EIEI ACOFI 2025
5º Congreso Latinoamericano de Ingeniería
Más Información: www.acofi.edu.co/eiei
Ingeniería de impacto para cuidar y transformar la vida
ACOFI: 50 años por la excelencia de la educación en ingeniería.

Arquitectura de nube con aplicaciones web y móvil para el soporte de la administración de proyectos comunitarios rurales

Urbano Eliécer Gómez Prada

Universidad Industrial de Santander
Bucaramanga, Colombia

Resumen

El sector rural en Colombia enfrenta desafíos en términos de acceso a servicios básicos que son afrontados por las familias que lo habitan, las cuales se agrupan en divisiones territoriales más pequeñas que formalmente son conocidas como Junta de acción comunitaria rural. Para superar estos desafíos, las comunidades emprenden proyectos comunitarios que tienen dificultades en el acceso a herramientas para el manejo de información y trazabilidad que requiere la gestión de recursos en especial cuando hay apoyo económico de instituciones gubernamentales.

Este documento presenta una arquitectura de nube de baja complejidad en operación desplegada en AWS que apoya la administración de proyectos para estas juntas que fue desarrollada siguiendo lineamientos de un sistema orientado por procesos que también fue propuesto, el documento también enfatiza en la importancia de promover la adopción de la arquitectura.
La arquitectura está compuesta por una App para el registro síncrono o asíncrono de las actividades, una aplicación web comunicados por una API REST; la App cumple con procesos de uso genéricos que reducen la complejidad de la experiencia de usuario para brindar facilidad de adopción y mitigar la resistencia al cambio y el arraigo cultural de los usuarios.

El desarrollo de la arquitectura se enmarca en tres metodologías: Enfoque exploratorio-descriptivo para diseñar el sistema de gestión de las comunidades rurales, Design Science Research (DSR) para construir y evaluar artefactos y Rational Unified Process (RUP) para la implementación y hay una comparación con TRL como ejemplo para la Educación en Ingeniería.

Palabras clave: arquitectura de nube; modelo de administración de proyectos; comunidades rurales; sistemas orientados a servicios

DOI: https://doi.org/10.26507/paper.4844

Abstract

The rural sector in Colombia faces challenges in terms of access to basic services, which are dealt with by the families living in these areas. These families are organized into smaller territorial divisions formally known as Rural Community Action Boards. To overcome these challenges, communities undertake local projects, but obtén face dificultéis in accessing tools for information management and traceability—both essential for overseeing resources, especially when projects receive financial support from government institutions.

This document presents a low-complexity cloud architecture deployed on AWS, designed to support project management for these boards. It was developed following guidelines of a process-oriented system, which is also proposed in the document. Additionally, the document emphasizes the importance of promoting the adoption of this architecture.

The architecture consists of a mobile app for synchronous or asynchronous activity registration, a web application connected by REST API; the app is designed with generic usage processes to simplify the user experience, ensuring ease of adoption and addressing resistance to change and users' cultural attachment.

The development of the architecture is framed within three methodologies: An exploratory-descriptive approach to design the management system for rural communities, Design Science Research (DSR) to construct and evaluate artifacts, and Rational Unified Process (RUP) for implementation and a comparison with TRL is presented as an example for engineering education.

Keywords*: cloud architecture; project administration model**;** rural communities; service-oriented systems*

1 Introducción

Las Comunidades rurales en Colombia suelen ser gestionadas por grupos de personas que se agrupan en una Junta de acción comunal (JAC) y se enfrentan a desafíos específicos en términos de acceso a servicios básicos (Ndlela & Worth, 2021) según la autosuficiencia y sostenibilidad de sus medios de vida.

Este artículo describe una arquitectura de nube llamada SARA desarrollada en un proyecto de Investigación que sigue un sistema orientado por procesos para la administración de comunidades rurales que también fue diseñado en el proyecto y orienta la forma de usar las tecnologías de la información (TI) buscando claridad y facilidad en el registro de las actividades que surgen de los procesos que llevan a cabo las comunidades en su autogestión y por ende ayudar a resolver necesidades de control de proyectos que tienen apoyo de las instituciones territoriales como las alcaldías.

