Un tercer elemento fue sumado al ecosistema formado por “Resuelvo Explorando” y “MoLE” mediante la adopción de un componente para la creación de widgets adaptables en aplicaciones móviles en la aplicación “Resuelvo Explorando”, dicha integración fue presentada en (Di Claudio et al., 2020). Este componente nos permite desarrollar widgets funcionales que se muestren de una u otra manera e incluso tengan diferentes funciones, según las restricciones de hardware propias del dispositivo al momento de su ejecución, como lo son el nivel de batería o el tipo de conexión a internet, o la falta de permisos para ejecutar recursos que nos proporciona el sistema operativo, como son el acceso a la cámara, al almacenamiento interno, o la ubicación mediante GPS, entre otros. Además, dicho componente, facilita la consulta de las restricciones de hardware que maneja y la obtención de permisos referentes a recursos del sistema operativo, realizando estas tareas, que suelen resultar bastante engorrosas, sin necesidad de intervención del programador.

La aplicación “Resuelvo Explorando” implementa (usando la componente descrita anteriormente) widgets adaptables en sus funciones de recolección de datos de tipo ubicación y fotografía, tal como fue presentado en (Di Claudio et al., 2020). Así, al realizar una de estas tareas, el voluntario obtendrá una mejora significativa en su experiencia de usuario al mostrársele un método de carga diferente según los permisos que éste haya otorgado a la aplicación y según las condiciones de hardware actuales del dispositivo con el que esté recolectando la muestra. Sin embargo, en “Resuelvo Explorando” existen otros tipos de datos y de tareas posibles a realizar por los voluntarios que son configurables durante la creación del formulario mediante “MoLE”. En la versión publicada en (Di Claudio et al., 2020), las tareas que involucran audio, video, lectura de códigos QR, entre otras, aún no disponen de una implementación adaptable. Esto, podría imposibilitar al voluntario llevar adelante una de estas tareas, como no es el caso de las tareas que sí poseen una implementación adaptable, donde siempre se muestra una alternativa posible al usuario para que este pueda completar la tarea.

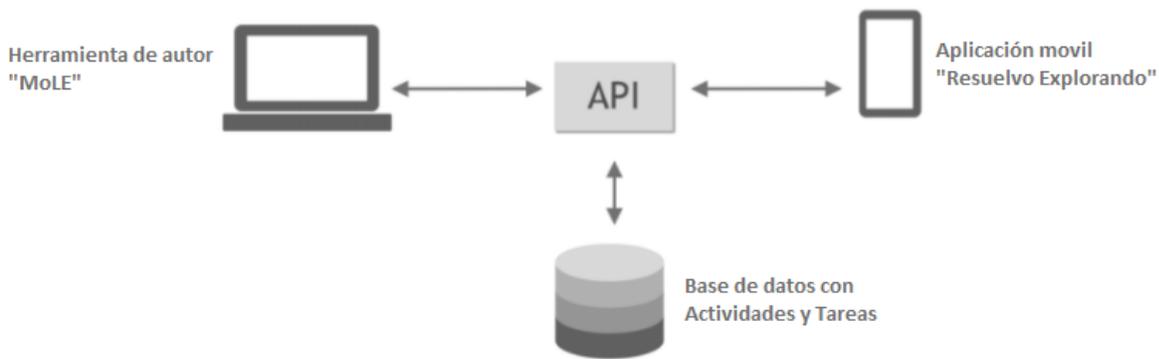
Además, como se vio en el relevamiento realizado y descrito en el capítulo anterior, la adaptación en artefactos de software reutilizables de Ciencia Ciudadana o resignificados para el uso en dicho dominio, presentan muy pocas funciones adaptables o, en algunos casos, ninguna, lo que resulta en una problemática a abarcar desde la ingeniería de software. La solución a dicha problemática no solo mejoraría la experiencia de los voluntarios al utilizar la aplicación, sino que además resultaría en una mayor cantidad de muestras recolectadas, al no estar estas imposibilitadas de tomarlas en ningún momento.

Es por ello que en este trabajo y a raíz de los puntos detallados anteriormente, se desarrolla, inicialmente, una ampliación de la componente para la creación de widgets adaptables para contemplar nuevos permisos y nuevas restricciones de hardware del dispositivo, como por ejemplo la restricción del modo ahorro de energía. Esta nueva versión del componente se utilizará para la creación de dos nuevos widgets adaptables, como son la recolección de datos a través de videos y a través de audio. El nuevo componente y los nuevos widgets serán adoptados en “Resuelvo Explorando”, a partir de la versión presentada en (Di Claudio et al., 2020) para disponer así de widgets adaptables en todas las funciones de recolección que lo necesiten. Adicionalmente, se actualizarán las dependencias de dichos proyectos con el fin de solucionar posibles problemas de compatibilidad, se realizará una revisión de código disponible del componente y se analizará la posibilidad de realizar refactoring en distintas áreas del mismo, así como otras mejoras. Además, y aprovechando la adopción de una nueva versión del componente, se analizarán posibles mejoras en los widgets adaptables ya existentes en la aplicación.

Este capítulo se organiza de la siguiente manera: Inicialmente se describe la arquitectura de la solución propuesta para la instanciación de proyectos de Ciencia Ciudadana y la recolección de datos en dichos proyectos a través de cada uno de los componentes que conforman dicho ecosistema. Luego se describirán las tecnologías usadas en cada uno de los componentes, justificando la elección de las mismas. Se continuará describiendo el componente para la creación de widgets adaptables desarrollado, tanto en sus funciones originales como en las implementadas en este trabajo. Finalmente, se especificará la aplicación “Resuelvo Explorando”, haciendo especial énfasis en los widgets adaptables que considera.

4.1 ARQUITECTURA DE LA SOLUCIÓN

El ecosistema que conforma las herramientas “Resuelvo Explorando” y “MoLE” constituyen la arquitectura que se muestra en la *Figura 1*.



*Figura 1 – Arquitectura de la solución propuesta
[Traducida del inglés de (Lliteras et al. 2019)]*

El ecosistema se compone entonces de una herramienta web de autor llamada “Mobile Learning Experience”, o simplemente por su abreviatura, “MoLE, la cual permite la creación de formularios de recolección de datos, llamados actividades. En ella los responsables de proyecto conforman las tareas de recolección de datos que deberán realizar los voluntarios que se unan al proyecto. Estos formularios resultantes son guardados por la misma aplicación en una base de datos mediante una API que esta posee. Además, desde la misma aplicación web es posible editar y eliminar las actividades y las tareas creadas.

La API dispone de los métodos HTTP estándares para la creación, eliminación y actualización de actividades y tareas, además de sus respectivos métodos de obtención de las mismas. Esta administra directamente una base de datos que utiliza el motor MongoDB, y que consta de dos simples colecciones: Actividad y Tarea, las cuales se relacionan en una cardinalidad de uno a muchos, es decir, una actividad posee un conjunto de tareas. Cada actividad posee unos campos básicos que identifican el proyecto (título y descripción), mientras que cada tarea consta de un tipo, el cual indicará cuál será el formato de dato que el voluntario debe tomar (por ejemplo, una foto o una ubicación) y una serie de descripciones que explican con mayor detalle al voluntario las acciones que debe realizar que varían según el tipo de tareas (por ejemplo, una consigna o una descripción de lo que debe incluir).

El tercer componente de la arquitectura es “Resuelvo Explorando”, una aplicación móvil disponible para dispositivos con sistema operativo Android y iOS que nos permite comunicarnos con la API descrita anteriormente y descargar las actividades, con sus respectivas tareas, que el usuario desee, para de esta forma, en el rol de voluntario, suscribirse a un proyecto de Ciencia Ciudadana. Luego, en dicha aplicación el usuario puede realizar las tareas de recolección de datos que se dispone para el proyecto descargado.

“Resuelvo Explorando” emplea para los distintos tipos de datos a recolectar y para la lectura de actividades y tareas, widgets adaptables a restricciones de hardware o falta de permisos, creadas con el componente para la creación de los mismos que se desarrolla más adelante en este capítulo.

En resumen, la arquitectura del ecosistema es bastante sencilla, existe una herramienta de autor para la creación de “actividades” que representan proyectos de Ciencia Ciudadana y en los que se configura los datos a recolectar por los voluntarios. El voluntario adquiere la aplicación móvil “Resuelvo Explorando” y descarga mediante un código QR o el nombre del mismo las tareas de recolección perteneciente a un proyecto de Ciencia Ciudadana ya creado. Finalmente, el voluntario, a través de la misma aplicación realiza la carga de datos de la muestra para cada una de las tareas.

4.2 TECNOLOGÍAS EMPLEADAS

En esta sección describiremos las tecnologías empleadas por cada elemento que compone la arquitectura de la solución descrita en la sección anterior, justificando la elección dichas tecnologías.

La herramienta de autor “MoLE” fue desarrollada con la tecnología React³¹, el cual es una biblioteca de JavaScript para la creación de interfaces de usuario interactivas de una sola página, de forma sencilla. Por otro lado, la aplicación “Resuelvo Explorando”, fue desarrollada en React Native³², la cual se trata de una extensión de la tecnología React utilizada en “MoLE”, que nos permite la creación de aplicaciones para Android y iOS. La particularidad de esta tecnología es que, si bien se escribe código en JavaScript, se recompila a código nativo, lo que permite una mejor performance de ejecución. El componente para la creación de widgets adaptables también fue desarrollado empleando la misma tecnología, React Native.

Cómo React Native está basado en React, la utilización de uno u otro es indistinta en términos de conocimientos requeridos, por lo que emplear una tecnología para la aplicación web y otra para las aplicaciones móviles no requirió tiempos y esfuerzos para aprender diferentes y nuevas tecnologías.

Además, utilizar tecnologías como React y React Native, nos brinda, al estar empleando una arquitectura basada en componentes, facilidades al momento de buscar reutilización y aislamiento de funcionalidades, característica que se consideró esencial para el desarrollo de componentes adaptables. Estos aspectos son muy importantes considerarlos en la propuesta de este trabajo, ya que esto hace que el componente desarrollado sea fácil de exportar e incorporar a cualquier nuevo proyecto.

Redux³³ también es otra de las tecnologías utilizadas en “MoLE” y “Resuelvo Explorando”. Este se trata de un contenedor del estado de aplicaciones, lo que nos permite mantener un estado inmutable de la aplicación y definir acciones que modifiquen ese estado. Esto ayuda a construir aplicaciones que se comuniquen de manera consistente, ya que tanto el estado como

³¹ <https://es.reactjs.org/>

³² <https://reactnative.dev/>

³³ <https://es.redux.js.org/>

las modificaciones en este son totalmente predecibles. Redux se añade como dependencia de nuestra aplicación y es compatible tanto React y React Native, así como con cualquier librería de creación de vistas.

Luego, tanto para el componente que facilita la creación de widgets adaptables como para la aplicación móvil “Resuelvo Explorando”, se decidió emplear Expo³⁴, el cual se trata de un framework y una plataforma para aplicaciones React Native universales que brinda un conjunto de herramientas para facilitar la implementación de aplicaciones móviles. Se decidió emplear dicha tecnología en este componente debido a que incorpora diversa cantidad de funcionalidad de funcionalidades, que pueden ser incorporadas a nuestra aplicación por medio de nuevas dependencias y que facilitan el desarrollo de muchas de las funcionalidades de nuestra aplicación.

Entre los componentes de Expo utilizados para el desarrollo del componente de creación: Expo-Permissions³⁵ y Expo-Battery³⁶. El primero de ellos, nos permite manejar los permisos de manera más sencilla y de forma indistinta entre Android y iOS, es decir, la propia extensión se encarga de solicitar de una u otra manera el permiso equivalente entre ambos sistemas operativos. Por otro lado, Expo-Battery contiene toda la funcionalidad requerida para la consulta de las condiciones de batería del dispositivo, como el porcentaje respectante y la condición de activación del modo ahorro de batería.

Para el componente de generación, además, se utilizó una librería propia de React Native, llamada NetInfo³⁷, que nos permite consultar las condiciones de conexión del dispositivo, esto es, si posee conexión y, en caso, el tipo de conexión.

El framework Expo también fue empleado en la implementación de la herramienta “Resuelvo Explorando”. Además de la necesidad de añadirlo como dependencia al utilizar el componente para la creación de widgets adaptables, Expo nos permite la fácil prueba en desarrollo de la aplicación en nuestro propio smartphone de manera local. Con solo tener la aplicación móvil de Expo Go, disponible tanto para Android³⁸ como para iOS³⁹, instalada en el dispositivo, podemos ejecutar allí el código local que poseamos en nuestra computadora de una manera rápida y fácil, lo que facilitó mucho las pruebas durante el desarrollo.

Otra razón por la cual se decidió el uso de Expo es por que posee extensiones para manejar de forma más sencilla los sensores del dispositivo como la cámara o la ubicación. Así, por ejemplo, se empleó como dependencia la extensión de Expo, Expo-Camara⁴⁰, que nos permite con solo

³⁴ <https://expo.io/>

³⁵ <https://docs.expo.io/versions/latest/sdk/permissions/>

³⁶ <https://docs.expo.io/versions/latest/sdk/battery/>

³⁷ <https://github.com/react-native-netinfo/react-native-netinfo>

³⁸ https://play.google.com/store/apps/details?id=host.exp.exponent&hl=es_PR

³⁹ <https://apps.apple.com/es/app/expo-client/id982107779>

⁴⁰ <https://docs.expo.io/versions/latest/sdk/camera/>

unas líneas de código, lanzar la cámara, tomar la fotografía y obtenerla. Esto redujo los tiempos de implementación y de aprendizaje y la complejidad del código en gran medida. Otros componentes fueron empleados para obtener la ubicación (Expo-Location⁴¹) y acceder al almacenamiento interno del dispositivo (Expo-ImagePicker⁴²).

La API fue desarrollada con LoopBack⁴³ en su versión 4. Se trata de un framework basado en Node.JS y TypeScript de código abierto y altamente extensible que nos permite la creación de API rápidamente. Le indicamos a este las entidades y atributos de nuestra base de datos y automáticamente genera la API y todos los métodos correspondientes. Nuestro sistema consta de una base de datos sencilla y no requiere consultas complejas, por lo que esta herramienta es muy adecuada y nos permite reducir tiempos y esfuerzos en el desarrollo.

La instanciación de la base de datos también quedó a cargo de LoopBack, solo teniendo que indicar el motor a utilizar y la dirección de la misma. Se empleó una base de datos en MongoDB⁴⁴, el cual es no relacional, sino basada en documentos. La decisión se basa en la sencillez en el uso de la misma.

Para la comunicación con la API, tanto desde “MoLE” como de “Resuelvo Explorando” fue empleada la librería Axios⁴⁵, la cual se trata de una librería de JavaScript para realizar peticiones HTTP a servidores y para manejar fácilmente las promesas resultantes.

4.3 COMPONENTE PARA LA CREACIÓN DE WIDGETS ADAPTABLES

En esta sección se detalla el componente para la creación de widgets adaptables desarrollado, así como también el manejo de permisos y las restricciones de hardware que emplea.

4.3.1 Permisos

Los dispositivos móviles pueden considerarse dispositivos sumamente personales, debido a la creciente cantidad de información personal que recopilan (Olejnik et al., 2017). Para controlar el acceso de aplicaciones de terceros a esta información personal y confidencial, y evitar que aplicaciones maliciosas accedan a recursos no autorizados, las plataformas móviles, como Android y iOS, dependen de un sistema de permisos con el cual los usuarios pueden permitir o denegar las solicitudes de permisos de las aplicaciones.

⁴¹ <https://docs.expo.io/versions/latest/sdk/location/>

⁴² <https://docs.expo.io/versions/latest/sdk/imagepicker/>

⁴³ <https://loopback.io/>

⁴⁴ <https://www.mongodb.com/es>

⁴⁵ <https://github.com/axios/axios>

En los actuales sistemas operativos móviles como Android⁴⁶ y iOS⁴⁷, los permisos representan el acceso por parte de aplicaciones a recursos sensibles que pueden llegar a poner en riesgo la privacidad del usuario. Existen dos tipos de permisos: de bajo riesgo y de peligro. Los primeros constituyen información que no afecta directamente a la identidad del usuario, como por ejemplo la zona horaria configurada en el dispositivo; estos son otorgados a las aplicaciones sin que el usuario intervenga. Por el contrario, aquellos considerados de peligro son los que pueden afectar directamente a la privacidad del usuario. Estos últimos se encuentran inaccesibles de forma predeterminada y se debe solicitar permiso al usuario para acceder a uno de estos. El usuario puede otorgar o denegar el permiso.

Estos recursos considerados peligrosos pueden ser:

- Datos restringidos, como el estado del sistema y la información de contacto del usuario
- Acciones restringidas, como el acceso a un dispositivo vinculado o a un sensor propio del dispositivo en que se ejecuta la aplicación, como la cámara.

Los sistemas de permisos son la principal defensa que las plataformas móviles, como Android y iOS, ofrecen a los usuarios para proteger sus datos privados de aplicaciones inseguras (Olejnik et al., 2017). Estudios recientes demuestran que a menudo nos encontramos con estas aplicaciones inseguras, que hacen un mal uso de los permisos para acceder a información privada o tienen trampillas, a través de las cuales otras aplicaciones maliciosas pueden hacer lo mismo (Gasparis et al., 2019). En esta tesina nos concentraremos en considerar el caso en que un usuario decida no otorgar esos permisos peligrosos, ya sea para proteger su privacidad como por otras cuestiones, para permitir la adaptación ante esta situación.

Una aplicación define en su código fuente los permisos que esta requiere para su funcionamiento, sin embargo, actualmente en iOS y en Android, la solicitud de permisos al usuario se realiza de manera dinámica. Es decir, se puede instalar la aplicación en nuestros dispositivos, sin dar ningún acceso a los recursos que necesita, pero una vez que la aplicación lo requiera se muestra al usuario un cuadro de diálogo donde este conceda o no el permiso y por tanto el acceso al recurso. Una vez que es otorgado un permiso a la aplicación este no se vuelve a pedir, sino que se toma como otorgada cada vez que ésta lo requiera.

En la *Figura 2* se muestra un cuadro de diálogo de una aplicación ejecutada en Android, en este caso Instagram, pidiendo el permiso de acceso a cámara, para poder tomar una fotografía.

