

DEHIA, una plataforma para la generación y ejecución de actividades de recolección de datos con intervención humana aplicada en el Programa E-Basura

Jose Arcidiacono^{*1}[000-0003-0300-5213], Alejandra B. Lliteras^{2,3} [0000-0002-4148-1299],
and Patricia Bazán⁴[0000-0001-6720-345X]

¹ UNLP, Facultad de Informática, La Plata, Buenos Aires, Argentina
josefine2_94@hotmail.com

² UNLP, Facultad de Informática, LIFIA, La Plata, Buenos Aires, Argentina

³ CICIPBA, Buenos Aires, Argentina

alejandra.lliteras@lifia.info.unlp.edu.ar

⁴ UNLP, Facultad de Informática, LINTI, La Plata, Buenos Aires, Argentina
pbaz@info.unlp.edu.ar

Abstract. En este trabajo se presenta el desarrollo, en progreso, de DEHIA una plataforma para crear, clonar y gestionar workflows de actividades de recolección de datos, que involucren la intervención humana, y cuya ejecución se realiza desde una aplicación móvil. Para ello, se presenta una arquitectura de solución conformada, por tres componentes: una plataforma Web que permite a usuarios finales crear, clonar y gestionar sus propias actividades de recolección de datos en diferentes dominios, una aplicación móvil para que voluntarios participen de la actividad de recolección de datos y una API responsable de la comunicación entre las dos primeras componentes. Se presenta, además, un caso de uso proponiendo una alternativa tecnológica inclusiva y en este caso particular, con la expectativa de aportar a la labor social que el programa E-Basura realiza desde el año 2009. La propuesta de este trabajo espera abrir un espacio de debate en miras de la democratización y apertura de este tipo de actividad para ser empleado en diferentes disciplinas.

Keywords: Plataforma Web, Aplicación Móvil, API REST, Recolección de Datos, Reuso, Conocimiento, Arquitectura, Workflow, Herramienta de Autor

1 Introducción

La recolección de datos por parte de personas es una actividad que se usa en diversas disciplinas y que se realiza desde hace ya largo tiempo. Inicialmente, la recolección de los datos se realizaba empleando medios físicos como papel y lápiz y luego, con el avance de la tecnología, este medio fue reemplazado por otros, como por ejemplo las planillas de cálculo [1].

En la actualidad, con la masificación en el uso de dispositivos móviles, diversas aplicaciones móviles son usadas para la recolección de datos con intervención humana, es decir, toma de datos que no son completamente realizadas por sensores, sino que necesariamente se requiere de una persona interactuando con la aplicación.

Algunas de esas aplicaciones son creadas ad-hoc como solución a un problema específico y no son posibles de reusar en otros dominios. Su objetivo es la de satisfacer la necesidad puntual de un proyecto. Por ejemplo, en [2] el autor propone una aplicación para recolectar datos relacionados a ambientes acuáticos, en [3] los autores presentan “GeoVIN” una aplicación para recolectar datos ayudando a la sociedad a reconocer vinchucas. Por otro lado, en [4] los autores proponen una aplicación para recolectar datos del virus Zika en el mundo.

Otra forma de obtener aplicaciones móviles para recolectar datos, que no implica desarrollo ad-hoc, es mediante el uso de herramientas de autor o plataformas Web que le permiten a los usuarios finales crear sus propias aplicaciones móviles sin necesidad de tener conocimientos de programación, por ejemplo, en [5], los autores presentan a Zooniverse, una plataforma que, entre otras cosas, permite generar aplicaciones de recolección de datos para diversos dominios. En [6] se presenta EpiCollect+ otra herramienta de similares características, mientras que en [7] los autores presentan “MOLLE” una herramienta de autor para generar actividades que podrían aplicarse también a la recolección de datos.

Las aplicaciones creadas ad-hoc, con desarrollo de software propio, no permiten su reuso por parte de la comunidad, es decir, su utilidad, está circunscripta al proyecto que la genera. En contraposición, las herramientas de autor, permiten y democratizan el acceso a la generación de nuevas aplicaciones a bajos costos y sin necesidad de conocimientos específicos de desarrollo de software, las herramientas son reusables, no así, generalmente, las aplicaciones que con ellas se generan.

Considerando lo anterior, es deseable, al usar este tipo de herramientas, contar con la posibilidad de compartir conocimientos y tareas comunes a diferentes proyectos, lo que daría un acceso compartido y adicionalmente la posibilidad de compartir experiencias de diversos actores.