La arquitectura es una tecnología en la que utilizan servicios de computación de manera flexible y escalable a través de internet, en lugar de depender de servidores físicos locales (Condori, 2012), y en este caso está conformada por un aplicativo web, una API REST y una App móvil.

La App funciona de manera asíncrona y cuando hay acceso a internet se sincroniza para migrar los datos al servidor AWS al que también se accede mediante el aplicativo web, para generar los mismos informes que desde la App se generan por la JAC pero que, para la alcaldía junto con informes adicionales consolidados y/o gráficos de varias comunidades.

El sistema y la arquitectura de nube se pondrán a disposición de una comunidad y una alcaldía en un próximo proyecto, junto a una estrategia de apropiación que mitigue las carencias de información en la gestión de proyectos, esta estrategia, según Dirven y Candia (2020) es necesaria debido al arraigo cultural en lo rural es muy alto.

Adicionalmente, en el Anexo 1, hay un esquema de la metodología junto a una consideración de la importancia de la educación en Ingeniería, de una compara las metodologías de desarrollo, en este caso la de SARA, con los niveles del TRL *(Technology Readiness Level),* una escala que mide la madurez de una tecnología en niveles desde 1 al 9 y mide el esfuerzo de alcance de la investigación, el desarrollo y la innovación (I+D+i) (MI CIENCIAS, 2016).

2 Marco de Referencia

El mejoramiento con TI es una necesidad expresada por el Ministerio de TIC de Colombia (2018) y por el plan estratégico del departamento de Santander (Daza et al, 2017), región de Colombia desde donde surge esta iniciativa, en cuanto a que es necesario incorporar TI para mejorar la calidad de vida de los habitantes rurales y los resultados productivos y económicos.

Un comunidad rural es una unidad de población que habita en divisiones territoriales más pequeñas al municipio que, según Daza et al (2017), generalmente consisten en un grupo de habitantes de viviendas dispersas en zonas rurales, pueden tener su propia identidad, cultura y necesidades específicas, y a menudo están compuestas por familias que se dedican a actividades agrícolas, ganaderas u otras relacionadas con el entorno rural que requieren el apoyo de entidades territoriales para su desarrollo, en Colombia tienen una densidad poblacional inferior a 150 habitantes por kilómetro cuadrado y son conocidas como JAC (Hoyos & Verhelst, 2019)

Uno de los apoyos a brindar a las JAC es el mejoramiento del manejo de información que surge de los proyectos comunitarios que llevan a cabo para su autogestión específicamente en aquellos en los que hay apoyo económico de entes gubernamentales. En muchas JAC, las actividades que ejecutan en estoy proyectos carecen de TI lo cual disminuye la posibilidad de trazabilidad de los recursos en cada fase realizada.

La información la puede brindar los sistemas de información, un conjunto de varios componentes orientados a la gestión de datos (Pressman, 2014) que en este caso corresponde a una arquitectura de nube llamada SARA, conformada por el sistema de administración de comunidades rurales, que

integra proyectos, beneficiarios, ejecutores, presupuestos, auditorias, operaciones, documentos de trazabilidad, entre otros (Autor, 2025).

La arquitectura está conformada por una plataforma web con un API REST, una App y próximamente dispositivos IoT, que buscan brindar el apoyo que necesitan las JAC interconectando aplicaciones y diversas organizaciones para contribuir con el desarrollo en una región en donde se requiere un enfoque como el presentado por Diga & May (2016), es decir, una comprensión holística de cómo las partes interactúan para construir y usar las aplicaciones.

La integración de las aplicaciones se soporta en una arquitectura de nube, un modelo de diseño de software que utiliza servicios de computación para construir, implementar y gestionar aplicaciones y recursos TI de manera flexible y escalable a través de internet en el que, en lugar de depender de servidores físicos locales, se aprovechan recursos de proveedores con servidores remotos (Condori, 2012) como *Amazon Web Services* (AWS), *Microsoft Azure* o *Google Cloud Plataform* (GCP), esto permite acceder a recursos informáticos según demanda sin la necesidad de inversión inicial en infraestructura (Molina & Tapia, 2020).

