

Sistema de Apoyo a la Toma de Decisiones para la Gestión de RSU en Chubut

Leo Ordinez¹, Emmanuel Zambrano¹, Laura Santander¹, Gustavo Diaz de Quintana², Sebastián De Pablo²

¹Universidad del Chubut, 9103 Rawson, Argentina,

²Consortio GIRSU Virch-Valdés, 9100 Trelew, Argentina,

ldordinez@udc.edu.ar, jezambrano@udc.edu.ar, lasantander@udc.edu.ar ,
diazdequintanagustavo@gmail.com, sebastiandepablo@gmail.com,
<http://udc.edu.ar> - <http://consorciogirsu.com.ar/>

Resumen Uno de los grandes desafíos que afectan a la población mundial actualmente tiene que ver con el manejo de los residuos. En particular, la gestión integral de los Residuos Sólidos Urbanos representa un sistema de manejo que busca reducir la cantidad de desechos que deben ir a disposición final.

En este trabajo se reportan las experiencias recolectadas en el desarrollo de un sistema de software, que le permita a los diferentes actores incorporar todos los datos asociados a estos procesos, transformarlos en información de gestión, y lo más importante, generar conocimiento para la toma de decisiones orientada a la mejora continua de los mismos.

1. Introducción

Uno de los grandes desafíos que afectan a la población mundial actualmente tiene que ver con el manejo de los residuos. En particular, la gestión integral de los Residuos Sólidos Urbanos (RSU) representa un sistema de manejo que busca reducir la cantidad de desechos que deben ir a disposición final. En la Argentina, según datos del Ministerio de Ambiente y Desarrollo Sustentable de la Nación [6], más del 90 % de los hogares se encuentra cubierta con un sistema de recolección de residuos. Mientras que, el promedio nacional respecto a disposición adecuada de RSU alcanza el 61 %. En particular, Chubut dispone adecuadamente el 53 % de sus RSU.

El proyecto aquí presentado afecta a la gestión de los RSU en la denominada comarca VIRCH-Valdés, que comprende las localidades de Rawson, Trelew, Gaiman, Dolavon y Puerto Madryn, abarcando una población de 250.000 habitantes aproximadamente, en un recorrido interurbano de 200 km. La situación hasta el año 2013 de la Gestión de RSU en el área, se caracterizaba por la existencia de basurales municipales e informales a cielo abierto, en áreas vecinas a las plantas urbanas de la Comarca. Estos emplazamientos no se encontraban bajo ningún control en cuanto a emisiones de gases, percolado de líquidos lixiviados, proliferación de vectores y capacidad de recepción de residuos. En el caso particular de la ciudad de Trelew, el sistema se encontraba totalmente colapsado, ya que

la disposición final de residuos ocupaba toda la extensión del terreno disponible para ello. Asimismo, Puerto Madryn poseía su basural en el recorrido turístico que lleva a la Península de Valdés y Rawson a la vera de la Ruta Provincial N° 1, que comunica ambas ciudades por el corredor costero.

El Consorcio Público Intermunicipal de Gestión de Residuos Sólidos Urbanos “GIRSU VIRCH Valdés” es un ente público no estatal creado por Ley Provincial e integrado por los municipios de Trelew, Puerto Madryn, Rawson, Gaiman y Dolavon. El Sistema GIRSU cuenta con dos plantas de separación y tratamiento (PST) en Trelew y Puerto Madryn, donde trabajan cerca de 30 cooperativistas en cada una. A la vez, el Consorcio cuenta con un Centro de Disposición Final (CDF), en el que se disponen los residuos luego de su tratamiento. Los RSU se someten a distintos procesos, llevados adelante por diferentes actores: recuperación, clasificación, separación, acopio, venta y disposición final. La empresa Urbaser S.A. es la concesionaria del servicio de transporte de residuos entre las PST y el CDF, así como de su disposición en el CDF. Las cooperativas son contratadas directamente por el consorcio. En las PST se realiza un trabajo en conjunto entre las cooperativas y Urbaser S.A. Las cooperativas de trabajo realizan la separación de residuos en la PST de Trelew y de Puerto Madryn, respectivamente.