Lo relevante de las herramientas de autor, es que, como se mencionó anteriormente, permiten a usuarios sin conocimientos de programación crear sus propias propuestas, lo que les otorga la libertad de generar la aplicación móvil acorde a sus necesidades sin depender de desarrolladores o expertos en tecnología. En particular, sucede muchas veces que, diferentes proyectos de recolección de datos comparten ciertas tareas. Por ejemplo, es común que para ciertas actividades, se necesite recolectar una foto del lugar donde se recolecta el dato y la posición geográfica del mismo o bien que hay una tarea que implique grabar audio. De este modo, este tipo de tareas, podría ser reusado para diferentes proyectos, sin la necesidad de definirlas una y otra vez para cada nuevo proyecto.

Por otro lado, las tareas, se suelen organizar en lo que se conoce como workflow [8]. De este modo, el usuario, realiza las tareas de recolección de datos, siguiendo algún patrón de workflows [9] (por ejemplo, secuencial, libre, selección exclusiva y selección múltiple), es decir estructurar las tareas de alguna manera. Cada proyecto, puede definir su propia estructura, pero con el fin de reusar conocimiento, sería deseable que estas estructuras, como si fueran plantillas, también pudieran compartirse, basados, por ejemplo, en la buena experiencia de los usuarios, al momento de realizar las tareas estructuradas de cierta manera.

En este trabajo se presenta, una arquitectura para dar soporte a una plataforma Web que permite a usuarios finales crear, clonar y gestionar workflows, los cuales podrán ser ejecutados usando una aplicación móvil. El foco de la plataforma está en la posibilidad de reusar tareas en diferentes proyectos, así como estructuras de workflows, a los cuales se les podrá configurar nuevas tareas, denominándolo clonación.

El trabajo se estructura de la siguiente manera: en la Sección 2 se presentan trabajos relacionados, en la Sección 3 se presenta la propuesta de solución, en la Sección 4 se presenta un caso de uso basado en el Programa E-Basura, en la Sección 5 se expone una breve discusión y finalmente en la Sección 6 se presentan algunas conclusiones y trabajos futuros.

2 Trabajos Relacionados

En esta sección analizaremos algunas herramientas que permiten crear soluciones tecnológicas para la recolección de datos. El criterio de análisis, ha sido, por un lado, algunas de las más usadas en la comunidad (Epicollect5, Zooniverse, Open Data Kit) y por el otro, el trabajo que se tomó como base para su mejora (MoLE). El objetivo es realizar una breve descripción de cada uno de ellos.

EpiCollect5¹ es una plataforma de creación de formularios para recolección de datos usando smartphones. Fue desarrollada en el Imperial College de Londres, inicialmente para el estudio de epidemiología (de ahí el nombre) [10]. Es la evolución de EpiCollect [10] y EpiCollect+ [6]. En cuanto al diseño de formularios, Epicollect permitía únicamente la recolección de fotos y posicionamiento del teléfono. EpiCollect+ añadió respuestas de texto y respuestas con opciones. Actualmente EpiCollect5 soporta diversos tipos de tareas, desde ingresar una fecha hasta leer códigos de barras. Permite también definir tareas como “obligatorias”, así como indicar saltos condicionales entre tareas (planificación). Las tareas deben crearse desde cero con el formulario: no se permite reuso de tareas o planificaciones. Expone una API pública, pero es de sólo lectura. Si bien EpiCollect y EpiCollect+ pueden encontrarse aún en repositorios públicos^{2,3}, EpiCollect5 es de código cerrado.

Zooniverse⁴ es un proyecto que lleva a cabo lo que se conoce como Ciencia Ciudadana [11], en el cual investigadores recurren a personas comunes para ayudarlos en el análisis de datos a una escala que no sería posible por su cuenta [12]. Un investigador puede crear una colección de sujetos (generalmente imágenes), en base al cual los usuarios tienen que realizar ciertas tareas. La plataforma permite la creación de “proyectos” donde se diseñan estas tareas para una determinada colección. La herramienta de creación permite distintos tipos de tareas, desde responder una pregunta hasta señalar algo en una imagen con herramientas de dibujo [13]. Es un proyecto de código abierto⁵. Tiene una API pública, que permite el envío de “clasificaciones” (tareas

¹ <https://five.epicollect.net/>

² https://github.com/ImperialCollegeLondon/EpiCollect_server/

³ <https://github.com/ImperialCollegeLondon/EpiCollectplus>

⁴ <https://www.zooniverse.org/>

⁵ <https://github.com/zooniverse/>

resueltas para un sujeto de la colección), así como operaciones sobre proyectos y colecciones, y descarga de resultados. No permite la reutilización de tareas y, si bien tiene una aplicación móvil, ésta está pensada para proyectos simples de una sola tarea (la resolución es mayoritariamente Web).