La arquitectura es una tecnológica que se debe acompañar de una estrategia para la apropiación que involucre los beneficiarios. La Estrategia es una necesidad con mayor relevancia en el agro que busca mitigar brechas en el uso de TI (Álvarez & Bernal, 2017), no se trata solo poner disponible la tecnología, se debe generar la capacidad o competencia para usarla y se deben generar cambios en la forma de llevar a cabo procesos para mejorar la calidad y los resultados productivos (Daza, Hernández, Quijano, & Serna, 2017), para lograrlo, se debe buscar primero la adopción, la cual es baja en el sector rural, especialmente en los productores de pequeña escala (Gómez-Prada, Orellana, & Salinas, 2020).

3 Metodología

La investigación en la que se enmarca este trabajo incorpora tres metodologías:

1. Un enfoque de investigación exploratorio-descriptivo soportado en la comprensión de las comunidades rurales de donde surge el sistema de administración que actualmente está en desarrollo y que se describe por Gómez-Prada (2025) y que involucra municipios con los administradores que se asignan (casi siempre desde la secretaria de infraestructura), comunidades, habitantes que se pueden comprender como beneficiarios y gestores de la comunidad, los proyectos y las actividades que consumen recursos y usan herramientas.
2. *Design Science Research* (DSR) siguiendo planteamientos propuestos por Hevner, et al (2004), se enfoca en construir y evaluar artefactos (modelos, métodos, instancias, etc.) resolviendo problemas (DSR tiene dos ciclos principales: el ciclo de diseño y desarrollo de artefactos y el ciclo de investigación: evaluación y contribución teórica)
3. *Rational Unified Process* (RUP) usando casos de uso que permitían la comprensión y aprovechando que es iterativo e incremental a través de ciclos que siguen pasos adaptados según cambios requeridos (Jacobson, Booch, & Rumbaugh, 2000).

En resumen, para implementar el prototipo que se presenta se siguieron los siguientes pasos:

1. Definir requerimientos según el sistema de gestión de comunidades propuesto y los casos de prueba en donde se identifican los procesos a contemplar.
2. Especificar modelos que incluye las entidades con sus atributos y diagramas de actividades para el seguimiento según criterios y generar el prototipo uno del aplicativo web y de la App.
3. Implementar el código según las herramientas de programación escogidas.
4. Revisar el cumplimiento de los requerimientos.

4 Resultados

El sistema de administración propuesto y la arquitectura de nube generada son un prototipo que involucra el manejo de proyectos que se realizan en una JAC, se espera continuar con la integración de otros aplicativos tales como la gestión de la atención en salud o en educación, la gestión de actividades por predio o proyecto productivo, el mantenimiento a las diferentes herramientas de un predio, como un potrero entre otras.

El sistema interconecta *Municipios* están conformados por *Comunidades* en donde hay *Habitantes* que se pueden ver como *Beneficiarios* que pueden actuar como *Gestor*. Los beneficiarios requieren *Proyectos* en los cuales los beneficiaros realizan *Actividades* con diferentes *Herramientas* en las que se consumen ciertos *Recursos* para trabajar sobre diferentes *Elementos* (como alcantarillas, puentes, vías, caminos o edificaciones, entre otros). Cada Proyecto requiere el apoyo de un *Manager* que es asignado por el municipio quien es garante de los recursos que administra la comunidad y por ello debe contar con información que muestre la trazabilidad y logro del objetivo, el cual debe ser el *Beneficio* de la comunidad.

La arquitectura de nube está dirigida a las JAC, actualmente está en fase de disponibilidad y busca convertirse en una alternativa para apoyar el desarrollo rural, está enmarcada en un proyecto de investigación alineado a expectativas como las expresadas por Tambotoh, et al (2015), en cuanto a la premisa de que la utilización de las TI tiene la capacidad de abordar problemas de desarrollo en las zonas rurales.