⁴⁶ <https://developer.android.com/guide/topics/permissions/overview>

⁴⁷ https://developer.apple.com/documentation/uikit/protecting_the_user_s_privacy/requesting_access_to_protected_resources



Figura 2 - Solicitud de permiso de acceso a la cámara por parte de la aplicación Instagram

A partir de dicho momento, si se concede el permiso a la aplicación, ésta podrá utilizar el recurso sin problemas, pero ¿qué pasa si no se concede? En muchos casos la aplicación no puede seguir funcionando o no puede ejecutar la funcionalidad deseada. Es el caso de, por ejemplo, la aplicación Google Fotos, que al no otorgar el permiso de acceso al almacenamiento interno muestra la interfaz de la *Figura 3*.

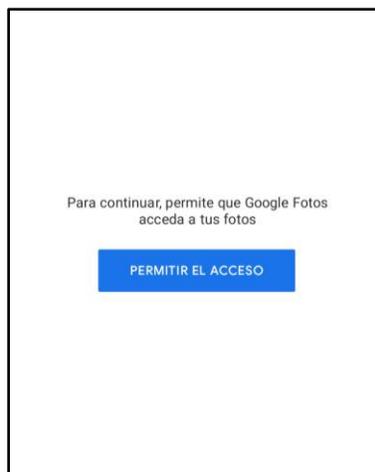


Figura 3 - Mensaje de la aplicación móvil Google Fotos al no otorgar permiso de acceso al almacenamiento del dispositivo

Otra cuestión importante al tratar con permisos es el grado de complejidad que presenta el correcto manejo de estos. Las peticiones de permisos se realizan mediante llamados asincrónicos y mediante librerías que implementa el sistema operativo sobre el que estamos construyendo nuestra aplicación. Esto, puede repercutir en tiempos de aprendizaje largos y dificultades para el manejo de permisos en el momento de la implementación.

El componente creado y presentado en esta sección se creó con el objetivo de simplificar en grandes medidas estas problemáticas contempladas: imposibilidad de utilizar una aplicación ante la falta de algún permiso y tiempos y esfuerzos considerables en el manejo de permisos en los sistemas operativos móviles.

Por un lado, el componente nos permite establecer distintas interfaces a renderizar, cada una necesita de una lista de permisos para cumplir su función. El orden establecido de estas interfaces hace su evaluación, la primera que resulte correcta, es decir que sus permisos sean otorgados por el usuario, se mostrará a esta. Finalmente existe un componente por defecto que no requiere permisos para su ejecución. De esta forma siempre existe una componente a renderizar, si se da o no los permisos necesarios, es decir, siempre existe una alternativa a mostrar al usuario. Dependerá de los programadores mostrar en las distintas interfaces distintas formas alternativas de realizar la función inicial. Solucionamos así la primera de las dos problemáticas propuestas.

Para solucionar la segunda problemática, el componente posee toda la implementación para la petición de permisos, sin que el programador tenga que tener conocimientos sobre los mismos. Al establecer la lista de interfaces a renderizar, solo debe colocar por cada una, en forma de constantes definidas, los permisos que se deben solicitar. El componente automáticamente mapea las constantes a permisos y solicita la petición al usuario. La respuesta de la misma también es manejada por la componente.

Además, el componente no diferencia los permisos de los dos sistemas operativos: Android y iOS, sino que lo maneja de forma transparente, siendo en ambos similares. Esto es gracias a el uso de Expo Permissions.

De esta forma el componente no solo facilita la administración de permisos de una aplicación a los programadores, sino que permite crear widgets adaptables en aplicaciones de una forma rápida y fácil.

4.3.2 Restricciones de hardware

El componente para la creación de widgets adaptables presentado en este capítulo, además de administrar las interfaces a renderizar con respecto a los permisos que otorgue el usuario a la aplicación, también permite hacerlo con respecto a las restricciones de hardware del dispositivo en el momento dado.

En este trabajo, se llama restricciones de hardware a cierta limitación o condición que debe ser cumplida sobre algún componente del dispositivo. No se tratan de recursos, por lo que no los

concede el usuario, sino que deben ser evaluados mediante la exposición de librerías del sistema operativo. Un ejemplo es el porcentaje de batería del dispositivo.

En particular, para la tesina, se consideran las siguientes restricciones de hardware sobre el dispositivo móvil: si posee conexión a internet (ACTIVA/INACTIVA), el tipo de conexión a internet (WIFI, 4G, 3G, 2G), el nivel de batería (porcentaje restante), la condición de activación del modo ahorro de energía (SI/NO).

Todas las restricciones previamente mencionadas pueden ser añadidas como condición para renderizar una cierta interfaz; así, por ejemplo, el primer componente, requiere que se disponga de una conexión a internet activa para poder mostrarse. Esto no quita, que además de dicha restricción de hardware pueda necesitar de algún permiso adicional, ambas opciones coexisten en el componente aquí propuesto.

A continuación, se describe cada una de las restricciones posibles que contempla el componente creado y presentado en esta tesina.

La necesidad de internet para renderizar una interfaz está contemplada por el componente, es decir, podemos renderizar o no un componente si nuestro dispositivo está conectado a una red de internet cualquiera sea el tipo. Además de esto podemos especificar qué tipo de conexión se requiere para dicha interfaz, teniendo en cuenta tanto WI-FI como las redes móviles 4G, 3G y 2G. De esta forma podemos renderizar, por ejemplo, una interfaz que consume grandes cantidades de datos solo si estamos conectados por WI-FI, pero en caso contrario, contar con una interfaz que consuma menos recursos. Lo anterior podría mejorar la experiencia de usuario al utilizar la aplicación.

Otra restricción contemplada es el nivel de batería que dispone el dispositivo en el momento de renderizar una interfaz. Podemos establecer un nivel numérico para una interfaz en concreto, por ejemplo, el 50%. De esta forma se busca que, si la aplicación requiere un consumo excesivo de la batería, tener una interfaz alternativa que sea más liviana y renderizar esta ante un nivel de batería menor, por ejemplo, por debajo del 20%. De esta forma se pretende que el usuario tenga menos posibilidad de quedarse sin batería durante el uso de la aplicación. El consumo de energía se ha vuelto un factor crítico en la satisfacción del usuario (Wilke et al., 2013), por lo que considerando un menor consumo en escenarios donde el nivel de batería es bajo se quiere aumentar esta satisfacción.

Por último, la posibilidad de contemplar si el dispositivo tiene el modo ahorro de energía activado es una restricción a la hora de mostrar una interfaz. Este modo fue añadido en las últimas versiones tanto de Android como de iOS y presenta un modo donde los recursos del sistema se limitan para ahorrar batería. De esta manera, por ejemplo, el sistema operativo puede bajar la frecuencia del procesador o el brillo de la pantalla. Esto significa que el usuario está intentado ahorrar batería por lo que mostrar una interfaz que consuma muchos recursos podría no ser beneficioso para este.

4.3.3 Uso del componente y ventajas

El componente de software para la creación de widgets adaptables en dispositivos móviles presentado en esta tesina, como bien se mencionó anteriormente, nos permite principalmente la creación de widgets que se adapten ante restricciones de hardware del dispositivo y la falta de permisos otorgados por el usuario, condiciones verificadas por el mismo componente en tiempo de ejecución.

Como bien se definió anteriormente, un widget es una pequeña aplicación con interfaz y funcionalidad propia⁴⁸, que a su vez puede ser insertadas en otras aplicaciones más complejas. En este caso, el componente permite crear widgets que permiten lanzar otros componentes con interfaces y funcionalidades variables, y que se presentarán al usuario según el resultado de las condiciones establecidas y necesarias para cada una. De esta forma, se logran interfaces adaptables a falta de permisos concedidos por el usuario o restricciones de hardware del dispositivo en el que la aplicación corre.

El componente facilita en gran medida la creación de estas interfaces a los desarrolladores de aplicaciones. Dicho componente posee todo el código necesario para la obtención de permisos, pedidos al sistema operativo, y la respectiva verificación de los mismos, todo en tiempo de ejecución, así como también, posee en su implementación la consulta de las condiciones de hardware actuales del dispositivo y la evaluación de estas. Es por ello, que el desarrollador ve limitada su tarea a definir las interfaces de su aplicación y su funcionalidad. Para cada una de estas definirá a modo de constantes, los permisos y restricciones de hardware requeridas para mostrárselo al usuario. Esto reduce los tiempos de desarrollo, la cantidad de líneas de código a escribir por el desarrollador y la complejidad de las mismas al crear este tipo de interfaces.

El uso del componente consiste básicamente en la creación de un listado de las distintas interfaces correspondientes a un mismo widget (cada una implementará su respectiva funcionalidad), el orden del listado determina el orden de evaluación de cada una a través de las condiciones requeridas para renderizar la misma, así, la primera que cumpla las condiciones (permisos necesarios obtenidos por el usuario y condiciones de hardware del dispositivo) será la mostrada al usuario. De este modo si el resultado de evaluar las condiciones establecidas por el desarrollador para la primera interfaz del listado resulta exitoso, dicha interfaz se renderiza; en el caso contrario, es decir, alguna de las condiciones no se cumple, ya sea la falta de algún permiso o alguna condición de hardware que no se cumple, se continúa a evaluar si es posible mostrar la siguiente interfaz de la lista.

Dicha lista de interfaces consiste de un arreglo donde cada componente es un diccionario. Cada diccionario se compone de, además del componente correspondiente a la interfaz a renderizar, de un arreglo de permisos a conceder por el usuario, del tipo de conectividad

⁴⁸ <https://www.w3.org/TR/widgets/>

requerida y las condiciones de batería necesarias (ya sea un porcentaje específico o la condición de activación del modo ahorro de energía). El listado debe tener al menos una componente.

Además, el componente requiere definir una interfaz por defecto. Esta no debe de precisar de ningún permiso ni de ninguna condición de hardware establecida para mostrarse. Esta interfaz, por tanto, no requiere ninguna evaluación ya que no presenta ninguna restricción para usarse y será mostrada al usuario solo en caso de que ninguna interfaz del listado cumpla con las condiciones necesarias, tratándose esta última interfaz de una última opción segura a mostrar al usuario.

El siguiente código muestra a modo de ejemplo la instancia de un arreglo de alternativas a mostrar con sus respectivas restricciones y permisos y el llamado a dicho componente con la definición de un componente por defecto. Dichos fragmentos de código fueron extraídos del widget adaptable para la obtención de un video desarrollado durante esta tesina y que se presenta más adelante.

```
componentList = [
  ({
    permission: [PermissionConstants.CAMERA],
    powerSaverCondition: PermissionConstants.NO_POWER_SAVER,
    batteryLevelRequire: 25,
    component: (<VideoTakeComponent setVideo={setVideo} task={task}/>)
  }),
  ({
    permission: [PermissionConstants.MEDIA_LIBRARY],
    component: (<VideoGalleryComponent setVideo={setVideo} task={task}/>)
  }),
]

render() {
  return (
    <PermissionAwareComponent
      permissionComponentList={this.componentList}      defaultComponent=
      {(<MultimediaDefaultComponent task={task} set={setVideo}/>)}
    />
  )
}
```

El componente soporta todos los permisos disponibles de Android y iOS, estos pueden verse listado en la documentación del framework utilizado para manejarlos⁴⁹. La equivalencia de estos es manejada automáticamente por el componente, no habiendo así diferencias si desarrollamos un permiso para Android, iOS o ambos, lo que permite la reutilización de código. Los permisos contemplados son: acceso a notificaciones, notificaciones de usuario, cámara, ubicación, grabación de audio, contactos, almacenamiento en modo lectura o escritura, calendario, recordatorios (solo para iOS) y brillo del sistema.

⁴⁹ <https://docs.expo.io/versions/latest/sdk/permissions/#permission-types>

Por otro lado, las restricciones de hardware contempladas por el componente se dividen en dos grupos conectividad y batería. En cuanto a conectividad, podemos establecer que un componente se renderice siempre que tengamos conexión (cualquiera sea el tipo), cuando no tenemos conexión, solo cuando estamos conectados a WI-FI o solo cuando lo estemos a conexión de datos móviles. Podemos especificar para esta última el tipo de conexión móvil entre 4G, 3G o 2G.

En el *Anexo A* de esta tesina, se presenta una guía de utilización del componente donde se describe la forma de importación, las dependencias necesarias y el modo de utilización del componente. Dicho componente se encuentra disponible para su libre uso en un repositorio creado en Github⁵⁰.

4.4 WIDGETS ADAPTABLES PARA LA RECOLECCIÓN DE DATOS

En esta sección se presenta cada uno de los widgets adaptables desarrollados como parte de esta tesina y mediante el componente para la creación de los mismos presentado y desarrollado en la sección anterior. Cada uno presenta diferentes alternativas para realizar una misma acción, adaptándose a las restricciones de hardware propuestas y a los permisos brindados por el usuario en su dispositivo móvil, necesarios para cada alternativa. A continuación, se lista y explica brevemente cada uno de ellos.

Los **widgets adaptables** desarrollados y sus alternativas de adaptación son:

- **Obtener una imagen:** mediante la captura de la misma con la cámara, mediante el acceso al almacenamiento del dispositivo, o mediante una descripción textual de la misma.
- **Obtener una ubicación:** mediante la utilización del GPS que posee el dispositivo móvil, mediante la ubicación manual por parte del usuario en un mapa, o mediante un texto donde el usuario indique la dirección.
- **Obtener un video:** mediante la grabación con la cámara del dispositivo móvil, mediante el acceso al almacenamiento interno, o mediante una descripción textual de lo que mostraría el video.
- **Obtener un audio:** mediante la utilización del micrófono del dispositivo para grabar un archivo de voz o mediante un texto que transcriba el audio.

De esta forma se presenta ante usuarios diferentes, con diferentes necesidades y/o preferencias de interacción, el widget se adapta a este en lugar de obligar al usuario a adaptarse a él. Como el sistema guarda algunas elecciones posibles de los usuarios, como su elección ante la pregunta sobre si otorgar dicho permiso, la interfaz se adapta ante estas elecciones una sola vez, sin tener que volver a preguntar una próxima vez. Dicha adaptación se muestra a

⁵⁰ <https://github.com/cientopolis/mutableWidget>

continuación para cada uno de los widgets desarrollados, en los que se los presenta con más detalle.

4.4.1 Widget adaptable para la obtención de una imagen

Este primer widget tiene la funcionalidad de obtener una imagen captada por el usuario, para lo cual es necesario el permiso de acceso a la cámara y, además, se consideraron restricciones de hardware para realizar dicha acción. El widget presenta dos alternativas ante el incumplimiento de estas condiciones.

En primera instancia, el widget evaluará mostrar una interfaz de cámara con la cual el usuario pueda tomar una fotografía en el momento. Para poder mostrar dicha interfaz, la aplicación solicitará al usuario el permiso de cámara necesario para poder accederla (se trata de un recurso considerado de peligro). Por otro lado, se consideró necesario, para mostrar esta interfaz de cámara, que el dispositivo tenga desactivado el modo ahorro de energía, esto debido a que si este lo tendría activado nos diría que el usuario quiere extender la duración de la batería de su dispositivo. Si en dicho caso, abrimos la cámara y tomamos una fotografía, acciones que pueden consumir batería considerablemente, lograríamos una experiencia de usuario desfavorable. Cabe destacar que, si esta restricción de hardware no se cumple, el permiso de cámara no se solicita por lo que se pasa directamente a evaluar la siguiente alternativa.

Si el permiso de cámara no fue concedido, o bien, las restricciones de hardware no fueron dadas, es decir, el modo ahorro de energía se encontraba activado, la aplicación no podrá tomar una fotografía en el momento. Sin embargo, el widget desarrollado evitará que el caso de uso termine mostrando una forma alternativa para obtener la imagen. Esta alternativa consiste en obtener dicha imagen desde el almacenamiento del dispositivo, es decir, una fotografía que el usuario haya tomado con anterioridad y ahora desee compartir.

En este segundo caso, el sistema necesita, para poder accederlo y obtener de allí una imagen, el permiso de acceso al almacenamiento del dispositivo. En caso de que se conceda, el usuario podrá seleccionar desde su galería de imágenes, aquella que desee compartir con la aplicación. En caso de no conceder dicho permiso, el sistema deberá adaptarse nuevamente.

La tercera y última alternativa, es una interfaz por defecto que no requiere ningún permiso ni tampoco fue considerada ninguna restricción de hardware para su utilización. En ella el usuario, en lugar de compartir una imagen que no se pudo obtener mediante las dos alternativas anteriores, puede completar un cuadro de texto describiendo lo que la imagen contendría.

De esta forma, el widget muestra dos formas de obtener una imagen, y, ante no poder disponer de ella por ninguna, una tercera interfaz permite describir textualmente la imagen que el usuario no compartió. Así, el usuario siempre podrá completar el caso de uso, y no se ve limitado, ya sea por preferencias de privacidad o por alguna limitación de su dispositivo, como puede ser, por ejemplo, su sensor de cámara está roto.

El circuito de funcionamiento del widget se puede ver en la Figura 4.