Aquí se reportan las experiencias recolectadas en el desarrollo de un sistema de software, que le permita a los diferentes actores (cooperativistas, concesionarios, municipios y consorcio intermunicipal) incorporar todos los datos asociados a estos procesos, transformarlos en información de gestión, y lo más importante, generar conocimiento para la toma de decisiones orientada a la mejora continua de los mismos. De este modo, se pretende abordar la construcción de conocimiento [2], manifestado a través de indicadores estadísticos, de los procesos y actores involucrados en la gestión de RSU. La distribución geográfica, el tipo de actividad desarrollada y el volumen de residuos procesados mensualmente (5.000 Tn) requieren un esquema superior al de un sistema de gestión tradicional. En particular, se busca que, mediante la construcción de conocimiento, basado en información coherente, consistente y disponible, se fortalezcan y optimicen las tareas realizadas por todos los actores involucrados en la cadena productiva del Consorcio (cooperativas de trabajo, proveedores, transportistas y compradores).

El objetivo de este trabajo es presentar en forma sistematizada las experiencias recolectadas en el desarrollo de un sistema para el soporte de decisiones, en el dominio de aplicación de la Gestión de RSU, mediante la construcción de indicadores de desempeño de los procesos.

El resto del trabajo se organiza de la siguiente manera: la Sección 2 presenta el enfoque metodológico utilizado para el desarrollo del proyecto; la Sección 3 expone las Historias de Usuario y requerimientos generales del sistema; en la Sección 4 se resumen los aspectos más destacados del proceso de desarrollo, lo cual incluye la infraestructura de soporte, el modelo arquitectónico y las vistas, entre otros; la Sección 5 aborda la construcción de conocimiento mediante indicadores estadísticos; y finalmente se exponen las conclusiones en la Sección 6.

2. Enfoque Metodológico

El tipo de organización descrita en la Sección 1 plantea un escenario favorable para el desarrollo de un proyecto de vinculación Universidad-Estado-Empresa [13]. En este sentido, el Consorcio solicitó asistencia a la Universidad del Chubut (UDC), mediante la suscripción de un Convenio para contratar pasantes de la Tecnicatura Universitaria en Desarrollo de Software. Estos pasantes, bajo la doble tutoría de un docente de la UDC y un miembro del Consorcio, llevarían adelante el proyecto.

Debido a la naturaleza del trabajo, la cual imponía la materialización de resultados parciales en forma temprana e incremental; la dinámica de las organizaciones involucradas; la disponibilidad de los tutores y las características de los desarrolladores, se eligió como metodología de trabajo Scrum [15][16].

Scrum adopta sus ideas del trabajo orientado a la pila de producto a desarrollar. No utiliza métodos jerárquicos para el desarrollo de actividades sino que se centra en las responsabilidades del equipo de proyecto. Esta visión es más adaptable a un proyecto de desarrollo dinámico y orientado a resultados, ya que el criterio del mismo es lograr alcanzar los objetivos más que producir determinado volumen de trabajo en un día.

La Pila de Producto (*Product Backlog*) es el corazón de Scrum. Básicamente, es una lista priorizada de requisitos. Metas que se quieren alcanzar, componentes que se quiere construir, descritos usando la terminología del usuario. Se las llama *Historias de Usuario (User Stories)*. La Pila de la Iteración (*Sprint Backlog*) son los requisitos comprometidos para el ciclo de trabajo que durará entre una y cuatro semanas según lo convenido con el Dueño del Producto (*Product Owner*).

Una Iteración se inicia con la planificación de la Pila de esa Interacción. Ésta debe ser realizable y consensuada entre todas las partes. Durante la ejecución de la Iteración se mantiene un Tablero Scrum, que muestra el estado de cada Historia de Usuario y su avance. El progreso de la Iteración se mantiene a través de reuniones diarias. A continuación se muestran los *roles* de Scrum y sus responsables en el proyecto.