Adicionalmente se espera integrar elemento de IoT como lectores de lluvia (pluviómetros) o de temperatura, que apoyen la gestión de la alcaldía en donde cada vereda tendría al menos un dispositivo para registrar periódicamente la información.

La adopción del sistema y de la arquitectura se debe dar por fases, una propuesta de, proceso de adopción es la siguiente (En la figura 7 del Anexo 1, hay esquema de la propuesta):

1. En la fase 0, es decir al iniciar, se tienen proyectos bajo la responsabilidad del presidente de la JAC y una alcaldía que no puede soportar en TI la administración de esos proyectos, pero que requiere información de estos.
2. En la fase 1, la mediada por la arquitectura (MPA), los representantes de la JAC y la alcaldía se conectan a través de la App y la Web para tener información de los proyectos.
3. En la fase 2, se espera la integración de otras Apps que gestionen otros sistemas diferentes al de proyectos comunitarios como salud, educación, entre otros.

4. Se espera una fase 3 en la que se integren otras comunidades u otros municipios que adopten el sistema y la arquitectura.

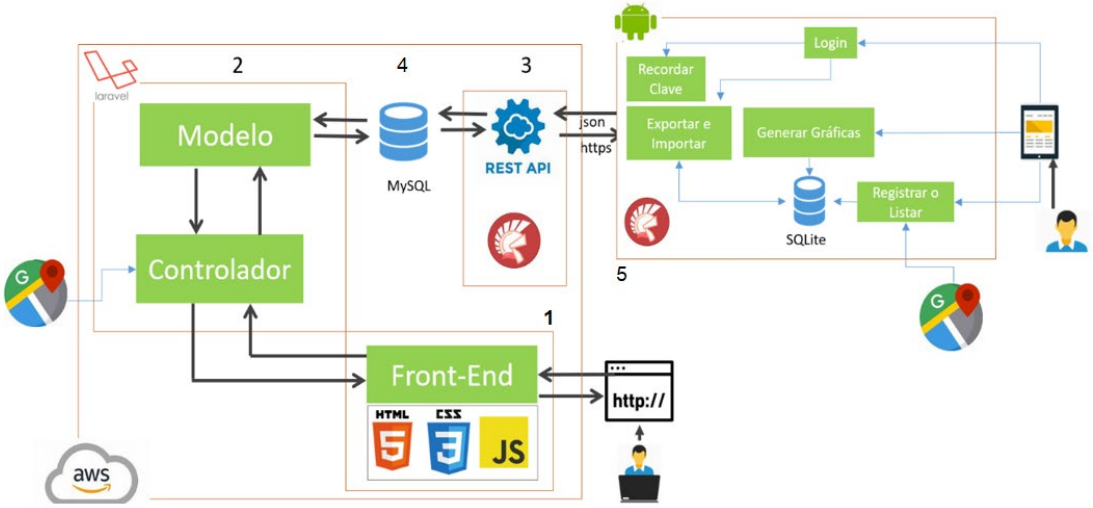
Los elementos principales de la arquitectura son presentados en la Figura 1, estos son descritos a continuación:

1. Front-End: es la parte de la aplicación web con la que el usuario interactúa directamente, incluyendo el diseño y la estructura visual.
2. Back-End: es la parte de la aplicación web donde se maneja la lógica del sistema, no es visible para el usuario y gestiona actividades del proyecto, como crear, leer, actualizar y eliminar (CRUD) tareas, responsables, fechas límite, etc.
3. API REST: Es una aplicación que permite la comunicación del Back-End con el Front-End y con las Apps, recibe y envía datos, recibe registros de actividad de los proyectos y envía datos sobre proyectos y datos maestros de la comunidad.
4. Base de datos en MySQL: Almacena maestros y operaciones de las JAC.
5. App: Es la aplicación móvil que recupera datos maestros de la API REST y permite el registro de actividades y el envío de datos al Back-End.