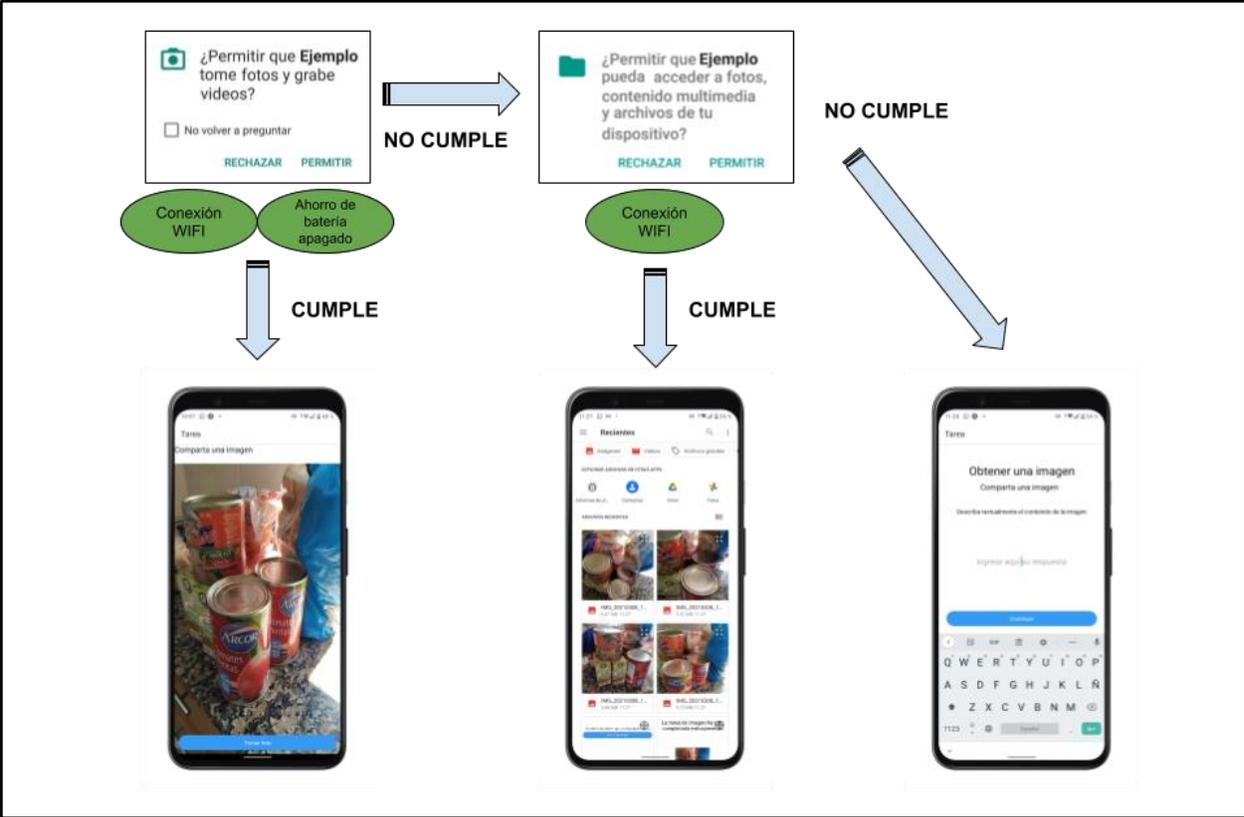


Figura 4 - Comportamiento del widget adaptable para la obtención de una imagen

En resumen, las condiciones necesarias (permisos necesarios o condiciones de hardware requeridas) para mostrar las diferentes alternativas al realizar la acción de obtención de fotografía, según fue establecida en la creación por medio del componente para la generación de widgets adaptables, puede visualizarse en la *Tabla 4*.

Tabla 4 – Resumen de las distintas formas de obtener una imagen que dispone el widget adaptable creado.

Acción: Obtener una imagen				
Forma de llevar adelante la acción	Permisos necesarios	Conexión a internet requerida	Porcentaje de batería requerido	Modo de ahorro de batería desactivado
Capturar una imagen utilizando la cámara	Acceso a la cámara	NO	NO	SI
Obtener una imagen desde el almacenamiento interno del dispositivo	Acceso al almacenamiento interno	NO	NO	NO
Obtener una descripción textual	Componente por defecto. No requiere ni permiso ni establece restricción de hardware			

A diferencia del presentado en (Di Claudio et al., 2020), al widget descrito en esta sección se lo mejoró mediante el uso de la nueva versión del componente, para incluir el modo ahorro de energía en la primera alternativa: requiere que esté desactivado para tomar acceder a la cámara.

4.4.2 Widget adaptable para la obtención de una ubicación

Este segundo widget adaptable desarrollado, ya presentado en (Di Claudio et al., 2020) y mejorado en esta tesina, posee toda la funcionalidad para obtener la ubicación del dispositivo, para lo cual es necesario el permiso de acceso al GPS del dispositivo, así como también otras restricciones de hardware que se consideraron necesarias establecer. El widget presenta dos alternativas.

Para obtener la ubicación el widget solicita al usuario el permiso de ubicación GPS requerido para el acceso al mismo (se trata de un permiso considerado de peligro). Por otro lado, para obtener la ubicación mediante dicho método, se estableció como restricción de hardware necesaria que el modo ahorro de batería debe estar desactivado. Esto, al igual que ocurría con el widget de cámara descrito anteriormente, es porque si dicho modo se encuentra activado asumimos que el usuario intenta ahorrar batería, por lo que realizar una tarea que tiene un gran consumo energético, cómo es obtener una ubicación mediante el GPS del dispositivo, podría acabar más rápido la batería, desfavoreciendo la experiencia de usuario.

Además, una segunda restricción de hardware fue establecida para el empleo de esta funcionalidad y es la necesidad de tener conexión a internet y que sea de tipo WI-FI. Se tomó esta decisión, en primera instancia para poder descargar el mapa donde ubicar al usuario y, se decidió permitir solo WI-FI por dos cuestiones: por un lado, para que la descarga de dicho mapa

no requiera un largo tiempo desfavoreciendo la experiencia de usuario, y, por otro lado, para que no tener el excesivo consumo de datos móviles que requiere descargar un mapa.

De esta forma, sólo podrá obtenerse la ubicación del usuario mediante el GPS del dispositivo, si se tiene conexión a internet y de tipo WI-FI y si el usuario concede el permiso de acceso a dicho GPS. Cabe señalar que, si las restricciones de hardware no se cumplen, la solicitud del permiso no se realiza, ya que de todas formas la funcionalidad no se podrá emplear.

Si estas condiciones no se dan no se podrá obtener la ubicación mediante GPS, por lo que el widget adaptable presenta una primera alternativa. En ella el usuario da su ubicación de manera manual, ubicando un punto en un mapa. Para esta alternativa no se requiere de ningún permiso, pero fueron dos establecidas las dos mismas restricciones de hardware que se describieron en la primera forma de obtener la ubicación: el dispositivo debe tener conexión a internet mediante WI-FI y debe tener desactivado el modo de ahorro de batería. Las dos restricciones de hardware se mantienen por los mismos motivos que fueron introducidas: el alto consumo de datos y tiempo que requiere descargar un mapa y no consumir demasiada batería respectivamente.

En esta segunda alternativa se mostrará al usuario el mapa descargado donde el usuario podrá situarse, el punto seleccionado se tomará como la ubicación del usuario. Así, se presenta una alternativa muy similar para cumplir la misma función, pero sin la necesidad del acceso al GPS, ya sea porque el usuario prefiere proteger su privacidad como si no le funciona correctamente.

Finalmente, si no fue posible mostrar la alternativa anterior al usuario, porque algunas de las restricciones de hardware no se cumplieron, se le presentará una tercera alternativa que es la configurada por defecto, es decir, no posee ninguna restricción de hardware ni requiere solicitar ningún permiso. Esta forma de cargar la ubicación consiste en agregar la misma de manera textual.

Nuevamente, tenemos tres formas para obtener la ubicación según las condiciones de hardware del dispositivo y de los permisos que desee otorgar a la aplicación. Nos aseguramos de esta tarea que el usuario pueda completar el caso de uso, a través de algunas de las formas alternativas que le presentamos.

El circuito de funcionamiento del widget se puede ver en la *Figura 5*.

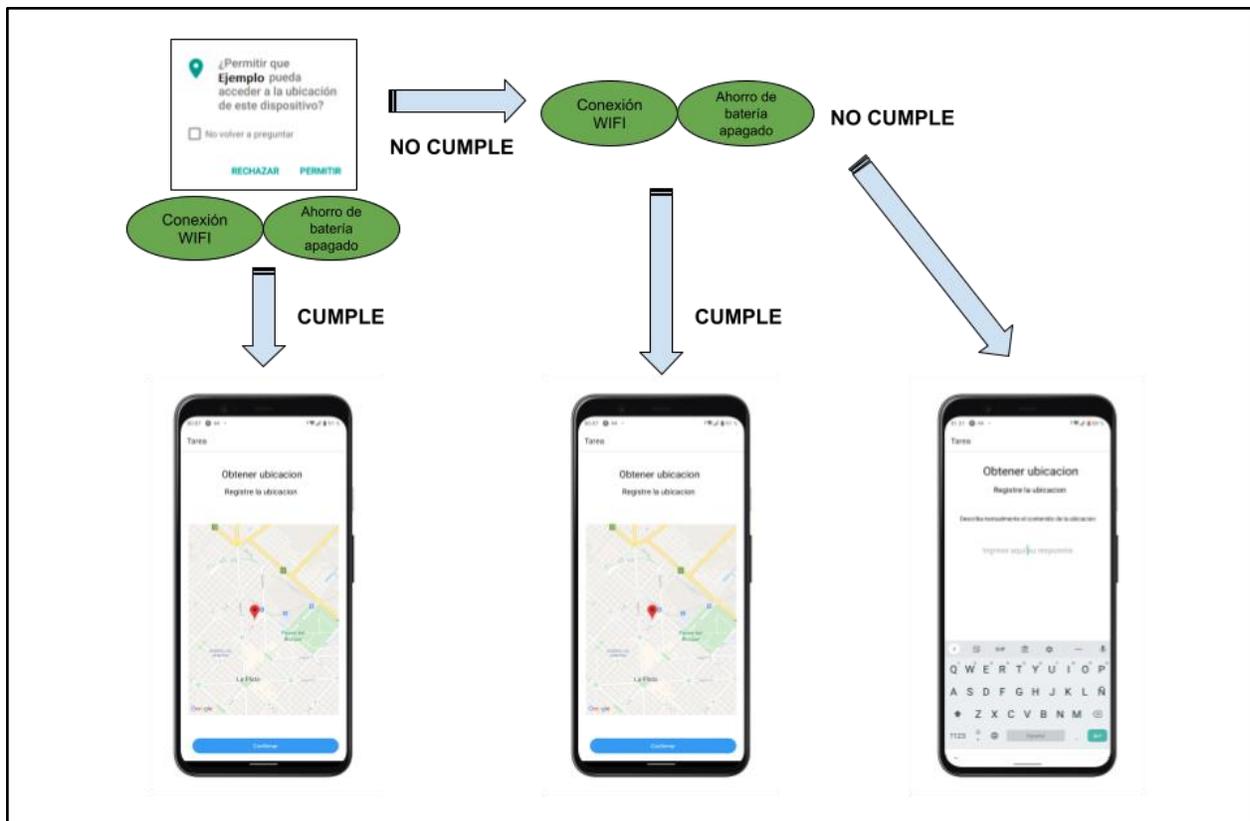


Figura 5 - Comportamiento del widget adaptable para la obtención de una ubicación

La *Tabla 5* resume las tres diferentes alternativas que el widget presenta al usuario para obtener la acción de obtener su ubicación, según las restricciones de hardware del dispositivo y los permisos otorgados a la aplicación:

Tabla 5 – Resumen de las distintas formas que provee el widget adaptable creado para obtener una ubicación

Acción: Obtener una ubicación				
Forma de llevar adelante la acción	Permisos necesarios	Conexión a internet requerida	Porcentaje de batería requerido	Modo de ahorro de batería desactivado
Obtener la ubicación mediante el uso del GPS del dispositivo	Acceso a la ubicación	WI-FI	NO	SI
Obtener la ubicación de forma manual en un mapa	Ninguno	WI-FI	NO	SI
Obtener una ubicación en forma textual	Componente por defecto. No requiere ni permiso ni establece restricción de hardware			

Como ya se mencionó, este widget fue presentado en (Di Claudio et al., 2020), sin embargo, fue mejorado al introducir la nueva versión del componente. Ahora, las alternativas para obtener la ubicación mediante GPS o de forma manual, requieren que el modo ahorro de batería esté desactivado, en lugar de requerir un porcentaje de batería específico como lo hacía en dicha ocasión.

4.4.3 Widget adaptable para la obtención de un video

El widget adaptable para la obtención de un video fue desarrollado íntegramente en esta tesina y presenta tres formas de realizar la tarea. El widget necesita de permisos de acceso a la cámara y micrófono y de un nivel de batería relativamente para poder grabar un video, por lo que, ante esto, dos alternativas para completar el caso de uso

Para grabar un video se debe acceder a la cámara y el micrófono, por lo que el widget requiere el permiso de acceso a dichos recursos para completar la acción. Por otro lado, y debido al alto consumo de batería que puede provocar grabar un video, se decidió considerar dos restricciones de hardware en lo que a batería se refiere: que el modo ahorro de batería se encuentre desactivado y que el nivel de batería sea mayor al 25%. El primero se agregó por las mismas razones que en los widgets anteriores, el hecho de tener activado este modo nos indica qué usuario intenta ahorrar energía, por lo que grabar un video en tales condiciones sería contradictorio para la experiencia del usuario. Por otro lado, se decidió fijar un porcentaje de batería, el 25%, para asegurarnos, por un lado, de que el usuario no se quede sin batería en mitad del video y, por otro, que el usuario no disponga al finalizar la grabación de un nivel de batería muy inferior, lo que puede repercutir en la experiencia de usuario final del mismo.

De este modo, si se concede el permiso de acceso a la cámara y al micrófono, el porcentaje de batería es mayor al 25% y el dispositivo no posee el modo ahorro de energía activado, el widget abrirá la cámara del dispositivo para tomar la grabación solicitada. En caso de que dicho permiso no se conceda o las restricciones de hardware no se cumplan, el widget presenta una segunda forma para obtener la grabación.

Esta segunda forma, alternativa de la primera, consiste en obtener un video del almacenamiento del dispositivo, es decir, que haya grabado con anterioridad y se guardó en el mismo dispositivo. Esta alternativa, no requiere de ninguna restricción de hardware, sin embargo, si necesita el permiso de acceso al almacenamiento en modo lectura. Si el usuario concede tal permiso, la aplicación mostrará un listado de los videos que el usuario disponga en su galería, pudiendo seleccionar uno para subirlo a la aplicación.

En caso de no otorgar el permiso necesario para acceder al almacenamiento, la forma de obtener el video detallada en el párrafo anterior no podrá ser mostrada al usuario. Con el fin de que el usuario no se quede completar el caso de uso, la alternativa por defecto configurada en

este widget muestra un cuadro de texto, donde el usuario puede describir textualmente lo que el video debería mostrar.

Por lo tanto, este widget implementa tres formas de obtener un video, que será la mostrada al usuario según los permisos que éste conceda, el porcentaje de batería y la condición de activación del modo ahorro de energía. Se pretende de esta manera mejorar la experiencia de usuario y siempre permitir completar la acción, aunque sea de formas alternativas.

El circuito de funcionamiento del widget se puede ver en la *Figura 6*.

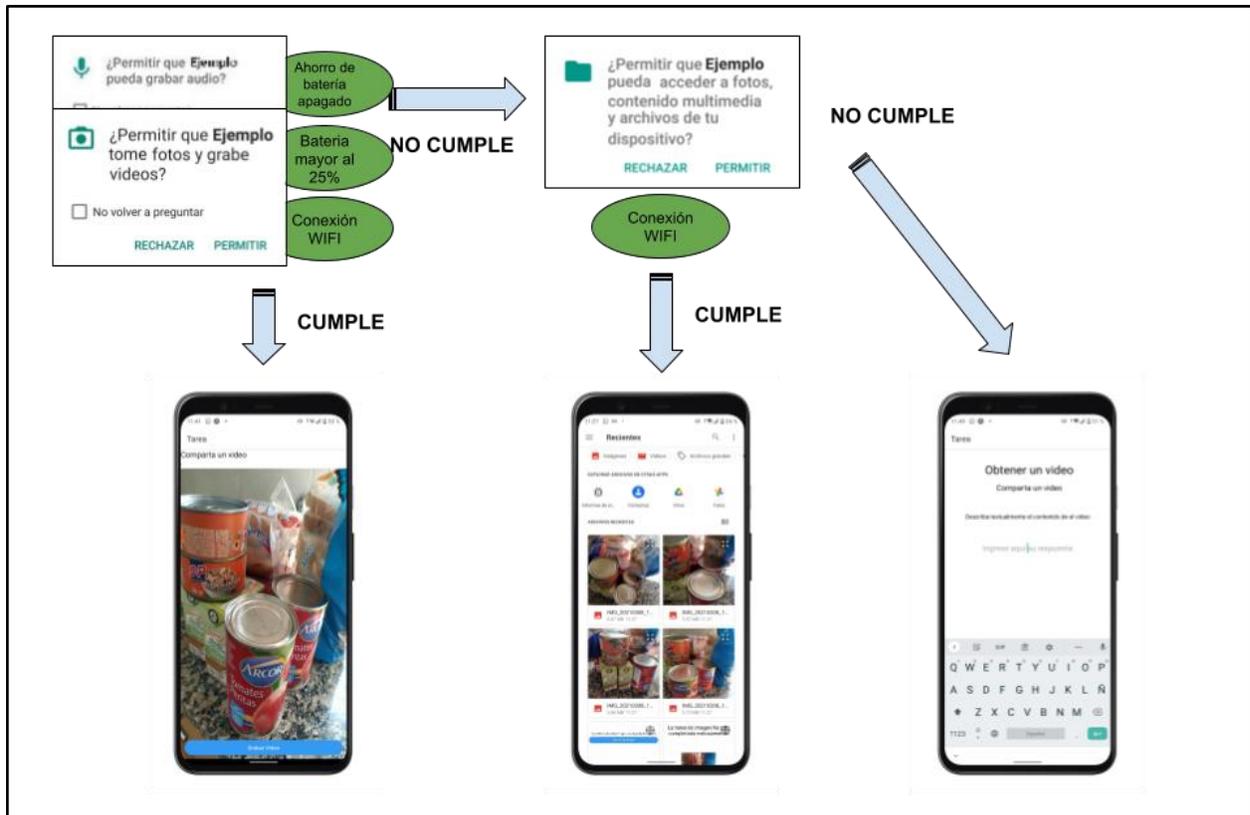


Figura 6 - Comportamiento del widget adaptable para la obtención de un video

Se resumen en la *Tabla 6* las diferentes alternativas al usuario para la acción de obtener un video, explicadas en los anteriores párrafos, con sus respectivas condiciones (permisos y restricciones de hardware), tal como fue codificada en la instanciación del componente para la creación de widgets adaptables.

Tabla 6 – Resumen de las distintas formas que provee el widget adaptable creado para obtener un video

Acción: Obtener un video				
Forma de llevar adelante la acción	Permisos necesarios	Conexión a internet requerida	Porcentaje de batería requerido	Modo de ahorro de batería desactivado
Captura de video mediante la cámara	Acceso a la cámara y al microfono	NO	25%	SI
Obtener el video mediante el almacenamiento interno del dispositivo	Acceso al almacenamiento interno	NO	NO	NO
Obtener una descripción textual del contenido del video	Componente por defecto. No requiere ni permiso ni establece restricción de hardware			

Este widget fue desarrollado en esta tesina, como parte de la extensión de “Resuelvo Explorando” que se detalla más adelante.