- Equipo de Desarrollo \Rightarrow *Pasantes UDC*
- Dueño del Producto \Rightarrow *Tutor Consorcio*
- Facilitador \Rightarrow *Tutor UDC*

La planificación general del proyecto se organizó en concordancia a la metodología a utilizar. En base a estimaciones previas realizadas entre los tutores (*Facilitador* y *Dueño del Producto*), se estableció un tiempo total de trabajo de cuatro meses. A la vez, de acuerdo a lo propuesto por Scrum, se realizarían entregas periódicas de software, con una frecuencia no superior a tres semanas. Esto determinó la realización de ocho iteraciones aproximadamente.

Scrum no es una metodología que imponga una gran cantidad de *Artefactos* o *Ceremonias*. Aunque sí es estricta en los pocos que establece. A fin de gestionar el proyecto con Scrum, se adoptó la herramienta Trello [17]. Esta herramienta permite el manejo del Tablero mediante una aplicación para teléfonos celulares; el establecimiento de etiquetas en cada Historia de Usuario, que identifiquen a

qué Iteración pertenecen; la definición de una fecha de vencimiento; la asignación a uno o varios miembros del Equipo de una Historia de Usuario; la posibilidad de adjuntar archivos a las Historias y la capacidad de incorporar Criterios de Aceptación a cada Historia, entre otras. Respecto al control del proyecto, Scrum adopta una herramienta denominada *Burndown chart*, el cual muestra cómo disminuye la cantidad de Historias de Usuario a medida que transcurren las Iteraciones. Para ello se utilizó un complemento de Trello, denominado Burndown for Trello [5].

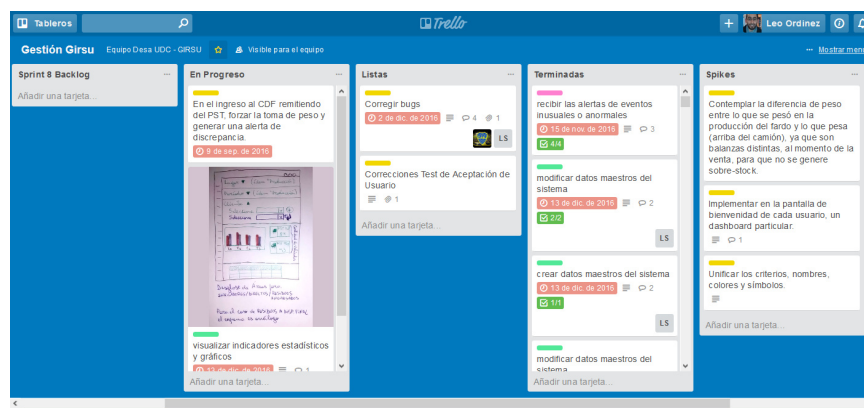


Figura 1. Captura de tablero en Trello

3. Historias de Usuario

Inicialmente se presentaron los siguientes requerimientos, los cuales luego fueron *transformados* en Historias de Usuario.

- Registrar el ingreso de materia a las plantas y al relleno. Tener en cuenta que puede no consignarse peso, pero sí todos los otros datos. En ese caso, hay que indicar el motivo de por qué no se registró el peso.
- Registrar el egreso de materia a las plantas y al relleno.
- Registrar la producción diaria de fardos.
- Registrar el peso diario de material recuperado suelto.
- Registrar la asistencia de los cooperativistas. Registrar los ingresos. Imprimir a media mañana los ingresos. Hacer firmar planilla de ingresos por el responsable de la cooperativa. Validar los datos de ingreso. Registrar los egresos. Al día siguiente imprimir la planilla de egresos del día anterior. Hacer firmar la planilla del día anterior por el responsable de la cooperativa. Validar los datos de egresos. Contemplar los permisos de salida en el segundo formulario.
- Registrar las ventas de material recuperado. Una venta tiene: registro de pesada, fecha, tipo de material, cantidad, precio por unidad y costo total.