La Figura 1 también presenta el proceso de consulta de los datos recibidos desde la App con los informes descritos, en donde, tras la migración de los registros a la nube, el usuario puede ingresar a consultar listados y gráficos. La descripción del proceso que se da en la web es:

1. El navegador solicita a la vista los registros por http.
2. El controlador la recibe y ejecuta la sentencia SQL en el modelo.
3. El modelo hace consulta y entrega datos al controlador
4. El controlador suministra los datos a la Vista.
5. La vista entrega el resultado en el navegador.

Figura 1. Arquitectura de SARA



El sistema propone manejar por proyectos las actividades que se realizan en una JAC iniciando con la gestión de proyectos comunitarios, pero se espera integrar más aplicativos a la arquitectura

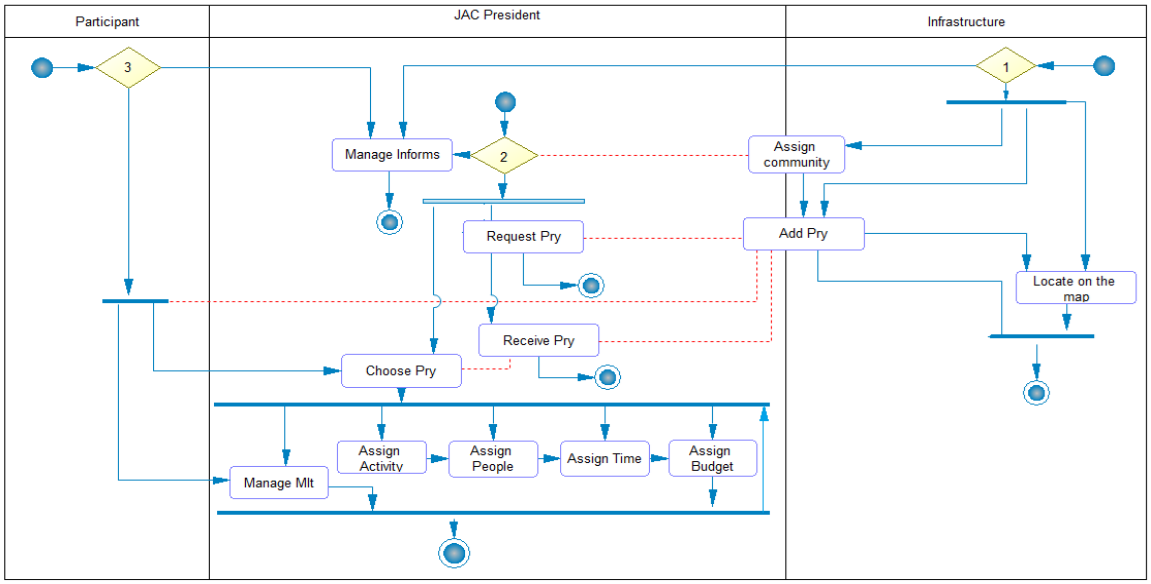
de nube con nuevas Apps que respondan a diferentes necesidades pero buscando que cada una siga los mismos pasos de operación de manera genérica en pro de mitigar la resistencia al cambio y el arraigo cultura, muy alto en habitantes rurales (Berman-Arévalo & Ojeda, 2020) (Gómez-Prada, Orellana, & Salinas, 2020) y así, facilitar el uso por parte de los campesinos. Un resumen de los requerimientos es (3 y 4 puede hacerse en móvil o web):

1. Gestionar maestros como| comunidades, beneficiarios, herramientas, tipos de actividad, municipios, entre otras.
2. Gestionar proyectos, incluye gestionar la ubicación vía *Google Maps*.
3. Gestionar Actividades
4. Generar informes de actividades por proyectos con filtros por comunidad, proyecto, fechas, actividad, observación; también se generan listados consolidados y gráficos de comunidades. actividades, participantes, dinero ejecutado, etc.