4.4.4 Widget adaptable para la obtención de una grabación de audio

Este último widget adaptable presentado, tiene como objetivo obtener un archivo de audio grabado para el usuario. Para ello la aplicación necesita contar con el permiso de acceso al micrófono. Si no es posible obtener este, el widget presenta una alternativa para que dicho caso de uso pueda completarse de una manera alterna.

En primera instancia se intentará mostrar al usuario la interfaz para grabar un audio. Dicha funcionalidad no requiere de ninguna restricción de hardware, ya que, no consideramos la grabación de audio como una tarea que requiera acceso a internet ni que consuma gran cantidad de energía que pueda bajar el nivel de batería, tanto que afecte la satisfacción del usuario. Si es necesario, que la aplicación que implemente este widget adaptable disponga del permiso de acceso al micrófono (considerado como de peligro por los sistemas operativos). Se le preguntará al usuario si este desea concederlo, y en caso de que esto ocurra se podrá mostrar la interfaz de grabación. En ella el usuario podrá grabar un audio, escucharlo y regrabarlo si lo considere necesario antes de compartirlo.

Si el usuario no concede el permiso de acceso al micrófono, el audio no podrá ser grabado. Como alternativa, se muestra una interfaz que consiste en un cuadro de texto donde el usuario puede transcribir aquello que debería contener el audio. Si bien esta tarea puede resultar

tediosa, y, lo más probable, es que le lleve más tiempo, evita que no se pueda completar el caso de uso. Disponemos así de una alternativa que no requiere permisos, ni debe cumplir con restricciones de hardware.

Este widget, que resulta ser el más sencillo de los aquí presentados, dispone de dos formas distintas de cargar un mismo dato, en primer caso en formato de audio y una alternativa en forma de texto.

El circuito de funcionamiento del widget se puede ver en la *Figura 7*.

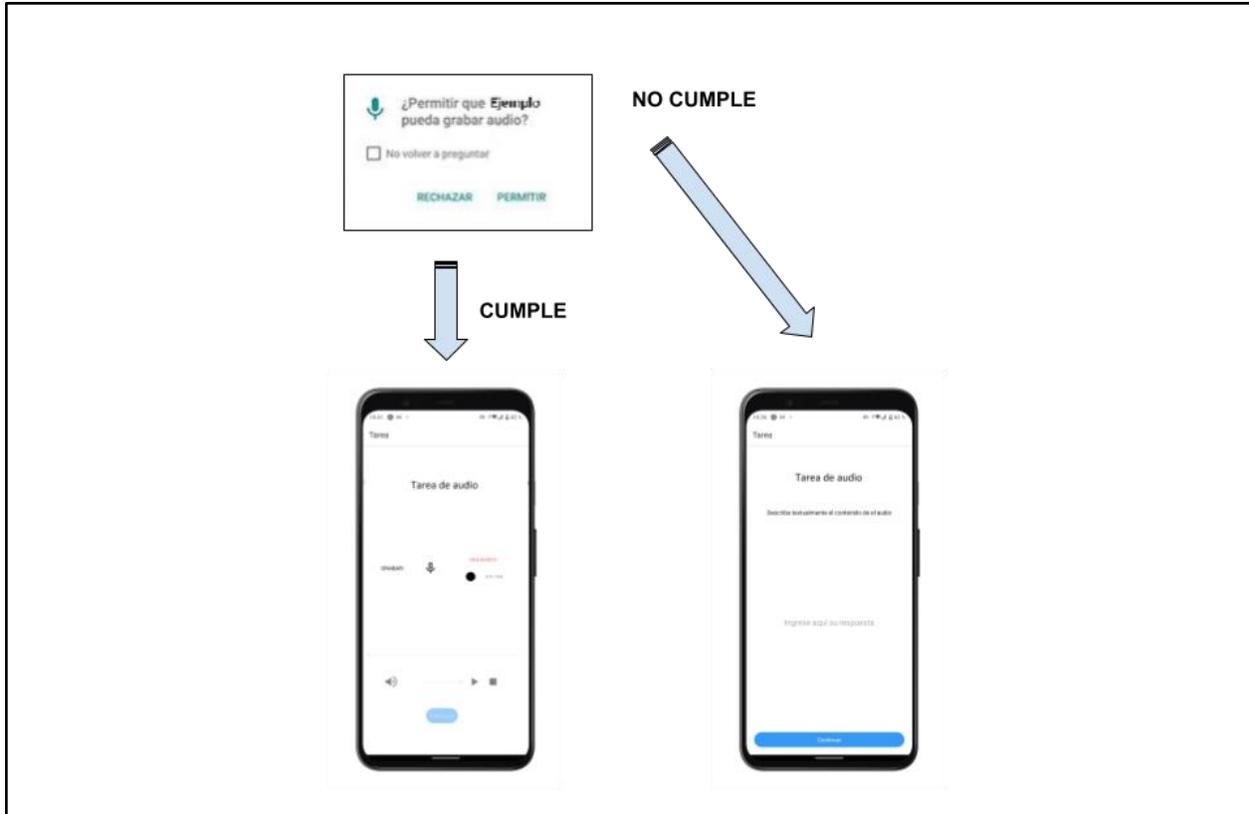


Figura 7 - Comportamiento del widget adaptable para la obtención de un audio

La *Tabla 7* muestra un resumen de las restricciones de hardware y permisos necesarios para cada una de las alternativas al usuario que presenta este widget adaptable.

Tabla 7– Resumen de las distintas formas que provee el widget adaptable creado para obtener un audio

Acción: Obtener un audio				
Forma de llevar adelante la acción	Permisos necesarios	Conexión a internet requerida	Porcentaje de batería requerido	Modo de ahorro de batería desactivado
Grabar un audio mediante el micrófono	Acceso al micrófono	NO	NO	NO
Obtener una descripción textual del contenido del video	Componente por defecto. No requiere ni permiso ni establece restricción de hardware			

4.5 “RESUELVO EXPLORANDO”: APLICACIÓN DE RECOLECCIÓN DE DATOS PARA CIENCIA CIUDADANA

Se presenta en esta sección, una evolución de la aplicación “Resuelvo Explorando”, presentada en (Di Claudio et al., 2020), para la recolección de datos en proyectos de Ciencia Ciudadana, configurable y reutilizables. La aplicación se caracteriza por mostrar una respuesta frente a: restricciones de hardware, como el estado de conectividad, el porcentaje de batería y la condición de activación del modo ahorro de batería; y ante la falta de permisos otorgados por el usuario a la aplicación, como lo son el acceso a la cámara, al almacenamiento, a la ubicación, al micrófono, etc. En la *Figura 8* se muestra la pantalla de bienvenida de la aplicación, configurada para una actividad de prueba.



Figura 8 - Pantalla de bienvenida de la aplicación “Resuelvo Explorando” configurado en una actividad de prueba

Esta nueva versión de la herramienta, implementa, a través de la integración de la extensión del componente para la creación de widgets adaptables realizado y presentada en la sección anterior, nuevos widgets adaptables en casos de uso que la aplicación ya disponía. Además, se mejoraron los widgets existentes, que fueron presentados en (Di Claudio et al., 2020), y se realizaron tareas de refactoring y mejoras en el código.

“Resuelvo Explorando” tiene como principal objetivo la recolección de datos en proyectos de Ciencia Ciudadana. Dicho proyecto puede ser configurado íntegramente desde una herramienta web llamada “MoLE”, presentada en (Lliteras et al., 2019). En ella, el jefe o encargado de proyecto definirá tanto los datos referentes al proyecto como cada una de las tareas a realizar por el voluntario en la toma de una muestra. La misma aplicación genera un código, y una versión QR del mismo, que el dicho responsable puede compartir para que voluntarios se unan desde la aplicación móvil “Resuelvo Explorando”.

Cada uno de estos proyectos, dentro del ecosistema que conforman “Resuelvo Explorando” y “Mole”, son llamados “Actividades” y poseen un título y descripción a completar por quien define dicha actividad, como carta de presentación del proyecto a los potenciales voluntarios. Además, se define en cada actividad un conjunto de misiones llamadas “Tarea”, que consisten

en obtener un dato de un tipo definido y que el voluntario debe completar para recolectar una muestra del proyecto.

El jefe o responsable de proyecto puede definir diferentes tipos de tareas entre los que se encuentran: preguntas de respuesta libre, preguntas de opción múltiple (se puede configurar una o varias opciones como correctas), consulta de ubicación, y diferentes tareas multimedia como la captura de una foto, la toma de video y la grabación de un audio. Cada una de estas tareas se compone de una consigna y otros datos configurables que varían según el tipo de tarea.

La aplicación web “MoLE” guarda, mediante el uso de la API desarrollada y descrita anteriormente, la configuración del proyecto creado. Luego, mediante la lectura QR o carga manual de dicho código, el voluntario, desde la aplicación “Resuelvo Explorando”, puede descargar dicha configuración en su dispositivo móvil. De esta forma el voluntario se unió al proyecto y podrá completar las tareas referentes a este y así completar la toma de una muestra. Una vez descargado dicho proyecto, este se almacena en el dispositivo, por lo que el voluntario podrá tomar la cantidad de muestras que desee sin necesidad de volver a descargarlo.

Para comenzar una tarea, el voluntario debe leer el código QR que la inicia haciendo uso de la cámara del dispositivo. Como se explicará más adelante, desde la nueva versión presentada en este trabajo, el código de una tarea puede ser ingresado a la aplicación en forma de texto. Una vez lanzada la tarea, la aplicación muestra al usuario el widget para completar la misma incluyendo la respectiva consigna propuesta por el jefe o responsable de proyecto en la configuración. Al finalizar la tarea, correcta o incorrectamente, se vuelve al listado de tareas pendientes, para que el usuario lea y comience la siguiente tarea o finalice la muestra tomada. El voluntario en cualquier momento puede dar por finalizada su colaboración, aunque no haya completado todas las tareas.

Al finalizar, la aplicación muestra al usuario un resumen de las tareas realizadas, incluyendo el número de completadas, fallidas y no completadas. Allí se mostrará al usuario el botón para finalizar y que los datos se envíen a los jefes de proyecto.

En la versión actual de “Resuelvo Explorando”, los datos recolectados quedan en el dispositivo del voluntario. Se propone en el capítulo “*Conclusiones y Trabajos Futuros*” algunas soluciones a implementar a futuro para que dichos datos lleguen directamente a los responsables del proyecto.

4.5.1 Widgets adaptables en “Resuelvo Explorando”

En la anterior versión de “Resuelvo Explorando”, presentada en (Di Claudio et al., 2020), se incluyeron widgets adaptables, creados con el componente, para las tareas de toma de imagen y de localización. Sin embargo, otro tipo de tareas, como la toma de video o de audio, requieren el acceso a permisos y se considera conveniente establecer restricciones de hardware para mejorar la experiencia de usuario ante estos casos de uso. Por ello, se consideró para este trabajo la inclusión de nuevos widgets adaptables para las tareas mencionadas, así como también para

la lectura de actividades y tareas, acciones que también requieren de permisos que el usuario puede no querer o poder conceder. Por otro lado, y ante una nueva versión del componente para la generación de widgets adaptables, se modificaron los ya existentes para la inclusión de mejoras en los mismos.

En esta sección se presenta la adopción de cada uno de los widgets adaptables desarrollados en este trabajo y presentados en la sección anterior, para distintas la realización de distintas tareas en una nueva versión de “Resuelvo Explorando”.

Se adoptó el widget adaptable para una toma de imagen a la tarea multimedia de foto disponible en “Resuelvo Explorando”. Si bien este widget adaptable ya fue considerado en lo presentado en (Di Claudio et al., 2020), se incluyeron en esta nueva versión las mejoras introducidas en el widget adaptable de obtención de imagen descrito en la sección anterior. La mejora en este widget se introduce gracias a que el componente para la generación de los mismos, permite establecer ahora como restricción de hardware la condición de activación del modo ahorro de batería. Se estableció que este debe estar desactivado para tomar la foto con la cámara del dispositivo. Además, se aplicó refactoring sobre el código de dicho widget.

Por otro lado, el widget adaptable para la obtención de ubicación presentado en este trabajo como evolución del presentado en (Di Claudio et al., 2020), también fue incluido en esta nueva versión de “Resuelvo Explorando”. Las principales diferencias están, nuevamente, en establecer como restricción de hardware que el modo ahorro de batería se encuentre desactivado para obtener la ubicación mediante el uso del GPS del dispositivo y, por tanto, fue quitada la restricción que establecía un porcentaje mayor al 50% para su utilización. Por otro lado, también se aplicó sobre este widget refactoring sobre su código.

De esta forma se mejoraron los widgets adaptables que ya existían para obtener una mejora en la experiencia de usuario al utilizar la aplicación, adaptándose más a las preferencias del usuario, como si este desea ahorrar batería o no.

En esta nueva versión de “Resuelvo Explorando” se adoptó el widget para la obtención de un video, descrito en la sección anterior, para la realización de una tarea multimedia de toma de video. Anteriormente, el usuario solo podía completar dicha tarea si concedía a la aplicación el permiso para de acceso a la cámara y/o al micrófono para grabar el video con ella. Ahora, mediante la incorporación de este widget, si el usuario no desea o no puede otorgar el permiso de cámara y/o el microfono no ve limitada su contribución, ya que la aplicación presenta dos alternativas para completar dicha tarea: obtener un video desde el almacenamiento interno, el cual requiere el permiso de acceso al mismo, o describir en forma textual el contenido que debería contener el video.

El widget para obtener un audio, descrito en la sección anterior, también fue adoptado por la aplicación móvil “Resuelvo Explorando” en esta nueva versión para la realización de tareas multimedia de grabación de audio. De esta forma, el voluntario, al realizar una tarea de este tipo, puede no conceder el permiso de acceso al micrófono y no quedarse sin completar dicha tarea.

Se le presenta como alternativa una interfaz donde puede ingresar en forma textual lo que debería contener el audio.

De esta forma, la nueva versión de “Resuelvo Explorando” presenta cuatro widgets adaptables para la realización de los distintos tipos de tareas sobre las que se consideraron necesario incluir algún tipo de adaptabilidad como son: la obtención de una localización, la toma de una foto, la toma de un video y la grabación de un audio. Se espera que de esta forma el usuario nunca acabe sin poder completar una tarea y se pierda la muestra que este pueda proveer. También se pretende una mejora en la experiencia de usuario al completar dichas tareas.

Por otro lado, tanto la descarga de una actividad como el lanzamiento de una tarea son casos de uso que en la versión de “Resuelvo Explorando” presentada en (Di Claudio et al., 2020) requería del acceso a cámara para su realización, no presentando ningún tipo de adaptación. En este trabajo fue identificada dicha problemática y se presenta una modificación del widget para la obtención de una imagen presentado con anterioridad y utilizado para la realización de una tarea de toma de foto para subsanar dicha problemática.

El widget de obtención de una imagen se modificó para emplear la cámara con una funcionalidad distinta, como lector de códigos QR. El código para dicha función ya se encontraba disponible y en funcionamiento dentro de “Resuelvo Explorando”. Como única alternativa a leer el código QR con la cámara, el usuario puede ingresar en forma de texto el código de la actividad o tarea a leer mediante QR. Esta alternativa fue configurada por defecto, por lo que no requiere ningún permiso ni se establece ninguna restricción de hardware, ingresar en forma de texto el código de la actividad o tarea a leer mediante QR. De esta forma, si el usuario no desea otorgar el permiso de cámara en dichos casos de uso puede de igual manera, a través del código correspondiente, descargar una actividad o lanzar una tarea. La alternativa que consiste en obtener la imagen mediante el almacenamiento interno, fue descartada para esta modificación, debido a que no se consideró de utilidad.

Por lo tanto, y además de la adopción de los nuevos widgets adaptables, se reutilizó en “Resuelvo Explorando” de manera parcial el widget adaptable para la obtención de una imagen y la funcionalidad que poseía la aplicación para incluir widgets adaptables en la descarga de actividades y en el lanzamiento de tareas.

4.5.2 Reutilización de componentes en “Resuelvo Explorando”

La nueva versión de “Resuelvo Explorando” emplea estrategias de reutilización en su implementación. Concretamente reutiliza componentes y widgets adaptables para distintos casos de uso. Esta capacidad de reutilización es en parte facilitada por el componente para la creación de widgets adaptables creado, donde establecemos libremente componentes a renderizar ante ciertas condiciones de hardware del dispositivo y ante ciertos permisos concedidos por el usuario.

La reutilización de widgets adaptables se pudo ver, tal como fue mencionada, en la descarga de actividades y en el lanzamiento de tareas. Se cambió la funcionalidad de la cámara para, en lugar de tomar una fotografía, detectar y leer un código QR y se eliminó la alternativa de obtener la imagen mediante el almacenamiento interno, debido a que se consideró que no es necesaria para este caso. Esta modificación fue reutilizada para los dos casos de uso propuestos, la descarga de una actividad y el lanzamiento de una tarea por lo que allí también existe una reutilización de widgets adaptables.

Un segundo caso donde se aplicó reutilización de componentes se puede encontrar en los componentes por defecto de los widgets adaptables presentado en la sección 4.4 y adoptados por “Resuelvo Explorando”. La mayoría de dichos widgets adaptables emplean como componente por defecto un cuadro de texto que el usuario debe completar. Se empleó el mismo componente para estos widgets, por lo que se muestra al usuario la misma interfaz, pero adaptada al tipo de tarea que el usuario realiza en ese momento.

Estos casos demuestran que el componente nos permite también la fácil reutilización de componentes para distintos widgets, así como también que los widgets desarrollados con dicho componente pueden adaptarse a distintos casos de uso dentro de una misma aplicación, lo que también reduce los tiempos y esfuerzos en el desarrollo.

5 CASO DE ESTUDIO

En este capítulo, se presenta un caso de estudio, donde se configura desde la herramienta “MoLE” (Llitas et al., 2019), la aplicación “Resuelvo Explorando” mejorada con la adopción de nuevos widgets adaptables, descrita anteriormente, para recolectar datos pertenecientes a una donación al Banco de Alimentos de La Plata⁵¹.