Open Data Kit⁶ es una plataforma para recolectar, administrar y usar datos en ambientes con recursos limitados. Es de propósito general: puede ser usado para censos, actividades educativas, medicina, ecología, etc. Tiene un editor, ODK Build, que permite crear tareas mediante drag-and-drop, y usa el estándar XForm del W3C⁷ para la implementación de los formularios [14]. Esta herramienta permite crear numerosos tipos de tareas. Las tareas se crean de manera secuencial, pero pueden conectarse en otro orden mediante condiciones. Permite crear traducciones para las tareas. También se puede previsualizar el formulario. Es un proyecto de código abierto⁸. No permite la reutilización de tareas ni de workflows.

Por otro lado, MoLE⁹ es una herramienta Web de autor de código abierto que permite a usuarios sin conocimiento sobre desarrollo móvil o programación, la creación de actividades mediadas por tecnología móvil [7]. Las actividades se componen de tareas. La herramienta Web permite estructurar las tareas de forma lineal, libre o de grafo, y estas pueden ser de tipo “elegir una opción”, “generar contenido multimedia”, “respuesta libre”, “recolección” o “depósito”. MoLE cuenta con una aplicación móvil llamada “Resuelvo Explorando”¹⁰ [15] para resolver las actividades creadas desde el editor Web. Tanto en la herramienta Web como la aplicación móvil admiten idiomas español e inglés. También existe una API¹¹ que permite la creación de actividades y el envío de respuestas. En su versión publicada, no permite el uso de condiciones ni el reúso de tareas ni de workflows.

De los trabajos analizados, se observa que no todos abordan el reúso de tareas y de planificaciones (o workflows). Esta falta de reúso, puede hacer que el conocimiento no circule en la comunidad como sería esperado y que no todos se pudiesen beneficiar de las buenas prácticas de otros proyectos, por ejemplo, desde la perspectiva de la experiencia de usuario al conocer que una determinada forma de planificación conlleva ventajas por sobre otra.

3 DEHIA

En este trabajo propone una arquitectura de solución la cual será implementada considerando una plataforma Web para que, usuarios finales, puedan diseñar, crear y reusar workflows con tareas para recolección y análisis de datos, y una aplicación móvil, desde la cual usuarios puedan ejecutar los workflows creados desde la plataforma y así recolectar datos que son enviados a la plataforma.

⁶ <https://getodk.org/>

⁷ <http://w3.org/TR/xforms>

⁸ <https://github.com/getodk>

⁹ <https://github.com/cientopolis/MOLE-Authoring-Tool>

¹⁰ <https://github.com/cientopolis/MOLE-mobile-app>

¹¹ <https://github.com/cientopolis/MoLE-API>

A continuación, se describe por un lado la arquitectura propuesta y luego, los componentes más relevantes de la misma.

3.1 Arquitectura

DEHIA es un sistema distribuido con una arquitectura de microservicios (Fig. 1).

La arquitectura está definida en cuatro capas: (a) Capa de clientes: en este nivel están las aplicaciones que sirven de interfaz entre los usuarios (autores, usuarios Web y usuarios móviles) y el sistema. Se comunican directamente con la capa de compuertas (gateways); (b) Capa de compuertas: en este nivel se encuentran las compuertas de entrada al sistema, que direccionan los pedidos de los clientes a diferentes servicios; (c) Capa de servicios: en este nivel se encuentran los servicios propiamente dichos. Tienen comunicación con la capa de compuertas y con la de persistencia y (d) Capa de persistencia: en este nivel están las distintas formas de persistencia para cada servicio de la capa de servicios. Se comunican únicamente con la capa de servicios.

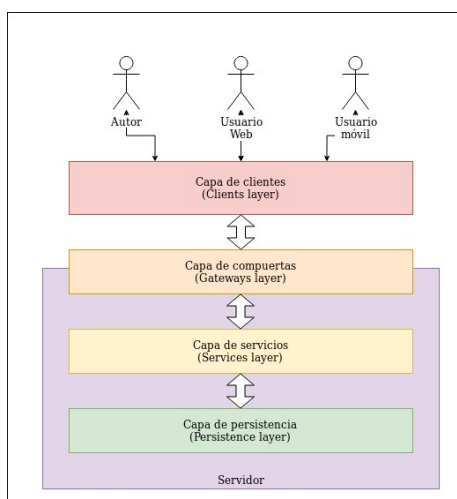


Fig. 1. Arquitectura por capas propuesta para DEHIA

3.2 Componentes

La solución propuesta está formada por tres componentes: la plataforma Web, desde la cual se crean y gestionan los workflows. Una aplicación móvil que permite ejecutar los workflows generados desde la plataforma Web. Y por último una API que permite la comunicación entre los dos componentes previamente mencionadas y es responsable de la persistencia de los datos.