- Generar los remito de salidas de ventas.
- Llevar un historial de precios de venta por producto. El producto tiene un “listado histórico de precios”.
- Control de Stock de materiales. Incluye todo el material que hay en las PST.
- Ajuste de Stock por despacho de materiales cedidos a la cooperativa.
- El alta de un nuevo material, transporte, etc., debe poder ser realizada por el operador inmediatamente y luego validada por un supervisor.
- Las ventas pueden ser la disposición de residuos. Esto vale para los privados.
- Registrar si las ventas están canceladas. Cada venta se registra, pasa a estado pendiente de pago, al pagarse pasa a estado pagada o cancelada.
- En el CDF se debe indicar cuánto residuo ingreso y a qué celda va.
- Registrar en el ingreso y egreso si es un transporte “del Girsu” o de Tercero.
- Armar un “Datos Maestros” para que los pueda cargar el Administrador.
- Armar un “Datos Maestros” para que los pueda cargar el Supervisor.
- No se borran datos. Se cambia el estado.
- Agregar controles de validación de datos. Por ejemplo, peso excesivo, etc.
- Todo lo que se registra debe tener un campo observaciones, ya sea que se muestre o no.
- Generar “remito control” para el traslado de material desde PST a CDF.
- Generar Alertas al Supervisor en caso de eventos anormales (peso, tiempo, etc.) .
- Generar un reporte con datos estadísticos configurables.

Los requerimientos iniciales se transformaron en 49 Historias de Usuario. Estas Historias se distribuyen entre las Iteraciones de acuerdo al Cuadro 1.

Iteración	1	2	3	4	5	6	7	8
Historias	10	10	4	6	8	6	3	12

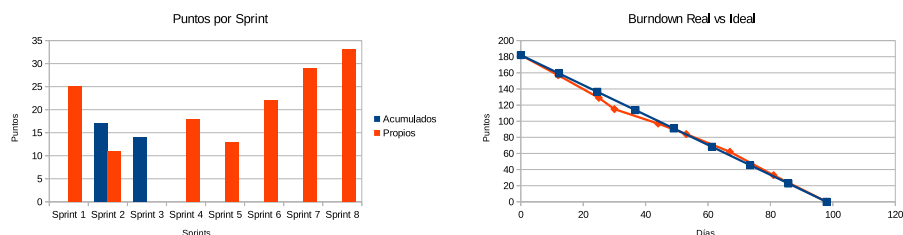
Cuadro 1. Historias de Usuario por Iteración

Se debe notar que la suma de los valores del Cuadro da como resultado 59, en lugar de 49. Esto se debe a que algunas Historias no fueron completadas en una Iteración y pasaron a la siguiente, con lo cual se cuentan dos veces. En particular, esto ocurrió en la Iteración 2 y la 3. Por otro lado, por razones de espacio no se consignan aquí las 49 Historias de Usuario. No obstante, a continuación se listan las *Épicas* en que se agruparon las Historias de Usuario.

- **Como Gerente quiero** gestionar cooperativas y cooperativistas **para** llevar un registro de los trabajadores involucrados en el proceso.
- **Como Supervisor quiero** administrar las asistencias de los cooperativistas **para** mantener un proceso productivo estable con retribuciones acordes.
- **Como Supervisor quiero** registrar el proceso de ingreso de material desde las ciudades a las PST **para** controlar de manera eficiente el proceso de gestión de RSU.

- **Como Supervisor quiero** registrar el proceso de egreso de material de las PST al CDF **para** controlar de manera eficiente el proceso de gestión de RSU.
- **Como Gerente quiero** gestionar los clientes del Consorcio **para** registrar con quiénes se comercia el material recuperado y los servicios del Consorcio.
- **Como Gerente quiero** administrar el proceso de ventas de material recuperado y de servicios **para** llevar un estado preciso de la contabilidad del Consorcio en ese aspecto.
- **Como Usuario del sistema quiero** contar con mecanismos automáticos de generación de alertas y controles de acceso **para** asegurar la consistencia, confiabilidad y seguridad de los datos manipulados.
- **Como Gerente quiero** disponer de Datos Maestros e indicadores estadísticos **para** facilitar la utilización del sistema en la toma de decisiones.