El Banco de Alimentos de La Plata se trata de una Organización de la Sociedad Civil (OSC), sin fines de lucro, que tiene como objetivo *“disminuir el hambre, la desnutrición y las malas prácticas alimentarias de la región, mediante el recupero de alimentos, para ser distribuidos a organizaciones comunitarias que prestan servicio alimentario a sectores necesitados”*, esto según indica su misión⁵². La organización cuenta con diferentes mecanismos de aporte, siendo uno de ellos las donaciones que realizan particulares, empresas y productores. El caso de estudio aquí presentado, instancia en “MoLE” todas las tareas de carga de la información requerida para realizar una de estas donaciones, para que luego sea descargada y completada mediante la nueva versión de “Resuelvo Explorando” presentada en este trabajo.

Con este caso de estudio se busca comprobar si los widgets adaptables creados e integrados en las diferentes tareas de “Resuelvo Explorando”, mejoran la capacidad para llevar adelante las correspondientes tareas a pesar de las restricciones de hardware del dispositivo y la falta de permisos concedidos por el usuario

Si bien el caso de estudio trasciende el campo donde va dirigida esta tesis, la Ciencia Ciudadana, abordando este dominio en particular se quiere mostrar el potencial de esta evolución de la aplicación “Resuelvo Explorando”, como parte de un proyecto que también posee un impacto social.

Dada la relevancia que posee cada donación, resulta muy importante que cuando un individuo o empresa decide realizar una donación, los datos de ella puedan ser cargados de la manera más sencilla y con la mejor experiencia de usuario posible para el donante, sin que se vea en ningún momento para este complejizada la tarea por la tecnología, sino que, por el contrario, la misma favorezca y estimule su participación.

El caso ha sido diseñado en base a la información existente en la página web oficial de la organización y considerando la posible participación de particulares, empresas y productores. Cuando un donante quiere registrar una donación, indica a cuál de las tres categorías antes mencionadas pertenece, pasa a indicar sus datos personales y luego indica si se trata de alimentos perecederos o no perecederos. Además, se deben ingresar los datos relativos al contenido a donar en sí, que consisten en una foto o un video del bulto de productos a donar

⁵¹ <https://bancoalimentario.org.ar/>

⁵² <https://bancoalimentario.org.ar/quienes-somos/#mision-vision-valores>

(esto para ayudar a tomar una dimensión del volumen de la donación), un audio donde el donante describa con mayor detalle las cantidades de cada producto y, por último, la ubicación donde se encuentran dicho paquete a donar.

Con este simple circuito de carga descrito, abarcamos el uso de todos los widgets adaptables que presenta la nueva versión de “Resuelvo Explorando”, ya sea los desarrollados con anterioridad (para evaluar las mejoras realizadas) como los presentados en esta tesina. De esta forma se pretende evaluar su correcto funcionamiento y exponer la aplicación a los posibles desafíos para los que fue preparada. Por ejemplo, al realizar la carga del video que se solicita, el usuario puede no conceder permiso de cámara a la aplicación para grabar dicho video, en tal caso no se quiere imposibilitar a este a cargar el dato solicitado, ya que se trata de un dato importante para obtener una dimensión del volumen de la donación. La aplicación “Resuelvo Explorando” debería poder adaptarse de manera transparente al usuario para que este no vea imposibilita la acción, ofreciéndole un modo alternativo de cargar el mismo dato. El mismo caso aplica ante una restricción de hardware que el dispositivo no cumpla y que sea necesario cumplir.

5.1 ESTRUCTURA DEL CASO DE ESTUDIO

En esta sección se presenta con más detalle el caso de estudio propuesto en este capítulo. El mismo será instanciado con la aplicación web “MoLE”, presentada en (Llitas et al., 2019) y luego descargado en la nueva versión de “Resuelvo Explorando”.

Para la descarga de un proyecto en “Resuelvo Explorando”, el usuario puede hacerlo mediante un código QR que genera la propia aplicación “MoLE”, si es que da el permiso de cámara para realizar dicha acción, tal como se muestra en la *Figura 9* o, en caso de que no se conceda dicho permiso puede ingresando un código propio del proyecto que también se muestra en “MoLE”. Encontramos acá una primera adaptación de la aplicación “Resuelvo Explorando”, en este caso para identificar el proyecto a descargar.



Figura 9 - Captura de código QR de la actividad correspondiente al caso de prueba

El proyecto descargado consiste en una serie de tareas de carga de datos para el registro de una donación para identificar al donante, como también obtener los datos pertenecientes a los productos a donar. Las tareas a realizar son las siguientes:

1. Una tarea de tipo pregunta de opción múltiple para saber si el donante se trata de un particular, una empresa o un productor.
2. Una tarea de tipo respuesta libre para indicar el nombre y apellido del particular o productor o bien el nombre de la empresa donante, además de poder incluir aquí información de contacto.
3. Una tarea de tipo pregunta de opción múltiple donde se indicará si se trata de alimentos perecederos o no perecederos.
4. Una tarea de tipo foto que consiste en fotografiar el conjunto de productos destinados a donar para dar una idea del tamaño del mismo.
5. Una tarea de tipo video, donde se debe grabar el conjunto de productos destinados a donar para dar una idea del tamaño del mismo.
6. Una tarea de tipo audio, donde el donante deberá describir los productos involucrados en la donación con sus respectivas cantidades
7. Una tarea de tipo ubicación, donde se indicará el lugar en el cual retirar los productos donados.

En la *Figura 10* se puede observar el listado de tareas que se le muestra al participante de la prueba luego una vez iniciada la actividad.

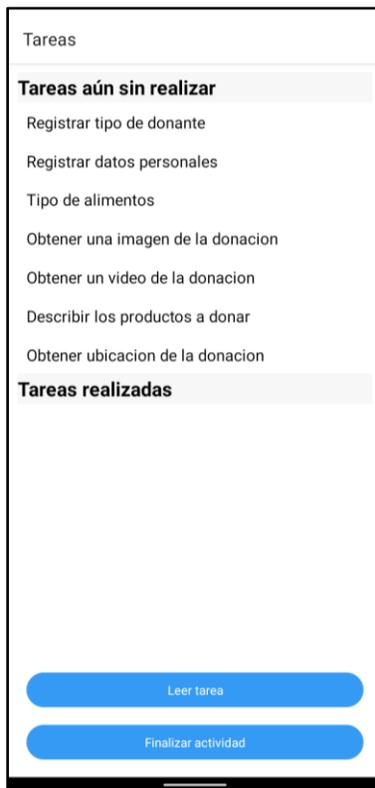


Figura 10 - Listado de tareas del caso de prueba en “Resuelvo Explorando”

El caso de estudio contiene las siguientes tareas para que el Banco de Alimentos de La Plata obtenga toda la información tanto del donante como de la donación, incluyendo el listado de productos y la ubicación de los mismos. Además, desde el punto de vista tecnológico, el caso de estudio contiene un conjunto de tareas que requieren la utilización de todos los widgets adaptables creados y detallados en el capítulo anterior, lo que nos permite comprobar si los mismos logran saltar las limitaciones propuestas en un principio.

A continuación, se detalla con mayor profundidad las tareas en las que consiste el caso de estudio presentado, además del comportamiento de la aplicación “Resuelvo Explorando” en su nueva versión, ante la carga de los datos requeridos por cada tarea.

La primera tarea consiste en una pregunta de opción múltiple para indicar si el donante se trata de un particular, un productor o una empresa (los tres tipos de donantes que tiene el Banco de Alimentos). La realización de dicha tarea no requiere el acceso a ningún recurso del dispositivo por lo que requiere la petición de ningún tipo de permiso al usuario, ni se consideró necesario establecer alguna restricción de hardware para mostrar la interfaz de este tipo de tareas, debido a su sencillez y que no requiere el uso de conectividad. La interfaz resultante en esta tarea puede verse en la *Figura 11*.

Tarea

Registrar tipo de donante

Particular

Empresa

Productor

Finalizar

Figura 11 - Tarea de opción múltiple para registrar el tipo de donante

Una situación similar ocurre en la tarea número tres del listado anterior, ya que también es de tipo pregunta de opción múltiple. En este caso consiste en indicar si los productos a donar en el bulto son perecederos, no perecederos o contiene ambos. La tarea puede verse en la *Figura 12*.

Tarea

Tipo de alimentos

Perecederos

No perecederos

Finalizar

Figura 12 - Tarea de opción múltiple para indicar el tipo de alimentos a donar

Para informar sobre los datos personales del donante, como el nombre y apellido y la información de contacto, se presenta un campo de texto libre que puede describir textualmente los datos que considere necesarios. Este tipo de tareas de texto libre, tampoco requieren ningún tipo de adaptación, por lo que pueden completarse sin problemas, como se muestra en la *Figura 13*.

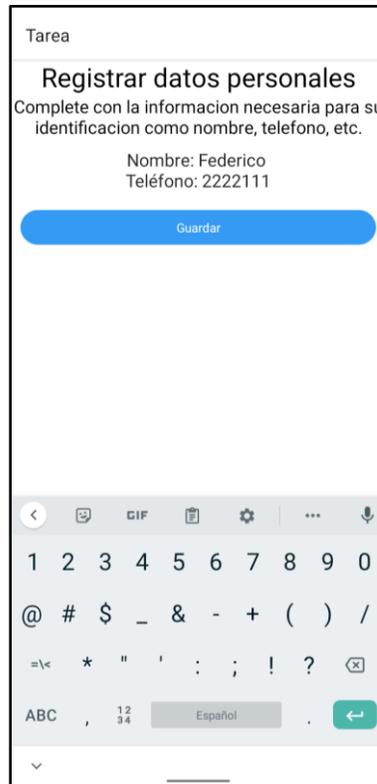


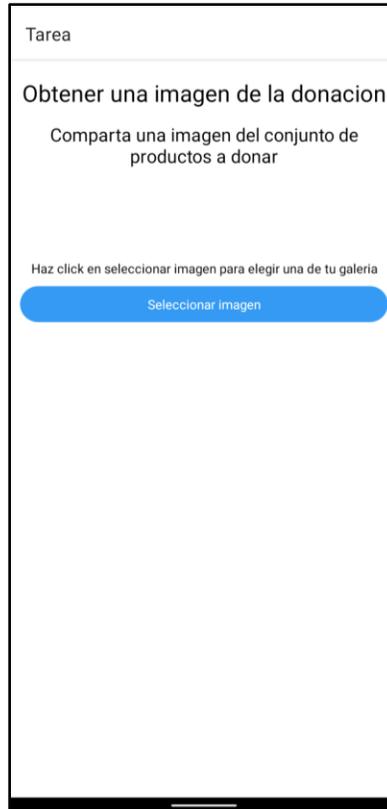
Figura 13 - Tarea de respuesta libre para la obtención de datos del donante

La cuarta tarea consiste en una tarea multimedia de toma de imagen. En ella el usuario debe compartir una imagen con el conjunto de productos que va a donar al banco de alimentos, con el fin de que los administradores tomen una dimensión del tamaño del bulto total a donar. Para tomar la fotografía, el usuario debe conceder el permiso de cámara y, además, debe tener el modo ahorro de batería desactivado. Si este último está activado, no se accederá a la cámara del dispositivo, ya que consideramos que requeriría un consumo de energía que implícitamente el usuario no desea. En caso de que este modo se encuentre desactivado, la aplicación solicitará al usuario el permiso correspondiente para acceder a la cámara, y el usuario debe aceptar el cuadro de diálogo que se le presenta, tal como se muestra en la *Figura 14a*. En caso de que no desee brindarlo, simplemente porque no lo desea o porque su cámara se encuentra en mal estado o no funciona, se le solicitará el permiso para acceder al almacenamiento interno del dispositivo. En caso de que lo conceda, el usuario podrá subir una imagen que tenga en el almacenamiento interno del dispositivo que haya sido tomada con anterioridad; el caso se muestra en la *Figura 14b*. Finalmente, en caso de no conceder el anterior permiso solicitado, se le muestra una última alternativa, que no requiere ningún permiso del usuario y consiste en cuadro de texto, donde el usuario detalle en forma escrita lo que no pudo mostrar en forma de imagen; esto último se muestra en la *Figura 14c*. De esta forma, el usuario siempre encuentra alguna alternativa para realizar su tarea.



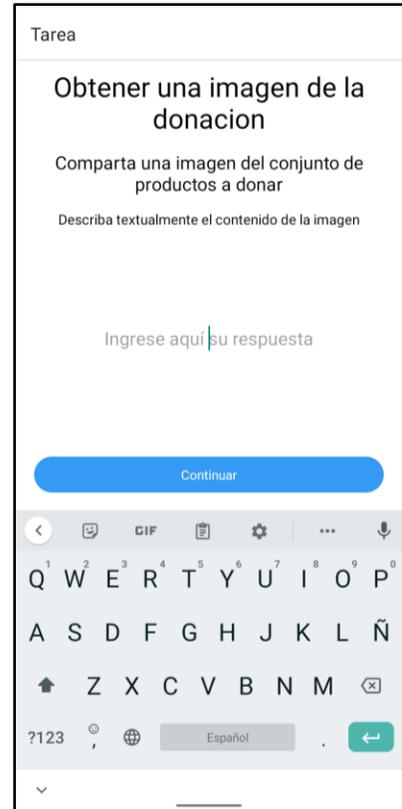
(c)

Interfaz para capturar una foto mediante la cámara



(b)

Interfaz de obtención de imagen mediante la búsqueda en el almacenamiento interno



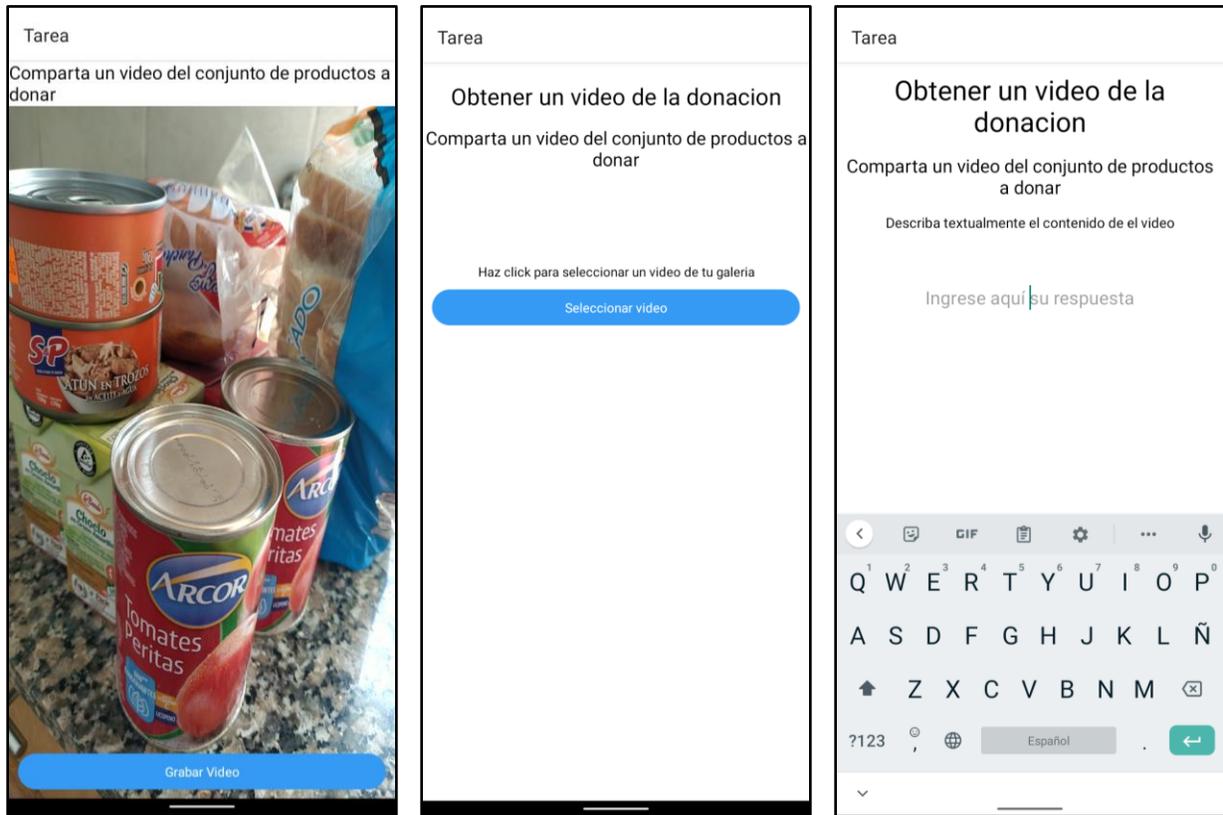
(c)

Interfaz de descripción textual de una tarea de imagen

Figura 14 - Alternativas para la obtención de una imagen

Una situación similar a lo descrito en el párrafo anterior ocurre al realizar la tarea de tipo multimedia que consiste en capturar un video, donde es necesario el permiso de acceso a la cámara y el de acceso al micrófono, en este caso mucho más, el modo ahorro de energía debe estar desactivado, ya que la tarea de grabar un video consume una mayor batería que otros tipos de tareas. Con el fin de que el dispositivo no se quede sin batería durante la grabación del video, lo cual resultaría en una experiencia de usuario desfavorable, el dispositivo debe disponer de un porcentaje de batería mayor al 20%. Por último, el dispositivo debe estar conectado a una red WI-FI para realizar la carga del video. Esta situación se muestra en la captura de pantalla presentada en *Figura 15a*. En caso de que no se otorgue el permiso correspondiente o que no se cumplan algunas de las restricciones establecidas, se solicitará al usuario, al igual que ocurría al tomar una tarea de tipo imagen, el permiso de acceso al almacenamiento del dispositivo. De esta manera el usuario podrá cargar un video que ya disponga en su dispositivo; esto no requiere el acceso a la cámara ni tiene el mismo consumo energético que la primera alternativa y resulta en el mismo resultado. Esta segunda alternativa se muestra en la *Figura 15b*. Finalmente, una última alternativa que se mostrará es un cuadro de texto, donde el usuario describa la donación, tal

como se muestra en la Figura 15c. Esta última opción, no requiere de ningún permiso para su funcionamiento ni de ninguna restricción de hardware, asegurando que siempre el donante pueda seguir adelante con su acción.