El proceso se describe en el diagrama de actividades de la 2, tiene tres inicios que se dan por momentos demarcados con cada uno de los rombos enumerados de cuando se dan:

1. La alcaldía asigna a la comunidad y le asigna un proyecto que puede ubicar sobre el mapa, lo cual le genera al líder la actividad pendiente de aceptar el proyecto (el primer paso es encontrar la necesidad, aunque no está en el software).
2. El presidente de la JAC asigna actividades, integrantes y/o herramientas, se habilita tras la recepción del proyecto asignado desde la alcaldía, de ahí adelante cuando vaya a registrar, escoge el proyecto que corresponda (puede tener varios proyectos activos).
3. El integrante de la comunidad puede reemplazar al presidente de la JAC para la asignación de actividades, aunque solo se puede dar si el presidente recibió el proyecto.

Figura 2. Diagrama de Actividades General



Se debe tener en cuenta que:

1. Las acciones de la JAC se realizan desde el móvil tras sincronizar los proyectos de la web, y viceversa, las actividades registradas en el móvil se sincronizan con el servidor.
2. El presidente o el integrante de la comunidad gestiona la multimedia escogiendo la actividad a la que le quiere hacer la asociación.
3. Cualquiera de los roles puede generar informes con alcance de su JAC o su municipio.

4.1 Aplicativo Web

En la Web se gestionan parámetros, proyectos, usuarios, comunidades, y se generan los informes tras la migración de datos. Estos son:

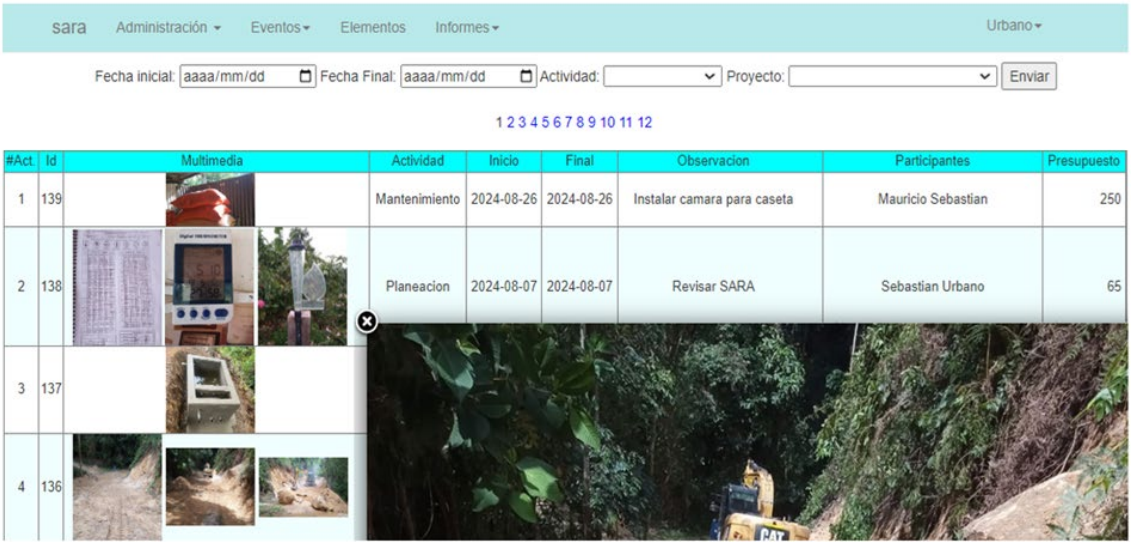
1. Informe general de actividades por proyecto, el cual muestra el listado de registros con la respectiva multimedia que se haya asignado, a cada objeto multimedia se le puede dar clic para ampliarlo (ver figura 3).
2. Informe de registros migrados a la web con el número, proyecto, actividad, fechas del evento, del registro y de sincronización, la cantidad de personas, las personas, las horas dedicadas y una observación
3. Informe de registros consolidados por Mes, App o usuario.
4. Informe de Participación (un calendario los días en que hubo registros).