A continuación, se describen cada una de las componentes mencionadas.

Plataforma Web. La plataforma Web, presenta una interfaz gráfica desde la cual un usuario puede acceder en forma anónima (invitado) o bien loguearse con su cuenta de

Google. En este caso, el usuario pasa a ser un autor. La Fig. 2 muestra la pantalla de login a la plataforma.

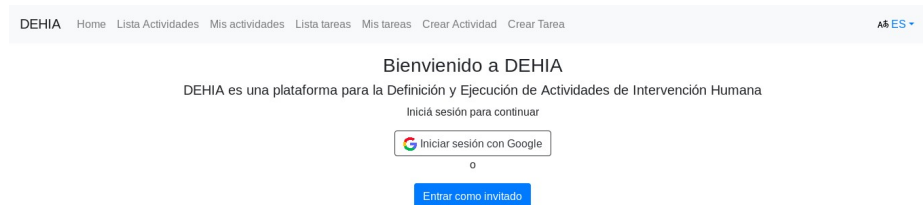


Fig. 2. Ingreso a la plataforma Web de DEHIA

Un autor, puede crear tareas y acceder al listado de las tareas creadas por él o bien a las que son de uso público. Es posible crear diferentes tipos de tareas, por ejemplo, simple, ingresar texto, ingresar número, sacar una foto, elegir una opción, opción múltiple, recolección, depósito, localización y audio, entre otras.

Una tarea se crea como pública o privada. En caso de ser pública está disponible para cualquier usuario, inclusive para los que ingresan al sistema de manera anónima. Las tareas privadas, sólo son visibles por sus autores. Todas las tareas, pueden ser reusadas dentro de su alcance de privacidad declarado.

Un autor, puede crear una planificación, la que se representa mediante un workflow. Esta puede ser secuencial (una tarea se realiza luego de otra en un orden establecido), selección simple o múltiple (es la idea de grafo, donde las aristas pueden ser condicionales) o bien libres (sin un orden preestablecido).

La planificación consiste en dos etapas: una primera etapa en la que se decide qué tareas formarán parte del workflow, creándose tareas si hacen falta, y una segunda etapa en la que se genera el workflow de forma gráfica mediante un editor.

El editor gráfico de workflows está conformado por nodos (círculos) que representan las tareas elegidas en la etapa anterior. En principio, todas las tareas son iniciales, pudiéndose agregar conexiones entre ellas (por ejemplo, después de la tarea A se resolverá la tarea B). Se permite también poner condiciones, por ejemplo “Si el usuario eligió la opción 1 en la tarea A, pasará a la tarea B. Si no, pasará a la tarea C”. Los grafos con ciclos son válidos, por ejemplo, para permitir la creación de menús. El sistema verifica que exista al menos una salida del grafo para que la actividad pueda terminar. La Fig. 3 muestra el editor gráfico para workflows. En dicha figura, las tareas iniciales se indican con círculos verdes, mientras que las finales en rojo. El resto de las tareas son intermedias (se visualizan en color gris). Las flechas que conectan tareas, tienen un cuadrado en caso de ser condicionales (por ejemplo, transición de la tarea 2 a la tarea 3 de la Fig.3) o bien un círculo cuando carecen de condicionalidad (por ejemplo, transición de la tarea 1 a la tarea 2 de la Fig.3).

Un workflow cuando se crea se encuentra en estado “nuevo”, cuando el autor decide que se puede iniciar la carga de datos, lo pone en estado “definitivo” lo que hace que ya no se pueda seguir modificando el workflow. Una vez que un workflow está

disponible, el autor puede decidir poner el workflow en un estado “cerrado” para no recibir datos temporal o definitivamente.



Fig. 3. Editor gráfico de Workflows

Una característica importante de esta plataforma Web es que permite reusar, por un lado, las tareas, y por el otro, los workflows creados generando “clonaciones” de un workflow con otras tareas y cambiando el autor del mismo al usuario que lo reusa. Desde la plataforma, se crean y se re utilizan planificaciones o workflows. En el primer caso, cuando se crea un workflow se indican las transiciones de una tarea a otra y en caso de ser necesario se indica la condición que se debe cumplir. Cuando se reusa un workflow existente, el usuario debe decidir qué tareas cambia y en caso de haber transiciones condicionales, la plataforma valida que se cumplan las mismas con las nuevas tareas. La Fig. 4 muestra la clonación de un workflow que involucra una transición condicional. En la Fig. 4.a. Se muestra la etapa de selección de tareas a reemplazar en la clonación, mientras que en la Fig. 4.b. Se muestra la clonación desde el editor gráfico.