(a)
Interfaz de grabación de video

(b)
Interfaz de obtención de video mediante almacenamiento interno

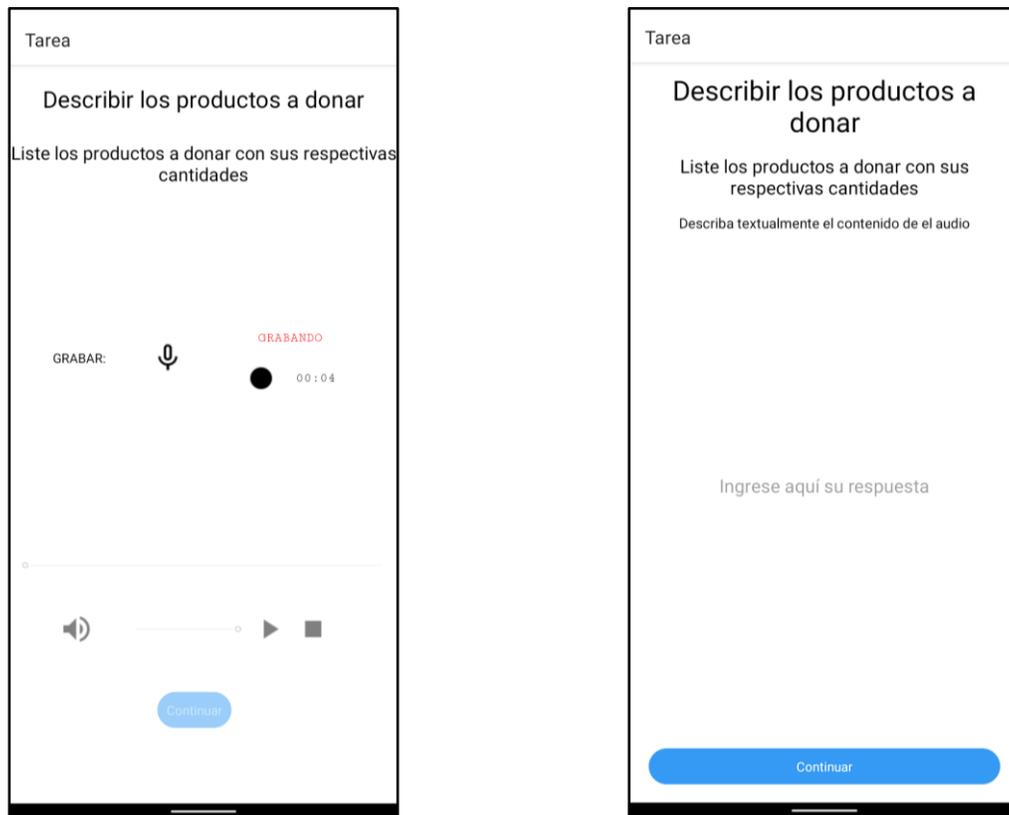
(c)
Interfaz de descripción textual de una tarea de video

Figura 15 - Alternativas para la obtención de un video

Tanto la tarea de obtención de imagen como la de obtención de video tienen el mismo objetivo, dar una idea de la dimensión de los productos a donar, que luego deben ser retirados por el Banco de Alimentos. Realizar ambas tareas no se considera necesario, por lo que, quien lleve adelante el caso de prueba puede elegir realizar una, otra o ambas si así lo desea.

La siguiente tarea establecida se trata de una tarea multimedia de tipo audio, en la que el usuario, mediante dicho formato, debe describir verbalmente el contenido del bulto a donar, esto incluye los productos y sus respectivas cantidades. En una primera instancia la aplicación solicitará al usuario el permiso para acceder al micrófono, requerido para grabar el archivo de audio. Si el usuario concede dicho permiso se mostrará, tal como se muestra en la *Figura 16a*, una interfaz para grabar audio. En dicha interfaz el usuario puede reproducir y volver a grabar su audio, en caso de que lo considere necesario, antes de completar la tarea y guardar el archivo.

En caso de que no se conceda el permiso de acceso al micrófono, se mostrará al usuario un cuadro de texto, donde escribir aquello que no detallo en forma de audio, de la manera en que se muestra en la captura de la *Figura 16b*. En este caso, se presenta al usuario dos alternativas para realizar una misma acción, la de describir los productos y sus respectivas cantidades, en forma de audio o de texto.



(a)

Interfaz para la grabación de un archivo de audio

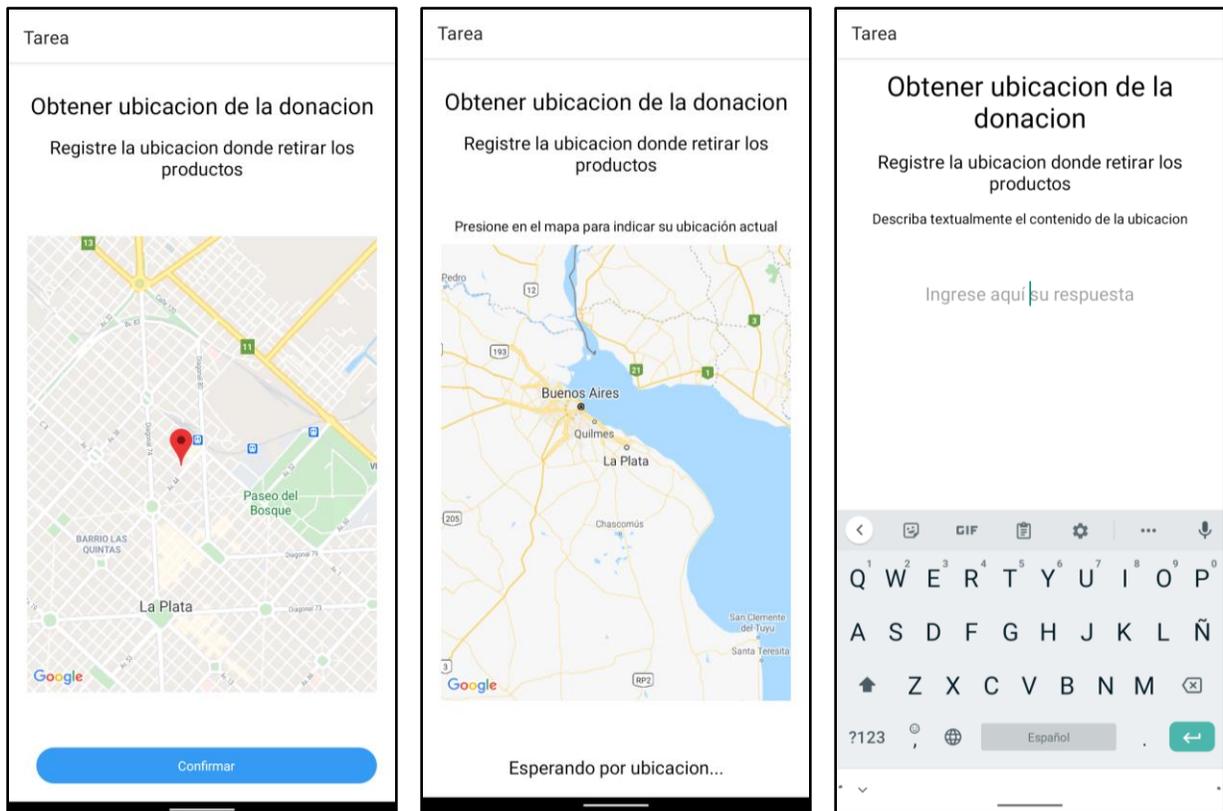
(b)

Interfaz para completar la tarea de audio mediante una descripción textual

Figura 16 - Alternativas para la obtención de un audio

Por último, la actividad de toma de ubicación tiene el objetivo de indicar la dirección donde se encuentra el bulto a donar (se asume un uso in-situ). Para tomar dicha ubicación el usuario dispone de tres alternativas tal como fue establecido el widget correspondiente. Una primera alternativa, la cual puede ver en la *Figura 17a*, se mostrará al usuario en caso de que este no disponga del modo ahorro de batería activado, debido a que puede consumir una gran cantidad de batería del dispositivo, que esté conectado a una red WI-FI, debido a que la tarea requiere cargar mapas y no consuma muchos datos móviles o tome un tiempo excesivo de carga, y que se conceda a la aplicación el permiso de acceso al GPS. Si dicho permiso no se concede, se mostrará

una interfaz, la de la *Figura 17b*, que considera las mismas restricciones de hardware, pero no necesita del GPS ya que el usuario ubicará manualmente el punto en que se encuentra. Una tercera alternativa que se mostrará en caso de que no se cumpla las condiciones para mostrar la anterior, se trata de un cuadro de texto donde el usuario ingresará manualmente la dirección de los productos, tal como se muestra en la *Figura 17c*. Nuevamente, tenemos diferentes alternativas para cargar un mismo dato, en este caso la ubicación, permitiendo al usuario que la tarea siempre se cumpla.



(a)
Interfaz para obtener la ubicación mediante el uso de GPS

(b)
Interfaz para obtener la ubicación mediante la colocación manual de un punto en el mapa

(c)
Interfaz para obtener la ubicación mediante una descripción textual

Figura 17 - Alternativas para la obtención de una ubicación

La realización de las tareas propuestas previamente no fue establecida bajo un orden estricto, sino que, por el contrario, el usuario puede realizar las tareas en el orden que desee. Este orden es configurable en la creación de la actividad dentro de la aplicación de creación “MoLE”, tal como se explica en (Llitas et al., 2019), siendo establecida como “Libre” para el caso propuesto en este capítulo.

Una vez completadas todas las tareas, la actividad está completa, por lo que el usuario ha finalizado la carga de datos. Actualmente, los datos recolectados quedan en el dispositivo en el cual se generaron.

5.2 PRUEBA CON USUARIOS

Una vez diseñado e instanciado (a través de “MoLE”) el caso de estudio propuesto en la sección anterior, se procedió a realizar una prueba de concepto con potenciales usuarios de la aplicación “Resuelvo Explorando”.

Se consideró importante probar la plataforma con usuarios sin conocimientos de programación ya que son el foco al que está destinado el uso de la aplicación móvil. Se quiere determinar tanto el correcto funcionamiento de la aplicación desarrollada como también el nivel de usabilidad que presenta.

Para llevar adelante la prueba, cada participante completo inicialmente una serie de datos demográficos que le fueron solicitados: edad, sexo y cantidad de horas diarias de uso de dispositivos móviles. Esto con el fin de obtener información relativa a la población sobre la que se realizó la prueba. A continuación, se le facilitó un dispositivo móvil con la aplicación “Resuelvo Explorando” ya instalada con un respectivo escenario de uso asignado. Los escenarios planteados reflejan diferentes situaciones que pueden enfrentarse los potenciales usuarios al utilizar la aplicación “Resuelvo Explorando”, que varían según se conceda o no permisos y en la condición del hardware del dispositivo. Se generaron los siguientes escenarios:

1. El usuario concede acceso pleno sobre todos los permisos solicitados y el dispositivo se encuentra conectado a internet, por medio de Wi-Fi, y posee un porcentaje de batería elevado, por encima del 50%, con el modo ahorro de batería desactivado.
2. El usuario no concede ningún permiso durante la prueba y el dispositivo no posee conexión a internet, tiene un bajo porcentaje de batería (menor al 25%) y tiene activado el modo ahorro de batería.
3. El usuario concede acceso pleno sobre todos los permisos solicitados y el dispositivo si bien posee conexión a internet mediante WI-FI, tiene el modo de ahorro de energía de batería activado.
4. El usuario concede acceso pleno sobre los permisos solicitados y el dispositivo no posee conexión a internet de ningún tipo y le queda un porcentaje alto de batería, por encima del 50%, con el modo ahorro de energía desactivado.
5. El usuario solo puede conceder permisos en dos de las cuatro tareas que lo requieren mientras que el dispositivo posee conexión a internet a través de redes WI-FI, el modo ahorro de energía desactivado y un porcentaje mayor al 50% de batería.

En la *Tabla 8* se muestra un resumen de las restricciones de hardware y permisos concedidos que se contemplaron para cada escenario de prueba.

Tabla 8 – Restricciones de hardware del dispositivo y permisos que puede conceder el usuario para los distintos casos de prueba planteados.

Restricciones y permisos	Escenario 1	Escenario 2	Escenario 3	Escenario 4	Escenario 5
Conexión WI-FI	x		x		x
Sin conexión a internet		x		x	
Porcentaje de batería por encima del 50%	x		x	x	x
Porcentaje de batería por debajo de 25%		x			
Modo ahorro de batería desactivado	x			x	x
Modo ahorro de batería activado		x	x		
Concede todos los permisos	x		x	x	
No concede ningún permiso		x			
Solo puede conceder dos permisos					x

Mediante estos casos se procura contemplar la mayoría de posibles escenarios de uso, donde por cada uno, la aplicación muestra variaciones en las formas de cargar los datos.

A cada usuario se le facilitó un dispositivo móvil con la aplicación “Resuelvo Explorando” instalada, acompañado de una guía impresa con los pasos a seguir para realizar la actividad de prueba en la aplicación. Estos deberían simular que registraban una donación al Banco de Alimentos. Las tareas que tuvieron que realizar los participantes fueron explicadas en dicha guía y consistieron en descargar la actividad creada en “MoLE” mediante el código textual o QR que le fue presentado y completar la mayor cantidad de tareas de la actividad descargada. Al finalizar su participación, se le entregó a cada usuario una encuesta que incluye un SUS (Simple Usability Scale), el cual se trata de un cuestionario simple, de diez preguntas que provee una visión global de aspectos subjetivos de la usabilidad de la aplicación dada (Brooke, 1996). Además, a cada usuario se le tomó el tiempo total del desarrollo de la prueba. Se realizaron las pruebas por separado para evitar la interferencia entre los usuarios.

5.3 CONCLUSIONES OBTENIDAS Y DISCUSIÓN

Si bien la prueba fue realizada con pocos participantes y no se incluyeron en ella usuarios habituales del Banco de Alimentos, situación que se vio limitada por la pandemia de COVID-19 existente en el momento, las pruebas realizadas sobre cinco escenarios y con cinco usuarios diferentes, uno para cada escenario, arrojan buenos resultados en términos del número de escenarios explorados y el éxito en el completado de tareas. Los resultados obtenidos pueden verse en (Di Claudio, 2021).

En las entrevistas posteriores a cada prueba, los usuarios destacaron la ventaja de no tener que brindar permisos a las aplicaciones para poder usarla, acción que en muchas otras aplicaciones imposibilitan por completo el uso de la misma. También, los participantes comentaron una mejora en la experiencia de usuario al obtener una aplicación que se adapta a ellos y no tener ellos que adaptarse a la aplicación.

Los resultados obtenidos de las pruebas, con respecto a la cantidad de tareas completadas por cada participante son las siguientes:

- Dos de los participantes completaron la totalidad de las tareas (7 de 7).
- Los restantes tres participantes consideraron que no era necesario obtener tanto la imagen como el video, por lo que dos obviaron la tarea de imagen (6 de 7) y otro la tarea de vídeo, completando todas las demás (6 de 7).
- Haciendo un conteo general de todas las tareas completadas por todos los usuarios en contraste con las que fueron obviadas (igualmente opcionales), se tiene un total de 32 sobre 35.

En resumen, todos los participantes de la prueba pudieron completar todas las tareas necesarias, ya que se propuso que realizar ambas tareas de imagen y video era opcional, siendo nada más que necesario la realización de una tarea que el participante considerase necesario (un usuario consideró sólo la tarea de imagen, mientras que otro sólo la de video). Esto indica que, si bien se trata de una prueba preliminar, se puede apreciar el éxito de los widgets adaptables diseñados y agregados a “Resuelvo Explorando”. Gracias a su incorporación, los usuarios pudieron completar, de una u otra manera, las tareas asignadas a pesar de la falta de permisos o de las restricciones de hardware que se le presentaron a muchos de ellos.

Por otro lado, a pesar de que no fue posible medir los tiempos de las tareas rigurosamente, durante las pruebas se observó un incremento en el tiempo total de aquellos usuarios que tuvieron que completar más tareas a través de los componentes por defecto, es decir, de manera textual. Esto es un resultado esperable, ya que, por ejemplo, grabar un archivo de audio requiere un menor tiempo que transcribir su contenido, u obtener una ubicación mediante GPS requiere un menor tiempo que detallar textualmente dicha ubicación.

Finalmente, por medio de los resultados del cuestionario SUS, se pueden observar resultados alentadores, ya que los participantes manifestaron en ellos que no tuvieron

dificultades para utilizar la aplicación. Los resultados del cuestionario pueden resumirse en la *Tabla 9*. Un aspecto importante a recalcar es que todos los participantes se mostraron muy de acuerdo en sentirse confiados y seguros en su uso, esto seguramente influenciado por la posibilidad de utilizar la aplicación sin necesidad de otorgar acceso a recursos privados. Dicha satisfacción se puede observar en los puntajes obtenidos mediante el cuestionario SUS, por encima de 85 puntos en todos los casos, lo cual refleja que los usuarios de prueba están satisfechos con la usabilidad de la aplicación (por encima de 80 puntos).

Tabla 9 – Puntajes obtenidos mediante la encuesta SUS por cada uno de los usuarios de prueba con su respectivo escenario asignado.

Usuario N°1	Escenario 1	92.5 puntos
Usuario N°2	Escenario 2	87,5 puntos
Usuario N°3	Escenario 3	95 puntos
Usuario N°4	Escenario 4	92,5 puntos
Usuario N°5	Escenario 5	87,5 puntos

A falta de más pruebas, podemos concluir sobre las pruebas realizadas que la usabilidad y la experiencia de usuario en esta nueva versión de “Resuelvo Explorando” resulta muy buena.

6 CONCLUSIONES Y TRABAJOS FUTUROS

Se presentó en esta tesina una extensión del componente para la creación de widgets adaptables, presentado en (Di Claudio et al., 2020), en el que se contemplan nuevas restricciones de hardware y permisos. Con dicho componente se generó una nueva versión de los widgets adaptables ya presentados en (Di Claudio et al., 2020) y se generaron nuevos para la obtención de un video y una grabación de un audio. Los nuevos widgets fueron adoptados por “Resuelvo Explorando”, una aplicación móvil de recolección de datos para Ciencia Ciudadana, resultando en una extensión de dicha aplicación también presentada en (Di Claudio et al., 2020).

Por otro lado, en el *Capítulo 3*, se presentaron diversos trabajos relacionados sobre adaptación y un relevamiento exploratorio realizado de artefactos de software utilizados para la recolección de datos de Ciencia Ciudadana.