Una vez definidas las Historias de Usuario, Scrum establece que se deben estimar las mismas para valorarlas y determinar el esfuerzo requerido para realizarlas. Existen diferentes técnicas para la estimación, aunque la más utilizada es la denominada *Poker Planning* [9][8]. En esta técnica se el Equipo de Desarrollo y el Facilitador realizan rondas de votaciones sobre cada Historia, asignándole un valor de la Sucesión de Fibonacci (*i.e.*, 1, 2, 3, 5, 8, 13, etc.). Las rondas se repiten hasta lograr un consenso en el valor. Cuando se produce una discrepancia, los participantes que votaron en forma extrema (mayor y menor) presentan sus argumentos y luego se vuelve a votar. A continuación se presentan los valores de las puntuaciones de cada Historia. En la Figura 2(a) se muestran los puntos totales de cada Iteración. En la Figura 2(b) su muestra la evolución de la carga de trabajo. El desagregado de puntos por iteración se detalla en el Cuadro 2.



(a) Puntos por iteración.

(b) Gráfico de *burndown*.**Figura 2.** Evolución de la carga de trabajo.

Iteración 1	3	2	3	3	2	3	2	2	3	2		
Iteración 2	3	3	3	1	2	5	3	3	2	3		
Iteración 3	5	3	3	3								
Iteración 4	3	2	1	5	5	2						
Iteración 5	3	1	1	2	1	1	3	1				
Iteración 6	3	5	5	3	3	3						
Iteración 7	8	8	13									
Iteración 8	3	3	3	3	2	3	2	2	3	3	3	3

Cuadro 2. Desagregado de puntos de cada Historia por Iteración

4. Desarrollo del Sistema

4.1. Stack Tecnológico

La arquitectura de soporte del sistema se basa en el patrón *Model-View-Controller*, con una variante para manejar de manera más eficiente el ruteo de la aplicación. Esto se debe que se utiliza el framework Laravel [10] como componente central de la infraestructura de software. Respecto a las tecnologías, se enumeran a continuación las principales: Apache, PHP, MySQL, Bootstrap3, JavaScript, CSS3, HTML5, JQuery y paquetes afines para propósitos específicos.

4.2. Modelado Arquitectónico

Desde el punto de vista arquitectónico, se siguió el principio de *Diseño emergente*. Este enfoque se adapta mejor a la metodología Scrum, que uno más clásico como el propuesto por el Proceso Unificado de Desarrollo [7]. El *Diseño emergente* plantea que la arquitectura se debe realizar en el momento que es necesaria para poder entregar valor al cliente [3], [4], [18]. En este sentido, se distinguen dos características que determinan el hecho de emerger: 1) el surgimiento repentino, el cual tiene que ver con la aparición casi-espontánea de la necesidad de modelar arquitectónicamente una determinada situación; y 2) el diferimiento o aplazamiento de una tarea, hasta que sea absolutamente necesaria para la continuidad del trabajo.

En base a lo anterior, en el caso del proyecto, se vio el surgimiento de la arquitectura en forma gradual, parcial y repentina, según la misma fue requerida. Esto involucraba su enfoque estático y dinámico. La metáfora que ejemplifica esta práctica es la de un rompecabezas que se va completando, pero hasta la última pieza que se pone no se sabe exactamente de qué figura se trata.

En vista del esquema de vinculación planteado en la Sección 2 y las personas que llevaban a cabo los diferentes roles, surgió como una necesidad que el Facilitador haga también las veces de *Arquitecto* y tenga un gran peso en la definición de la arquitectura del sistema. Este aspecto es una limitación inherente de la metodología, la cual asume que el Equipo de Desarrollo cuenta con la experiencia suficiente para el abordaje de tareas de análisis y diseño. A fin de mantener una alta dinámica en el trabajo y detectar situaciones potencialmente

conflictivas a tiempo -sobre todo relacionadas con el modelado de los datos y el diseño de la arquitectura-, se estableció un canal de comunicación a través de un grupo de WhatsApp. Este medio de comunicación permitió fácil y ágilmente compartir bocetos, diagramas, *mock-ups*, etc.

4.3. Vistas y Pantallas

La utilización del framework Laravel y su motor de plantillas *Blade*, en el desarrollo de un sistema que maneja diferentes tipos de datos e información, organizada en forma de tablas, favoreció la tarea de desarrollo. No obstante, ese favorecimiento trae aparejadas algunas limitaciones que el mismo framework impone, como el compartir datos entre varias vistas o componer una vista a partir de diferentes modelos de datos [11]. Esto fue una situación que requirió en determinados momentos intervenciones especiales y estrategias para poder continuar.