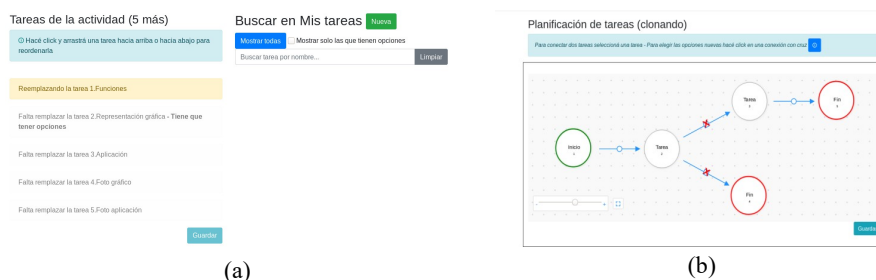


Fig. 4. Clonación de un workflow

La plataforma Web, fue implementada usando React¹² y Redux¹³ (Javascript). La información de una actividad de recolección de datos, queda guardada en formato JSON¹⁴ para luego ser descargada desde la aplicación móvil.

¹² <https://reactjs.org/>

Cada actividad creada se guarda en el perfil de usuario del autor y queda disponible para ser compartida vía código QR con los usuarios que ejecutarán la actividad y recolectarán los datos. La Fig. 5 muestra una lista de actividades creadas por el autor que se encuentra logueado.

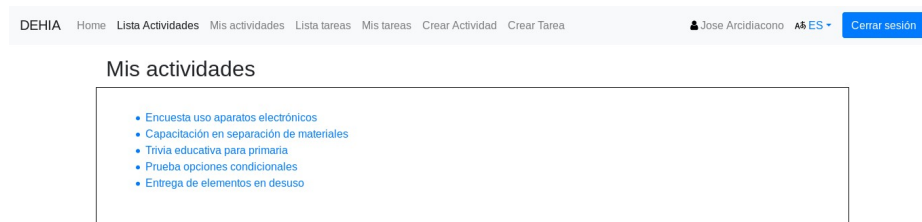


Fig. 5. Listado de actividades creadas por un usuario con perfil autor

Aplicación Móvil. La aplicación móvil, permite descargar desde la plataforma Web (servidor) los workflows en estado disponible. La Fig. 6 muestra una imagen representativa de un usuario leyendo desde su dispositivo móvil un código QR para descargar una actividad creada desde la plataforma Web (descrita en la sección anterior).



Fig. 6. Representación de un usuario final leyendo un QR

Una vez descargada una actividad, es posible ejecutarla y luego cuando se disponga de conectividad, enviar los datos recolectados desde la aplicación de vuelta al servidor.

En el dispositivo móvil del usuario final se presenta una lista de las actividades que haya descargado. El usuario elige una actividad, y las tareas se le van presentando de acuerdo a la planificación creada por el autor (workflow). Actualmente, la aplicación móvil soporta poco más de diez tipos de tareas, incluyendo respuestas de texto, imágenes, opción múltiple, recolección y depósito de elementos marcados con un código QR. Cuando el usuario llega al final de la actividad, puede enviar el resultado, que será recibido y procesado por la API que se describe en la sección siguiente.

La implementación de la aplicación móvil ha sido realizada usando ReactNative¹⁵.

API. La API REST está definida con tres funciones principales: crear workflows de tareas (desde la plataforma Web) que van a ser resueltas por personas que usan la aplicación móvil, recibir las respuestas enviadas por dicha aplicación, y mostrar los

¹³ <https://redux.js.org/>

¹⁴ <https://www.json.org/>

¹⁵ <https://reactnative.dev/>

resultados obtenidos (en la plataforma Web). La API tiene como funcionalidad adicional, la de validar el ingreso de usuarios autenticados en el sistema para diferenciar un autor de un invitado.