Como resultado de la búsqueda de trabajos relacionados referentes a la adaptación, se obtuvieron diferentes artículos dentro de la computación basada en el contexto, un conjunto de aplicaciones que presentan algún tipo de adaptabilidad y dos trabajos referentes a permisos donde se busca mejorar la experiencia de usuario ante la falta de estos. Sin embargo, no se ha obtenido mediante la búsqueda ningún trabajo que contemple tanto restricciones de hardware como falta de permisos para adaptarse, ni tampoco algún componente de software que permita la instanciación de los mismos.

A continuación, se presentó el relevamiento exploratorio realizado, el cual tuvo como objetivo recopilar artefactos de software utilizados por la comunidad en proyectos de Ciencia Ciudadana para la recolección de datos. Se describió cada uno de ellos y se analizó en detalle para su comportamiento ante la falta de permisos y ante ciertas condiciones de hardware contempladas en el desarrollo de esta tesina. Se obtuvo como conclusión que la mayoría de los artefactos encontrados no presenta ningún tipo de adaptación ante la falta de permisos, dicha adaptación se encuentra presente sólo en algunos casos y de manera parcial. Con respecto a restricciones de hardware, la mayoría de los artefactos de software encontrados se adapta ante la falta de conectividad, sin embargo, deja de lado el tipo de conectividad y las condiciones de batería del dispositivo.

En el *Capítulo 4*, se presentó una extensión de la componente para la generación de widgets adaptables. En ella se consideró nuevas restricciones de hardware, como la condición de activación del modo ahorro de energía, y nuevos permisos a definir como necesarios para mostrar un componente al usuario. Además, se realizaron ciertas mejoras en el código: se aplicó refactoring sobre el código del componente y se actualizaron las dependencias utilizadas para el desarrollo del componente a las nuevas versiones disponibles, para solucionar problemas de compatibilidad existentes, lo que también acarreó modificaciones en el código.

La utilización de este componente tiene como principal objetivo facilitar al desarrollador la creación de widgets adaptables ante restricciones de hardware del dispositivo y la falta de permisos otorgados por el usuario. Además de esto, el componente provee toda la funcionalidad para solicitar permisos al usuario, así como también para consultar las condiciones de hardware y evaluarlas en tiempo de ejecución, tareas que generalmente conllevan gran cantidad de tiempo y esfuerzos a los programadores. Así, el desarrollador ve limitada su tarea a desarrollar las interfaces de las distintas alternativas con su respectiva funcionalidad, solo teniendo que definir los permisos necesarios y las condiciones de hardware a evaluar para cada alternativa.

Este componente se encuentra disponible en un repositorio de GitHub⁵³ para emplearse por cualquiera que así lo deseara. Además, se acompaña esta tesina con un anexo que explica su modo de importación y utilización en nuevos proyectos.

Por otro lado, se presentaron en esta tesina distintos widgets adaptables desarrollados con la componente: para la obtención de una imagen, de un video, de una grabación de audio y de una localización geográfica. Los widgets de obtención de imagen y de localización geográfica, se tratan de una mejora de los presentados en (Di Claudio et al., 2020). Se actualizaron estos a la nueva versión del componente y se cambiaron algunas restricciones de hardware necesarias para cada alternativa según se consideró conveniente para la mejor experiencia del usuario.

En estos widgets se pudo apreciar cómo el usuario puede realizar una misma acción, la carga de un tipo de dato, de diferentes maneras según los permisos que éste concedió a la aplicación y las restricciones de hardware propias del dispositivo al realizar dicha acción. De esta forma, el usuario nunca ve imposibilita la carga de un tipo de dato, asegurando así su contribución y la mejora de su experiencia de usuario.

Finalmente, estos widgets fueron adoptados por la aplicación móvil “Resuelvo Explorando” en una nueva versión, que resulta en ser adaptable en la realización de todos los tipos de tareas que requieran permisos otorgados por el usuario o puedan considerar restricciones de hardware para llevarse adelante. De esta forma, el voluntario que utilice la aplicación y esté preocupado por su privacidad y no quiera otorgar un permiso de, por ejemplo, acceso a la cámara, o aquel potencial voluntario que no pueda utilizar la cámara de su dispositivo, ya sea porque esté rota u otra razón, podrá completar la tarea aportando de otra manera el dato. De la misma forma, por ejemplo, un usuario que posee un nivel de batería muy bajo, no estará obligado a tomar un video con el dispositivo (tarea que requiere un alto porcentaje de batería), ya que la aplicación sabrá adaptarse para mostrar una alternativa que consuma menos batería, mejorando así la experiencia de usuario del voluntario.

Si bien el dato obtenido puede diferir en calidad dependiendo si se obtiene de una u otra forma, por ejemplo, la veracidad de la ubicación obtenida por GPS es mayor a la ubicación manual

⁵³ <https://github.com/cientopolis/mutableWidget>

por medio de un mapa. Dependerá del proyecto, así como de la consideración de los responsables del mismo si dicho dato puede ser incluido o no para su trabajo.

Se presentó también en este trabajo un caso de estudio que consiste en el registro de una donación al Banco de Alimentos de La Plata⁵⁴. El caso logra demostrar las diferentes alternativas que el usuario se encuentra al completar una tarea, así como también, el potencial de la componente para la creación de widgets adaptables desarrollado. Si bien este caso de estudio excede a la Ciencia Ciudadana, nos permite dejar en evidencia el potencial de la aplicación desarrollada y de la adaptabilidad de la misma.

Sobre dicho caso de estudio se realizaron pruebas en potenciales usuarios, presentando a cada uno un escenario distinto: no podían otorgar ciertos permisos o el dispositivo tenía ciertas restricciones de hardware.

Si bien la prueba consistió en una muestra probabilística de solo cinco usuarios uno sobre cada escenario de propuesto, y, por tanto, no se puede llegar a una conclusión fehaciente, los participantes de la prueba pudieron completar todas las tareas necesarias y de esta manera completar los registros de donación, a pesar de las condiciones adversas que le impusieron. Además, los participantes valoraron la adaptación de la que dispone la aplicación y recalcaron la mejora en la experiencia ante otras aplicaciones que de la misma manera no fuera posible utilizarlas. La usabilidad de la aplicación pudo ser evaluada a través de un cuestionario SUS que fue completado por los participantes, observando en el resultado de estos que los usuarios no tuvieron problemas y se sintieron cómodos. El resultado final es una prueba de conceptos exitosa.

A continuación, se detallan algunas de las posibles líneas de trabajo futuro a partir de los trabajos presentados en esta tesina.

Con respecto al componente para la creación de widgets adaptables, se considerarán nuevos permisos y restricciones de hardware que puedan afectar el comportamiento de una aplicación, así como también posibles modificaciones en los actuales. También, se probará con desarrolladores el uso del componente para determinar la mejora en la codificación de otras nuevas aplicaciones móviles adaptables.

Sobre la nueva versión de la aplicación móvil “Resuelvo Explorando”, se trabajará a corto plazo en realizar más pruebas con usuarios para poder determinar más fehacientemente las ventajas de la mejora presentada en esta tesina. Se seguirá contemplando en estas pruebas como principal aspecto a medir la cantidad de tareas completadas bajo los distintos escenarios. Dichas pruebas se vieron limitadas actualmente por la situación de pandemia de COVID-19.

Actualmente, todas las tareas en “Resuelvo Explorando” son opcionales para los voluntarios. Como trabajo a futuro se buscará que el responsable de proyecto, quien configura

⁵⁴ <https://bancoalimentario.org.ar/>

el mismo, pueda establecer a cada una de las tareas como opcionales u obligatorias para completar una muestra.

Además, desde “MoLE” es posible configurar un workflow de tareas que defina el orden de realización de estas por parte del voluntario en tres formas: libres (es decir, sin un orden establecido), en orden secuencial de creación o manual, donde es posible establecer libremente que tareas de la completitud de otras. Se pretende en un futuro llevar este workflow a “Resuelvo Explorando”, ya que actualmente el orden de las tareas no es considerado como una restricción, el voluntario puede realizar las tareas en el orden que desea, sin importar el orden establecido en el workflow.

La edición colaborativa de proyectos y la resolución colaborativa de tareas son otros aspectos a considerar en un futuro a largo plazo. Para esto se considerará la inclusión de usuarios en el ecosistema. La colaboración entre responsables de proyectos puede ayudar a la definición de los mismos, así como una mejor experiencia de estos. Por otro lado, la resolución colaborativa de tareas ayudará a que distintos voluntarios puedan salir a recolectar muestras de manera conjunta.

Por último, se considerará la creación de una API con la que la aplicación móvil “Resuelvo Explorando” se pueda comunicar para almacenar los datos recolectados por los voluntarios. De esta forma se buscará que las muestras tomadas puedan ser accedidas por los responsables del proyecto de una forma más sencilla. Actualmente, dichos datos quedan en el dispositivo del voluntario. Se considerará también, la creación de una interfaz web donde los creadores del proyecto puedan ver y obtener los datos de las muestras recolectadas y subidas a la API en cuestión.

De esta forma, se buscarán más opciones de configuración para las tareas y actividades y enriquecer el ecosistema que conforma “MoLE” y “Resuelvo Explorando”, así como mejorar la experiencia de los voluntarios y de los responsables de proyecto al utilizar dicho ecosistema para obtener muestras de Ciencia Ciudadana.

REFERENCIAS

- Aanensen, D. M., Huntley, D. M., Feil, E. J., Al-Own, F., & Spratt, B. G. (2009). EpiCollect: Linking smartphones to web applications for epidemiology, ecology and community data collection. *PLoS One*, *4*(9). <https://doi.org/10.1371/journal.pone.0006968>
- Abusair, M., Di Marco, A., & Inverardi, P. (2017). Context-aware adaptation of mobile applications driven by software quality and user satisfaction. *In 2017 IEEE International Conference on Software Quality, Reliability and Security Companion*, 31–38.
- Andriotis, P., Sasse, M. A., & Stringhini, G. (2017). Permissions snapshots: Assessing users' adaptation to the Android runtime permission model. *In 2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6. <https://doi.org/10.1109/WIFS.2016.7823922>
- Appenfeller, L. R., Lloyd, S., & Szendrei, Z. (2020). Citizen science improves our understanding of the impact of soil management on wild pollinator abundance in agroecosystems. *PLoS One*, *15*(3). <https://doi.org/10.1371/journal.pone.0230007>
- Berland, A., Roman, L. A., & Vogt, J. (2019). Can Field Crews Telecommute? Varied Data Quality from Citizen Science Tree Inventories Conducted Using Street-Level Imagery. *Forests*, *10*(4), 349–351. <https://doi.org/10.3390/f10040349>
- Boho, D., Rzanny, M., Wäldchen, J., Nitsche, F., Deggelmann, A., Wittich, H. C., Seeland, M., & Mäder, P. (2020). Flora Capture: a citizen science application for collecting structured plant observations. *BMC Bioinformatics*, *21*(1), 1–11. <https://doi.org/10.1186/s12859-020-03920-9>
- Bonney, R., Cooper, C., & Ballard, H. (2016). The Theory and Practice of Citizen Science: Launching a New Journal. *Citizen Science: Theory and Practice*, *1*(1). <https://doi.org/10.5334/cstp.65>
- Bonney, R., Cooper, C., Dickinson, J., Kelling, S., Phillips, T., Rosenberg, K. V., & Shirk, J. (2009). Citizen science: a developing tool for expanding science knowledge and scientific literacy. *BioScience*, *59*(11), 997–984.
- Bonney, R., Phillips, T. B., Ballard, H. L., & Enck, J. W. (2016). Can citizen science enhance public understanding of science? *Public Understanding of Science*, *25*(1), 2–16. <https://doi.org/10.1177/0963662515607406>
- Bonney, R., Shirk, J. L., Phillips, T. B., Wiggins, A., Ballard, H. L., Miller-Rushing, A. J., & Parrish, J. K. (2014). Next Steps for Citizen Science. *Science*, *343*(6178), 1436–1437. <https://doi.org/10.1016/j.biocon.2013.07.037>
- Boone, M. E., & Basille, M. (2019). Using iNaturalist to Contribute Your Nature Observations to Science 1. *EDIS*, *4*, 5–5.
- Britton, C. J., Level, A. V., & Gardner, M. A. (2013). Crowdsourcing: Divide the Work and Share the

- Success. *Library Hi Tech News*, 30(4), 1–5. <https://doi.org/10.1108/LHTN-03-2013-0017>
- Brooke, J. (1996). SUS: A quick and dirty usability scale. *Usability Evaluation in Industry*, 189.
- Brossard, D., Lewenstein, B. V., & Bonney, R. (2005). Scientific knowledge and attitude change: The impact of a citizen science project. *International Journal of Science Education*, 27(9), 1099–1121. <https://doi.org/10.1080/09500690500069483>
- Brunette, W., Sundt, M., Dell, N., Chaudhri, R., Breit, N., & Borriello, G. (2013). Open Data Kit 2.0: Expanding and refining information services for developing regions. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, 1–6. <https://doi.org/10.1145/2444776.2444790>
- Chau, M. M. (2020). Rapid Response to a Tree Seed Conservation Challenge in Hawai‘i Through Crowdsourcing, Citizen Science, and Community Engagement. *Journal of Sustainable Forestry*, 1–19.
- Cohn, J. P. (2008). Citizen science: Can volunteers do real research? *BioScience*, 58(3), 192–197. <https://academic.oup.com/bioscience/article-abstract/58/3/192/230689>
- Crall, A. W., Newman, G. J., Stohlgren, T. J., Holfelder, K. A., Graham, J., & Waller, D. M. (2011). Assessing citizen science data quality: An invasive species case study. *Conservation Letters*, 4(6), 433–442. <https://doi.org/10.1111/j.1755-263X.2011.00196.x>
- Daume, S. (2016). Mining Twitter to monitor invasive alien species—An analytical framework and sample information topologies. *Ecological Informatics*, 31, 70–82.
- Daume, S., & Galaz, V. (2016). “Anyone know what species this is?” - Twitter conversations as embryonic citizen science communities. *PLoS One*, 11(3). <https://doi.org/10.1371/journal.pone.0151387>
- Davids, J. C., Rutten, M. M., Pandey, A., Devkota, N., Oyen, W. D. V., Prajapati, R., & Giesen, N. V. D. (2019). Citizen science flow—an assessment of simple streamflow measurement methods. *Hydrology and Earth System Science*, 23(2), 1045–1065.
- Delaney, D. G., Sperling, C. D., Adams, C. S., & Leung, B. (2008). Marine invasive species: Validation of citizen science and implications for national monitoring networks. *Biological Invasions*, 10(1), 117–128. <https://doi.org/10.1007/s10530-007-9114-0>
- Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), 4–7. <https://doi.org/10.1007/s007790170019>
- Di Claudio, F., Corporaal, F. R. M., Lliteras, A. B., Grigera, J., & Gordillo, S. E. (2020). Adaptación de una aplicación móvil de recolección de datos para Ciencia Ciudadana. In *VI Simposio Argentino Sobre Tecnología y Sociedad (STS 2020)-JAIIO 49 (Buenos Aires)*. In Press.
- Di Claudio, Federico. (2021). Resultados caso de estudio de tesina "Adaptabilidad sobre permisos y hardware de dispositivos móviles para favorecer la recolección de datos en Ciencia Ciudadana" (Version 1) [Data set]. Zenodo. <http://doi.org/10.5281/zenodo.4593452>

- Dickinson, J. L., Zuckerberg, B., & Bonter, D. N. (2010). Citizen science as an ecological research tool: Challenges and benefits. *Annual Review of Ecology, Evolution, and Systematics*, *41*, 149–172. <https://doi.org/10.1146/annurev-ecolsys-102209-144636>
- Droege, S. (2007). Just because you paid them doesn't mean their data are better. In *Citizen Science Toolkit Conference. Cornell Laboratory of Ornithology*, 13–26.
- Ellul, C., Gupta, S., Haklay, M. M., & Bryson, K. (2013). A platform for location based app development for citizen science and community mapping. In *In Progress in Location-Based Services* (Issue 9783642342028, pp. 71–90). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-34203-5_5
- Elmalaki, S., Wanner, L., & Srivastava, M. (2015). CAreDroid: Adaptation framework for android context-aware applications. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, 2015-September*, 386–399. <https://doi.org/10.1145/2789168.2790108>
- Fortson, L., Masters, K., Nichol, R., Borne, K., Edmondson, E., Lintott, C., Raddick, J., Schawinski, K., & Wallin, J. (2012). Galaxy Zoo. *Advances in Machine Learning and Data Mining for Astronomy*, 213–236.
- Gasparis, I., Qian, Z., Song, C., Krishnamurthy, S. V, Gupta, R., & Yu, P. (2019). Figment: Fine-grained Permission Management for Mobile Apps. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications, 2019-April*, 1405–1413. <https://doi.org/10.1109/INFOCOM.2019.8737436>
- Hart, A. G., Carpenter, W. S., Hlustik-Smith, E., Reed, M., & Goodenough, A. E. (2018). Testing the potential of Twitter mining methods for data acquisition: Evaluating novel opportunities for ecological research in multiple taxa. *Methods in Ecology and Evolution*, *9*(11), 2194–2205. <https://doi.org/10.1111/2041-210X.13063>
- Heigl, F., & Zaller, J. G. (2014). Using a citizen science approach in higher education: a case study reporting roadkills in Austria. *Human Computation*, *1*(2). <http://thebartonmethod.com/index.php/jhc/article/view/10>
- Hesley, D., Burdeno, D., Drury, C., Schopmeyer, S., & Lirman, D. (2017). Citizen science benefits coral reef restoration activities. *Journal for Nature Conservation*, *40*, 94–99.
- Hines, D., & Sibbald, S. L. (2015). Citizen science: Exploring its application as a tool for prodromic surveillance of vector-borne disease. *Canada Communicable Disease Report*, *41*(3), 63–67. <https://doi.org/10.14745/ccdr.v41i03a04>
- Horn, G. V, Mac Aodha, O. M., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., & Belongie, S. (2018). The iNaturalist Species Classification and Detection Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8769–8778.
- Irwin, A. (1995). *Citizen Science: A Study of People, Expertise and Sustainable Development*. Psychology Press. <https://doi.org/10.1177/016224399702200406>

- Kim, S., Mankoff, J., & Paulos, E. (2013). Sensr: Evaluating a flexible framework for authoring mobile data-collection tools for citizen science. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 1453–1462. <https://doi.org/10.1145/2441776.2441940>
- Kobori, H., Dickinson, J. L., Washitani, I., Sakurai, R., Amano, T., Komatsu, N., Kitamura, W., Shinichi, •, Kazuo Koyama, T., Ogawara, T., Miller-Rushing, • A J, Kobori, H., Dickinson, J. L., Washitani, I., Sakurai, R., Amano, T., Komatsu, N., Kitamura, W., Takagawa, S., ... Ogawara, T. (2016). Citizen science: a new approach to advance ecology, education, and conservation. *Ecological Research*, 31(1), 1–19. <https://doi.org/10.1007/s11284-015-1314-y>
- Koke, J., Heimlich, J., Kessler, C., Ong, A., & Ancelet, J. (2007). *Project Butterfly WINGS: Winning Investigative Network for Great Science*. Summative Evaluation Report, Institute for Learning Innovation.
- Li, L., Bartel, A., Bissyandé, T., Klein, J., Le Traon, Y., Arzt, S., ..., & MacDaniel, P. (2015). Iccta: Detecting inter-component privacy leaks in android apps. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, 1, 280–291.
- Lliteras, A. B., Grigera, J., Corporaal, F. R. M., Di Claudio, F., & Gordillo, S. E. (2019). A Flexible Web Authoring Tool for Building Mobile Learning Experiences. In *Argentine Congress of Computer Science*, 1184 CCIS, 69–83. https://doi.org/10.1007/978-3-030-48325-8_5
- Louv, R., & Fitzpatrick, J. W. (2012). Citizen science: Public participation in environmental research. *Cornell University Press*.
- Milic, E. M., & Stojanovic, D. (2017). EgoSENSE: A Framework for Context-Aware Mobile Applications Development. *Engineering, Technology & Applied Science Research*, 7(4), 1791–1796. <https://doi.org/10.48084/etasr.1203>
- Mirri, S., Prandi, C., & Salomoni, P. (2014). A context-aware system for personalized and accessible pedestrian paths. *Proceedings of the 2014 International Conference on High Performance Computing and Simulation, HPCS 2014*, 833–840. <https://doi.org/10.1109/HPCSim.2014.6903776>
- Olejník, K., Dacosta, I., Machado, J. S., Huguenin, K., Khan, M. E., & Hubaux, J. P. (2017). SmarPer: Context-Aware and Automatic Runtime-Permissions for Mobile Devices. In *2017 IEEE Symposium on Security and Privacy (SP)*, 1058–1076. <https://doi.org/10.1109/SP.2017.25>
- Patridge, E. F., & Bardyn, T. P. (2018). Research electronic data capture (REDCap). *Journal of the Medical Library Association: JMLA*, 106(1), 142–143.
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2013). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414–454.
- Pescott, O., Walker, K., Pocock, M., Jitlal, M., Outhwaite, C., Cheffings, C., ..., & Roy, D. (2015). Ecological monitoring with citizen science: the design and implementation of schemes for recording plants in Britain and Ireland. *Biological Journal of the Linnean Society*, 115(3), 505–521.