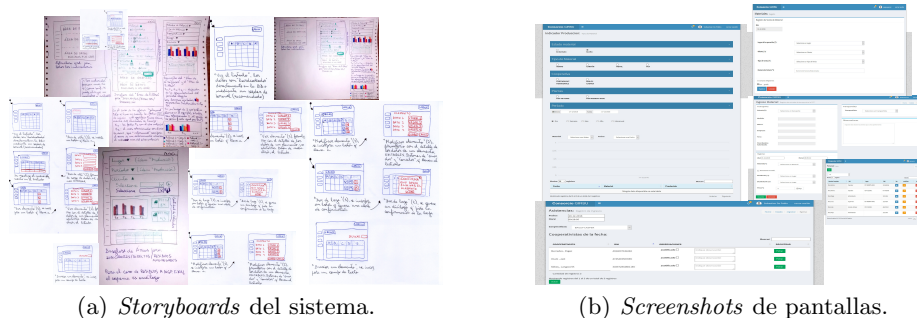


Figura 3. Diseño e implementación de pantallas.

Un aspecto destacado, es la situación análoga a la presentada en el caso de la arquitectura, en la que el Facilitador también actúa como Arquitecto. En este caso de las vistas, pantallas y la cuestión de Interfaz de Usuario, en general. Al igual que el caso de la arquitectura, esto se vió potenciado por el uso de WhatsApp. Una técnica que ayudó mucho el desarrollo de las interfaces gráficas y clarificó las cuestiones estructurales fue la de *storyboarding* [12]. En la Figura 3(a) se muestran algunos de los bocetos realizados y la Figura 3(b) se muestran capturas de pantallas del sistema.

5. Generación de Conocimiento

Como se mencionó en la Sección 1, el propósito principal de este sistema es la generación de conocimiento no trivial acerca del proceso de gestión de RSU. Ese conocimiento se construye en base a información y experiencia sistematizada en

reglas de negocio. A la vez, la base de todo está asentada en los datos reales, que se recolectan de la operatoria diaria y que contextualizados hacen la información del proceso.

La manera elegida para la representación del conocimiento obtenido es a través de indicadores estadísticos, los cuales pueden ser graficados. La visualización de un indicadores en un Tablero de Control (*dashboard*), permite la toma de decisiones de manera más ágil y sencilla. En la Figura 4 se muestran los indicadores elaborados.

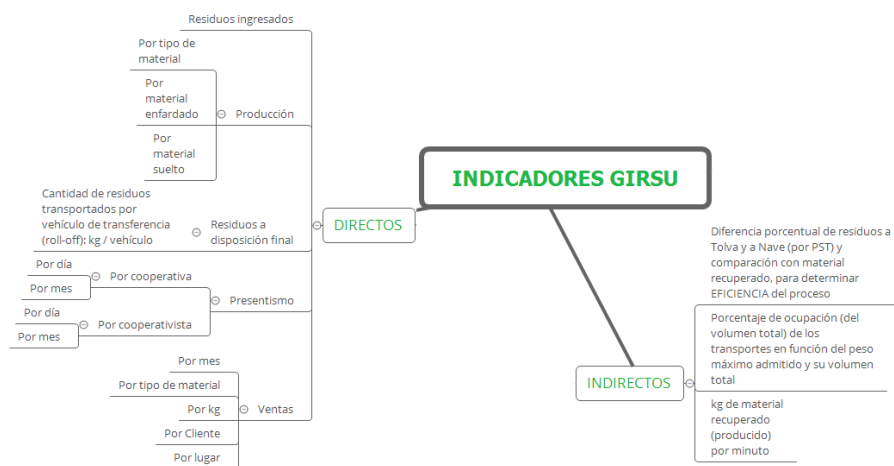


Figura 4. Indicadores del proceso

El modelo de indicadores de la Figura 4 representa un conjunto de instrumentos, metodologías y herramientas que permiten abordar desde un punto de vista cuali-cuantitativo el análisis y la evolución espacio-temporal del Sistema de Gestión de RSU VIRCH-Valdés, sus interrelaciones, y los diversos actores que intervienen. Este sistema de indicadores se basó en lo propuesto por [1] y [14].