Respecto a la funcionalidad de “creación de workflows”, la API permite la creación de actividades y tareas, así como la información del dominio a los que pertenecen ambas. Permite crear planificaciones para las actividades, que consisten en un conjunto de “saltos” (es decir, tareas destino y condiciones para avanzar en el workflow para cada tarea), así como información sobre cuáles tareas son opcionales y cuáles son los puntos de inicio del workflow (“tareas iniciales”). También brinda listados paginados de actividades y tareas (listado público y listado del autor actual) y se puede consultar información sobre el usuario actual. Ofrece los listados de atributos de tareas y actividades como idiomas, tipos de tarea, dominios, etc. Por último, se pueden consultar individualmente tareas y actividades (siempre que sean accesibles por el autor) y descargar la definición de un workflow en formato JSON, listo para importar desde la aplicación móvil.

La funcionalidad de “recepción de respuestas”, es la encargada de recibir las respuestas enviadas por la aplicación móvil y de guardarlas en la base de datos. Es responsable de realizar una serie de chequeos para conservar sólo datos válidos. Mientras que la funcionalidad responsable de la “obtención de resultados”, se encarga de devolver al servidor una lista de resultados para una determinada actividad de un usuario.

La API REST está implementada en Symfony¹⁶, Express¹⁷ y NodeJS¹⁸.

4 Caso de uso

Como caso de uso de la arquitectura y los componentes propuestos en este trabajo, se tomó la necesidad existente en el Programa E-Basura de la UNLP¹⁹, el cual aborda la problemática de los Residuos de Aparatos Eléctricos y Electrónicos (RAEE), transformándolos en acciones para la inclusión digital, la equidad social y la protección ambiental. Una de estas acciones, consiste en recibir equipamiento del sector de las TIC en desuso, el cual luego de un proceso de reutilización, reparación y reciclado de tecnología es donado a diversas instituciones educativas y/o de bien público y sin fines de lucro. En la actualidad, este Programa cuenta con un formulario Web²⁰, en el cual quien quiere donar elementos RAEE, lo completa para que miembros del Programa se pongan en contacto para retirarlos. Este formulario, no siempre resulta ágil, y ante la posibilidad de contar con una aplicación móvil para recibir este tipo de información, enriquecida con datos relevados a partir de las ventajas adicionales de los dispositivos móviles, es que se trabajó de manera conjunta para generar una propuesta de workflow que considere aspectos del formulario y sume otra información relevante como, por ejemplo, fotos de los elementos que serían donados. Fotos que fueran tomadas en el lugar en donde se encuentran los RAEE, lo que puede permitir a los

¹⁶ <https://symfony.com/>

¹⁷ <https://expressjs.com/>

¹⁸ <https://nodejs.org>

miembros del Programa, tener noción del espacio físico donde retirarán los residuos y de posibles complicaciones, por ejemplo, cuando los residuos se encuentran a gran altura o en zonas de difícil acceso por la acumulación de otros elementos en desuso.

Siguiendo las etapas de creación de actividades propuestas en [16] se generó el co-diseño conceptual de la actividad con las variantes necesarias para enriquecer la recolección de datos y considerando en este caso, que la posición donde se realice cada tarea no es relevante. De esto modo, se generó un workflow con 22 tareas de diferentes tipos²¹. Estas tareas, fueron planificadas de manera bifurcada (selección múltiple/exclusiva), usando un workflow con transiciones (algunas de ellas condicionales). La estructura general de la planificación se visualiza en la Fig. 7.

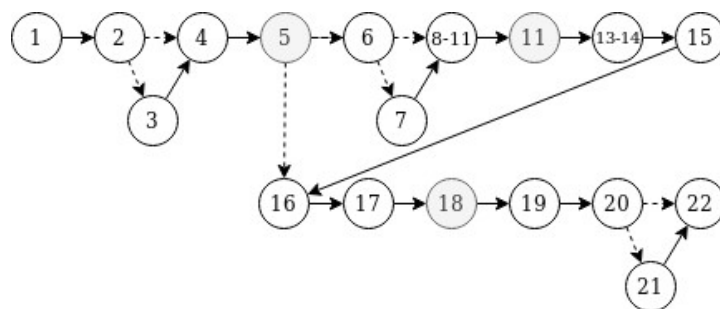


Fig. 7. Actividad conceptual para entregar elementos RAEE

En la Fig. 7, se puede observar con círculos cada una de las tareas, con línea continua las transiciones no condicionales entre tareas, en este caso, la tarea 1 es obligatoria y consiste en completar datos de contacto. Una vez realizada la tarea 1, se realizará la tarea 2. Por otro lado, las líneas punteadas, indican transiciones condicionales, por ejemplo, la tarea 2 implica seleccionar si la donación la realiza una persona particular o una institución. Dependiendo de la elección que realice el usuario en este punto, se pasará a la tarea 3 o a la tarea 4 respectivamente.