- Portas, A. M., Barnard, L., Scott, C., & Giles Harrison, R. (2016). The National Eclipse Weather Experiment: use and evaluation of a citizen science tool for schools outreach. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2077). <https://doi.org/10.1098/rsta.2015.0223>
- Price, C. A., & Lee, H. S. (2013). Changes in participants' scientific attitudes and epistemological beliefs during an astronomical citizen science project. *Journal of Research in Science Teaching*, 50(7), 773–801. <https://doi.org/10.1002/tea.21090>
- Roberts, H. (2017). Using Twitter data in urban green space research. *Appl. Geogr*, 81, 13–20. <https://doi.org/10.1016/j.apgeog.2017.02.008>
- Robinson, L. D., Cawthray, J. L., West, S. E., Bonn, A., & Ansine, J. (2018). Ten principles of citizen science. In *Citizen Science: Innovation in Open Science, Society and Policy*, 27–40.
- Rowbotham, S., McKinnon, M., Leach, J., Lamberts, R., & Hawe, P. (2019). Does citizen science have the capacity to transform population health science? *Critical Public Health*, 29(1), 118–128.
- Santori, C., Keith, R. J., Whittington, C. M., Thompson, M. B., Van Dyke, J. U., & Spencer, R. J. (2021). Changes in participant behaviour and attitudes are associated with knowledge and skills gained by using a turtle conservation citizen science app. *People and Nature*, 3, 66–76. <https://doi.org/10.1002/pan3.10184>
- Sboui, T., Ayed, M. B., & Alimi, A. M. (2017). A UI-DSPL approach for the development of context-adaptable user interfaces. *IEEE Access*, 6, 7066–7081.
- Schilit, B. N., & Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *IEEE Network*, 8(5), 22–32.
- Schmeller, D. S., Henry, P. Y., Julliard, R., Gruber, B., Clobert, J., Dziöck, F., ..., & Henle, K. (2009). Advantages of volunteer-based biodiversity monitoring in Europe. *Conservation Biology*, 23(2), 307–316.
- Schmitz, H., Howe, C. L., Armstrong, D. G., & Subbian, V. (2018). Leveraging mobile health applications for biomedical research and citizen science: a scoping review. *Journal of the American Medical Informatics Association*, 25(12), 1685–1695. <https://doi.org/10.1093/jamia/ocy130>
- Silvertown, J. (2009). A new dawn for citizen science. *Trends in Ecology & Evolution*, 24(9), 467–471.
- Simpson, R., Page, K. R., & De Roure, D. (2014). Zooniverse: Observing the World's Largest Citizen Science Platform. In *Proceedings of the 23rd International Conference on World Wide Web*, 1049–1054. <https://doi.org/10.1145/2567948.2579215>
- Soui, M., Diab, S., Ouni, A., Essayeh, A., & Abed, M. (2017). An ontology-based approach for user interface adaptation. *Advances in Intelligent Systems and Computing*, 512, 199–215. https://doi.org/10.1007/978-3-319-45991-2_13

- Spicer, H., Nadolny, D., & Fraser, E. (2020). Going Squirrely: Evaluating Educational Outcomes of a Curriculum-aligned Citizen Science Investigation of Non-native Squirrels. *Citizen Science: Theory and Practice*, 5(1), 14. <https://doi.org/10.5334/cstp.275>
- Sullivan, B. L., Wood, C. L., Iliff, M. J., Bonney, R. E., Fink, D., & Kelling, S. (2009). eBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10), 2282–2292. <https://doi.org/10.1016/j.biocon.2009.05.006>
- Van Strien, A. J., Van Swaay, C. A. M., & Termaat, T. (2013). Opportunistic citizen science data of animal species produce reliable estimates of distribution trends if analysed with occupancy models. *Journal of Applied Ecology*, 50(6), 1450–1458. <https://doi.org/10.1111/1365-2664.12158>
- Van Tonder, B., & Wesson, J. (2008). Using adaptive interfaces to improve mobile map-based visualisation. *ACM International Conference Proceeding Series*, 338, 257–266. <https://doi.org/10.1145/1456659.1456689>
- Varela, G., Paz-Lopez, A., Becerra, J., & Duro, R. (2016). A Framework for the Development of Context-Adaptable User Interfaces for Ubiquitous Computing Systems. *Sensors*, 16(7), 1049. <https://doi.org/10.3390/s16071049>
- Vercayie, D., & Herremans, M. (2014). Citizen science and smartphones take roadkill monitoring to the next level. *Nature Conservation*, 11, 29–40. <https://doi.org/10.3897/natureconservation.11.4439>
- Walton, P., & Maiden, N. (2019). Integrated software reuse: management and techniques. *Routledge*.
- Wang, Y., Kaplan, N., Newman, G., & Scarpino, R. (2015). CitSci.org: A New Model for Managing, Documenting, and Sharing Citizen Science Data. *PLoS Biology*, 13(10), 1–5. <https://doi.org/10.1371/journal.pbio.1002280>
- Wiggins, A., & Crowston, K. (2011). From conservation to crowdsourcing: A typology of citizen science. In *2011 44th Hawaii International Conference on System Sciences*, 1–10.
- Wijesekera, P., Egelman, S., Wagner, D., Beznosov, K., Baokar, A., & Hosseini, A. (2015). Android Permissions Remystified: A Field Study on Contextual Integrity. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, 514.
- Wilke, C., Richly, S., Götz, S., Piechnick, C., & Aßmann, U. (2013). Energy consumption and efficiency in mobile applications: A user feedback study. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 134–141. <https://doi.org/10.1109/GreenCom-iThings-CPSCoM.2013.45>
- Wing, S., Horton, R. A., Marshall, S. W., Thu, K., Tajik, M., Schinasi, L., & Schiffman, S. S. (2008). Air pollution and odor in communities near industrial swine operations. *Environmental Health Perspectives*, 116(10), 1362–1368. <https://doi.org/10.1289/ehp.11250>

Wright, A. (2016). REDCap: A Tool for the Electronic Capture of Research Data. *Journal of Electronic Resources in Medical Libraries*, 13(4), 197–201.
<https://doi.org/10.1080/15424065.2016.1259026>

A. ANEXO A – INSTRUCTIVO PARA EL USO DEL COMPONENTE

En este anexo se describe la forma de utilización componente para la creación de widgets adaptables, empleado en este trabajo para la adaptación de la herramienta “Resuelvo Explorando”. Dicho componente puede ser empleado en la creación de aplicaciones móviles, Android y/o iOS, que utilicen el framework React Native.

A.1. AGREGAR COMPONENTE A UN PROYECTO

Para el correcto funcionamiento del componente, el mismo necesita de una serie de dependencias instaladas en nuestro proyecto:

- Expo: Expo es un framework que nos brinda un conjunto de herramientas para facilitar el desarrollo de aplicaciones usando React Native. Es utilizada por la componente para el manejo de permisos de manera indiferente para iOS y Android.
- Expo-Permissions: Extensión del framework Expo, que consiste en un módulo para el fácil y mejor manejo de permisos en los sistemas operativos Android y iOS.
- Expo-Battery: Extensión del framework Expo, que consiste en un módulo que contiene todos los métodos necesarios para consultar el estado actual de la batería y las configuraciones del usuario con respecto a esta.
- NetInfo: Es una librería propia de React Native, que nos permite manejar las conexiones a internet que dispone el dispositivo.

Para la instalación de Expo y NetInfo se deben ejecutar los siguientes comandos en npm o yarn (gestor de dependencias que estemos usando):

```
npm install --global expo-cli
npm install @react-native-community/netinfo
```

```
yarn add --global expo-cli
yarn add @react-native-community/netinfo
```

Una vez instalado Expo, para importar los módulos para el manejo de permisos y consultas del estado de la batería se debe ejecutar, utilizando el motor de dependencias de Expo:

```
expo install expo-permissions
expo install expo-battery
```

Para agregar el componente para la creación de widgets adaptables a un proyecto primero se debe descargar desde su repositorio en Github mediante el siguiente link: [Repositorio Componente](#)

Una vez descargado basta con agregar en la ubicación que se prefiera, dentro del proyecto, los tres archivos que se necesitan para su utilización:

- “constants.js”: Declara las constantes para manejar permisos, conectividad y batería.
- “permissionMap.js”: Realiza el mapeo entre las constantes definida en su respectivo archivo y los permisos que declara Expo-Permissions.
- “permissionAwareComponent.js”: Contiene los métodos y lógica para el renderizado de uno u otro componente según las condiciones de hardware y permisos evaluadas. Este es el archivo que define el componente “PermissionAwareComponent” que luego será importado en el componente que se necesite.

La estructura de carpetas del proyecto debería ser similar a la que se muestra en la *Figura 18*.

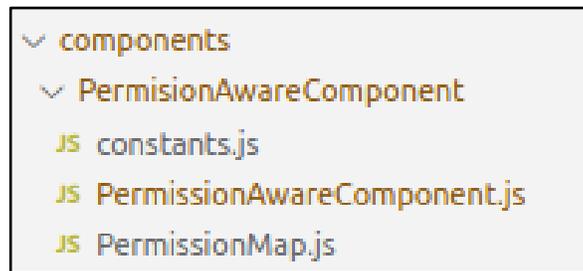


Figura 18 - Estructura de carpetas al importar el componente para la creación de widgets adaptables.

A.2. USO DEL COMPONENTE

Para el uso del componente debemos importar en el archivo correspondiente la clase “PermissionAware”, la cual se trata del componente a renderizar y “PermissionConstants” que contiene las constantes que representan los permisos y restricciones de hardware disponibles para el uso del componente. Para una mayor comodidad, es posible importar ambas desde el archivo “permissionAwareComponent.js” como se muestra a continuación:

```
import {
  PermissionAware,
  PermissionConstants,
} from '../PermissionAwareComponent/PermissionAwareComponent'
```

Para definir qué componente debemos renderizar ante ciertas permisos otorgados y restricciones de hardware dadas, debemos definir un arreglo que contenga diccionarios. Cada diccionario consiste en un componente posible a renderizar que será efectivo si las condiciones que también se describan en dicho diccionario sean dadas. La evaluación se realiza en el orden que fueron establecidas las componentes del arreglo.

En cada diccionario se debe definir:

- Un arreglo llamado “permission” que contendrá los permisos necesarios definidos a modo de constantes anteriormente importadas. Si el componente no requiere ningún permiso, obviar esta clave.
- El valor “connectionRequire” indica el tipo de conexión necesaria. En caso de que no se requiera definir esta restricción, no debe colocarse esta clave.
- El valor “batteryLevelRequire” permite establecer el porcentaje de batería necesario. Puede obviarse en caso no sea necesario.
- El valor “powerSaverCondition” permite establecer a modo de constante si el modo de ahorro de energía debe estar activado o desactivado. Si esto no es una condición necesaria para la alternativa, debe obviarse.
- Finalmente, el valor “component” consiste en un componente a renderizar en caso de que las condiciones impuestas por los valores anteriores se cumplan.

Todos los valores posibles de las propiedades detalladas fueron importadas anteriormente y se listan más adelante.

Cabe aclarar que una vez la evaluación de las opciones se realiza de manera secuencial, deteniéndose ante la primera alternativa que resulte efectiva.

El siguiente código muestra un ejemplo de instanciación de este diccionario:

```
componentList = [
  (
    Permission:[PermissionConstants.CAMERA,
    PermissionConstants.AUDIO_RECORDING ],
    conectionRequire: PermissionConstants.WIFI,
    batteryLevelRequire: 50,
    powerSaverCondition: PermissionConstants.NO_POWER_SAVER,
    component: (<ComponenteA />)
  ),
  (
    Permission:[PermissionConstants.CONTACTS],
    conectionRequire: PermissionConstants.ANY,
    component: (<ComponenteB />)
  ),
]
```

En el ejemplo, el *ComponenteA* se muestra al usuario en caso de que este haya concedido permisos para acceder a la cámara y el micrófono, este conectado a la red WI-FI, disponga de un porcentaje de batería mayor o igual al 50% y no tenga activado el modo ahorro de energía en su dispositivo. En caso de que algunas de estas condiciones no se cumplan, el componente pasa a evaluar si mostrar al usuario el siguiente componente. El *ComponenteB* se mostrará al usuario si este posee conexión a internet de cualquier tipo y el usuario dio permiso a la aplicación para acceder a sus contactos.

Finalmente dentro del método *render()* del componente donde queramos invocar el widget adaptable debemos invocar el componente empleado. Debemos asignarle como propiedad

“permissionComponentList” que será el arreglo de diccionarios creado y mediante la propiedad “defaultComponent” un componente por defecto que se renderizará solo en caso de que ninguna alternativa presente en el arreglo anterior haya sido valida. Definir ambas propiedades del componente es obligatorio.

El siguiente código muestra un ejemplo de invocación del componente:

```
render() {
  return (
    < PermissionAwareComponent
      permissionComponentList={this.componentList}
      defaultComponent={({ < DefaultComponent /> })}
    />
  );
}
```

En este ejemplo se puede ver que se va renderizar el widget adaptable, con la alternativa que se cumpla dentro del arreglo definido anteriormente y que ahora fue definido como una propiedad del componente. La segunda propiedad definida establece que el componente *Default Component* se renderiza por defecto, es decir, en caso de que no sea posible ninguna de las alternativas del arreglo definido.

A.3. PERMISOS Y RESTRICCIONES DE HARDWARE

En esta sección se listarán los permisos y restricciones de hardware que maneja el componente presentado. Todas están definidas como constantes en el archivo “permissionMap.js”, pero pueden ser accedidas con mayor facilidad importando “PermissionConstants” desde el archivo “permissionAwareComponent.js”.

A continuación, se detallan las constantes que representan a los permisos que el componente maneja y que pueden ser establecidos como necesarios para mostrar un widget concreto.

- CAMERA: Acceso a la cámara del dispositivo
- AUDIO_RECORDING: Acceso al micrófono del dispositivo
- CONTACTS: Acceso a los contactos del usuario
- MEDIA_LIBRARY: Acceso al almacenamiento interno del dispositivo en modo lectura y escritura
- MEDIA_LIBRARY_WRITE_ONLY: Acceso al almacenamiento interno del dispositivo en modo solo escritura
- CALENDAR: Acceso a los datos de calendario del usuario
- REMINDERS (exclusivo de iOS): Acceso a los recordatorios del usuario.
- SYSTEM_BRIGHTNESS: Acceso al brillo de pantalla del dispositivo para cambiar el mismo.

- NOTIFICATIONS: Permiso para enviar notificaciones al usuario de tipo “user facing” o “push”.
- USER_FACING_NOTIFICATIONS: Permiso para enviar notificaciones al usuario de tipo “push”.
- LOCATION: Acceso a la ubicación del dispositivo mediante el uso de GPS.

Las constantes para el manejo de batería son las siguientes:

- NO_POWER_SAVER: El modo de ahorro de energía debe estar desactivado.

Por último, las constantes para especificar el tipo de conexión a internet son las siguientes:

- WIFI: Requerir conexión de tipo WI-FI.
- CELLULAR: Requerir conexión de datos móviles. También puede especificarse que tipo, si 2G, 3G o 4G de la siguiente manera: CELLULAR[‘4G’] para el 4G por ejemplo.
- ANY: Requerir conexión a internet de cualquier tipo.
- NONE: Requerir que el dispositivo no disponga de conexión a internet.

No se requiere la utilización de constantes definidas para definir un porcentaje de batería, esto consiste simplemente en establecer un número de 1 a 100.

Cabe mencionar que el manejo tanto de permisos como de restricciones de hardware es transparente al sistema operativo en que se corra la aplicación realizada, salvo en algunos casos donde el permiso es exclusivo de un sistema operativo, en tal caso, fue aclarado en el listado.