El caso particular de los indicadores indirectos, éstos son los que permiten comprender mejor las relaciones dinámicas y los impactos que éstas tienen para la calidad de vida de la población y su influencia en el ambiente. A la vez, a partir de ellos, se toman las principales decisiones estratégicas y estructurales del Consorcio, como la compra de maquinaria, contratación de personal y ampliación de obras para el CDF.

6. Conclusiones y Trabajos Futuros

En este trabajo se reportó la experiencia obtenida en un proyecto de vinculación tecnológica, en el que intervinieron diferentes actores privados (cooperativas y sociedad anónima), un consorcio de Municipios y la Universidad del Chubut.

El trabajo consistió en el desarrollo de un sistema de apoyo a la toma de decisiones, mediante la generación de conocimiento no trivial del proceso de gestión de RSU. El trabajo abordó aspectos metodológicos, tecnológicos, específicos del dominio de aplicación y técnicos.

En cuanto a los próximos pasos, se está trabajando en la infraestructura de comunicaciones para el mejor soporte del sistema. A la vez, se planea abordar el estudio y desarrollo de nuevos indicadores que representen mayores niveles de conocimiento del proceso y de su impacto en la población. Así como en modelos de simulación, para analizar diferentes escenarios.

Referencias

1. ACUMAR: El sistema de indicadores de acumar (marzo 2014), <http://www.acumar.gov.ar/content/documents/3/1583.pdf>
2. Bellinger, G.D.C.A.M.: Data, information, knowledge, and wisdom (2004), <http://cife.stanford.edu/sites/default/files/TR048.pdf>
3. Bloomberg, J.: What is emergent about emergent architecture?, <https://intellyx.com/2015/09/28/what-is-emergent-about-emergent-architecture/>
4. Bradley, C.: An introduction to agile emergent architecture - always intentional, <http://www.scrumcrazy.com/An+Introduction+to+Agile+Emergent+Architecture+--+Always+Intentional>
5. Colombo, S.: Burndown for trello, <https://www.burndownfortrello.com>
6. de Ambiente y Desarrollo Sustentable, M.: Mapas críticos gestión de residuos marzo 2016 (2016), <http://observatoriorsu.ambiente.gob.ar/estadisticas>
7. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
8. Mahnič, V., Hovelja, T.: On using planning poker for estimating user stories. *J. Syst. Softw.* 85(9), 2086–2095 (Sep 2012)
9. Moløkken-Østvold, K., Haugen, N.C., Benestad, H.C.: Using planning poker for combining expert estimates in software projects. *Journal of Systems and Software* 81(12), 2106 – 2117 (2008)
10. Otwell, T.: Laravel, <http://laravel.com>
11. Otwell, T.: Laravel - views, <https://laravel.com/docs/5.1/views>
12. Pichler, R.: Agile scenarios and storyboards, <http://www.romanpichler.com/blog/agile-scenarios-and-storyboards/>
13. Sabato, J., Botana, N.: La ciencia y la tecnología en el desarrollo de américa latina. In: Herrera, A. (ed.) *América Latina: ciencia y tecnología en el desarrollo de la sociedad*. Editorial Universitaria, Santiago de Chile (1970)
14. Sarafian, D.R.: Indicadores para la gestión integral de residuos sólidos urbanos. In: *Reciclar 2016*. Ministerio del Interior, Rafaela, Argentina (2016)
15. Schwaber, K.: *Agile Project Management With Scrum*. Microsoft Press, Redmond, WA, USA (2004)
16. Stellman, A., Greene, J.: *Learning agile: understanding Scrum, XP, Lean, and Kanban*. O'Reilly (2014)
17. Trello, I.: Trello, <http://trello.com>
18. Vincent, M.: Emergent architecture – just enough just in time, <https://www.agilealliance.org/resources/sessions/emergent-architecture-just-enough-just-in-time/>