El co-diseño conceptual, se usa como base para luego crear desde la plataforma Web propuesta, las tareas que aún no existan y luego definir el workflow que podrá ser ejecutado desde la aplicación móvil.

La Fig. 8, muestra cómo se crea la tarea 1 desde la plataforma Web.

Una vez disponibles todas las tareas con las que se trabajarán, se procede a crear o a reusar la estructura de un workflow. Para este ejemplo, se creará uno nuevo.

¹⁹ Programa E-Basura. <https://E-Basura.unlp.edu.ar/>

²⁰ <https://encuestas.linti.unlp.edu.ar/index.php/369463/lang-es>

²¹ Más información de las tareas del workflow disponible en: https://docs.google.com/document/d/1KzN61x10Cq6az-_51jx9e1sJmX_51LspXw0ymWk_OeM/edit#heading=h.v9wjnku9vg0s

Fig. 8. Creación de una tarea para ingresar datos

La Fig. 9 muestra cómo se crea una transición condicional desde la tarea 2 a la tarea 3 desde la plataforma Web.

Fig. 9. Creación de una transición condicional en el workflow

Una vez creado el workflow desde la plataforma Web y puesto como disponible para los usuarios, se le comparte un código QR de acceso para que éstos, desde la aplicación móvil, lo accedan. Cuando el usuario accede al código QR de inicio desde la aplicación móvil, el workflow quedará disponible para ser ejecutado (Fig. 10.a). Luego de esto, el usuario, podrá ejecutar el workflow desde su aplicación móvil (Fig. 10.b) y comenzar a realizar la tarea inicial del workflow (Fig. 10.c)



Fig. 10. Workflow descargado en la aplicación móvil del usuario

Con lo previamente descripto, quedan presentadas algunas de las pantallas relevantes relacionadas a la creación de tareas, gestión de un workflow y su posterior acceso y ejecución desde la aplicación móvil.

5 Discusión

El caso de uso ha sido realizado considerando algunos de los datos presentes en uno de los formularios online existentes en el Programa E-Basura de la UNLP. El workflow generado desde la plataforma Web ha sido ejecutado desde la aplicación móvil por miembros del Programa y, luego de esto, su experiencia de usuario ha sido reportada como satisfactoria. En este punto, los miembros consideraron que el workflow generado para ser ejecutado desde la aplicación móvil, es más sencillo de usar que su versión en formulario online, esto se debe a que las tareas son más guiadas (debido a las condiciones para definir los datos a recolectar) y por permitir recolectar otro tipo de datos (por ejemplo, audio y fotos in-situ). Datos que, con el formulario online, no eran posible recolectar.

Una necesidad similar respecto a la recolección de datos, puntualmente para reciclado de computadoras, es la que tiene el proyecto “Recicla tu Compu, Recicla tu Mundo”²² de la Facultad de Informática, de la UNLP. Miembros de este proyecto, también han colaborado con pruebas preliminares y han emitido un juicio de similares características a las mencionadas por los miembros del Programa.

En el marco del aislamiento preventivo social obligatorio (DECNU-2020-297 válido a partir del 19 de marzo de 2020), a los autores de este trabajo, no les ha sido posible realizar pruebas con los usuarios de las instituciones que habitualmente donan al Programa, sin embargo, éstos esperan que la experiencia preliminar de los usuarios

²² <http://sedici.unlp.edu.ar/handle/10915/91462>

consultados hasta el momento sea replicada en los diferentes usuarios de las instituciones. Uno de los fines de las pruebas con usuarios es lograr retroalimentar la propuesta para que el impacto en la sociedad se multiplique satisfactoriamente y constituya una solución aplicable en más dominios.

6 Conclusiones y Trabajos Futuros

En este trabajo se presentó el desarrollo preliminar de una plataforma Web que permite crear, clonar y gestionar workflows para realizar actividades de recolección de datos y de una aplicación móvil que permite ejecutarlos; así como de una API REST que comunica ambas componentes. El foco del trabajo se pone en proponer una solución adecuada para usuarios sin conocimientos de programación y en permitir el reuso de tareas y de workflows para una mejor circulación del conocimiento y de las experiencias de los usuarios. El desarrollo considera una arquitectura distribuida basada en microservicios. Se presentó también un caso de uso donde la arquitectura propuesta y sus componentes se usan para definir una actividad de recolección de datos co-diseñada con miembros del Programa E-Basura de la UNLP. Este Programa contempla los problemas sociales y ambientales; contribuye con la alfabetización digital, la reducción de la brecha social de sectores vulnerables de la sociedad y con la disposición final segura para Residuos de Aparatos Eléctricos y Electrónicos (RAEE).

Si bien el presente trabajo, se encuentra en proceso, el mismo está en una etapa avanzada de desarrollo y ha sido validado, de manera preliminar, con un grupo de usuarios. Se espera realizar más pruebas con usuarios finales en el corto plazo de, por un lado, la plataforma Web como herramienta de autor para crear, clonar y gestionar workflows, y por el otro, pruebas de usuario con voluntarios recolectando datos mediante la ejecución de los workflows creados desde la plataforma Web.

Por otro lado, se espera agregar un módulo para la ejecución de los workflows desde la plataforma Web y sumar la edición colaborativa para la co-creación (diseño participativo) por un lado de las tareas y por otro de los workflows.

Agradecimientos. A los miembros del Programa “E-Basura” de la UNLP y a los miembros del Proyecto “Recicla tu Compu, Recicla tu Mundo” de la Facultad de Informática, UNLP., por el apoyo para realizar pruebas preliminares de usuarios.

Referencias

1. Morris, C. A., Deochand, N., & Peterson, S. M. (2018). Using Microsoft Excel® to build a customized partial-interval data collection system. *Behavior analysis in practice*, 11(4), 504-516.
2. Cochero, J. (2018). AppEAR: Una aplicación móvil de ciencia ciudadana para mapear la calidad de los hábitats acuáticos continentales.
3. Balsalobre, A., Ceccarelli, S., Cano, M. E., Ferrari, W. A., Cochero, J., & Martí, G. A. (2018). GeoVin: Ciencia Ciudadana para aprender de las vinchucas de Argentina. In

II Congreso Argentino de Ciencia Abierta y Ciudadana (CIACIAR)(Universidad Nacional de San Martín, 2 de noviembre de 2018).

4. Kelvin, A. A., Banner, D., Pamplona, L., Alencar, C., Rubino, S., & Heukelbach, J. (2016). ZIKATracker: A mobile App for reporting cases of ZIKV worldwide. *The Journal of Infection in Developing Countries*, 10(02), 113-115.
5. 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).
6. Aanensen, David M., et al. "EpiCollect+: linking smartphones to web applications for complex data collection projects." *F1000Research* 3 (2014).
7. Lliteras A.B., Grigera J., Corporaal F.R.M., Di Claudio F., Gordillo S.E. (2020) A Flexible Web Authoring Tool for Building Mobile Learning Experiences. In: Pesado P., Arroyo M. (eds) Computer Science – CACIC 2019. CACIC 2019. Communications in Computer and Information Science, vol 1184. Springer, Cham
8. Sprinks, J., Wardlaw, J., Houghton, R., Bamford, S., & Morley, J. (2017). Task Workflow Design and its impact on performance and volunteers' subjective preference in Virtual Citizen Science. *International Journal of Human-Computer Studies*, 104, 50-63.
9. Kunze, Matthias, and Mathias Weske. Behavioural models: From modelling finite automata to analysing business processes. Springer, 2016.
10. Aanensen, David M., et al. "EpiCollect: linking smartphones to web applications for epidemiology, ecology and community data collection." *PloS one* 4.9 (2009).
11. Bonney, R., Cooper, C. B., 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), 977-984.
12. Simpson, Robert; Page, Kevin R.; De Roure, David. Zooniverse: observing the world's largest citizen science platform. En *Proceedings of the 23rd international conference on world wide web*. 2014. p. 1049-1054.
13. Trouille, Laura; Lintott, Chris J.; Fortson, Lucy F. Citizen science frontiers: Efficiency, engagement, and serendipitous discovery with human-machine systems. *Proceedings of the National Academy of Sciences*, 2019, vol. 116, no 6, p. 1902-1909.
14. Hartung, Carl, et al. Open data kit: tools to build information services for developing regions. En *Proceedings of the 4th ACM/IEEE international conference on information and communication technologies and development*. 2010. p. 1-12.
15. Lliteras, A. B., Grigera, J., dal Bianco, P. A., Corporaal, F. M., & Gordillo, S. E. (2018, October). Challenges in the design of a customized location-based mobile learning application. In *2018 XIII Latin American Conference on Learning Technologies (LACLO)* (pp. 315-321). IEEE.
16. Lliteras, A. B., Challiol, C., & Gordillo, S. E. (2017, October). Location-based mobile learning applications: a conceptual framework for co-design. In *2017 Twelfth Latin American Conference on Learning Technologies (LACLO)* (pp. 1-8). IEEE.