4.2 API REST

El API REST cumple la función de comunicar el App con el servidor a través de JSON, un intermediario que cuenta con métodos básicos que las App usan para migrar y recibir los datos que fue realizado en Delphi con *DataSnap*[1] usando *Rad Studio* (un entorno de desarrollo integrado (IDE) de la empresa Embarcadero que incluye herramientas y lenguajes para el desarrollo de aplicaciones multiplataforma con servicios flexibles de nube y amplia conectividad para Linux, Windows, Mac y Android), el cual ofrece su propia herramienta para probarla, llamada *RAD Server API Explorer* (Embarcadero Tech, 2022) en la que el desarrollador envía solicitudes HTTP (*get, post, put, delete*) al servidor y ver las respuestas. REST es un acrónimo de *Representational State Transfer* y es un estilo arquitectónico para construir servicios web que proporcione acceso a datos.

1 DataSnap es una tecnología de middleware, con entornos de desarrollo de software de Delphi y C++Builder (Guo & Zhang, 2012), permite el desarrollo de aplicaciones distribuidas que se comunican a través de la red utilizando una arquitectura multicapa para transferir datos y llamar procedimientos remotos tiene componentes TDataSetProvider y TDataSetProviderAdapter. (Guo & Zhang, 2012). Una vez creado el servicio se pueden utilizar varias herramientas para pruebas, como el explorador integrado en el IDE o herramientas externas para revisar las respuestas a solicitudes HTTP como las hace Postman (Embarcadero Tech, 2022)

Figura 3. Ejemplo de uno de los informes web



4.3 App

La App es el software que el usuario usa en su móvil para registrar información de los procesos que lleva a cabo y que se migran para consultas desde la Web, fue desarrollada también con *Rad Studio* y *SQLlite*. La Figura 4 presenta una App en tres de los pasos, es decir:

1. Registrar actividad: escoger proyecto, tipo de actividad y digitar: tiempo, beneficiarios, dinero y observaciones; la App sincroniza los proyectos gestionados desde la web.
2. Generar informe de registros.
3. Generar informes gráficos consolidados por mes.
4. Migrar los registros al Server e Importar registros (requiere login y tiene opción de recordar contraseña; la conexión se habilita si hay internet).

Para mejorar la interfaz, el usuario puede escoger si el registro de datos se da en un único formulario o cambiar el tamaño de la letra ingresando un dato a la vez y tiene componentes para pasar texto a voz y viceversa

4.4 Ubicación en mapa

SARA permite ubicar los proyectos en un mapa, para ello se utilizó el componente *Google Maps*, una herramienta de mapas en línea desarrollada por Google. Permite a los usuarios buscar ubicaciones[2], el cual brindó la capacidad de dibujar geometrías para ubicar elementos o áreas de interés en los que se ejecutan proyectos como construcción de vías, en síntesis, los pasos son:

1. Integrar la API de *Google Maps* en el aplicativo web[3]:
2.

2 ¿Qué es Google Maps? https://www.arimetrics.com/glosario-digital/google-maps
3 Descripción de la API de Maps https://developers.google.com/maps/documentation/embed/get-started?hl=es-419

3. Obtener clave API desde la página de credenciales de Google Cloud
4. Asignar la API utilizando tu clave API.
5. Habilitar el contenedor para el mapa añadiendo el elemento en tu HTML para el mapa, especificando el tamaño deseado.
6. Inicializar el mapa configurando el centro y el nivel de zoom.

Figura 4. Registrar y consultar de actividades de un proyecto (desde la App)

4.5 Configuración de la arquitectura de nube

Tras tener claro el propósito del sistema, se decidió una estructura de microservicios por su escalabilidad, flexibilidad y resiliencia, gestionando despliegue, seguridad y rendimiento, y siguiendo los siguientes pasos adoptados de Wittig & Wittig (2023):

1. Diseño con Microservicios definiendo componentes, responsabilidades e interacciones y en la API especificando los contratos.
2. Configuración del entorno en AWS con ECS para orquestación de contenedores en una instancia EC2 (Elastic Compute Cloud) y la gestión de datos.
3. Generación del contenedor Docker[4] para el despliegue y para la orquestación junto a la configuración de clusters y servicios en ECS.
4. Configuración de escalabilidad y balanceadores de carga (Se usó auto escalado).
5. Pruebas y Validación: Se realizaron pruebas unitarias y de carga con Apache JMeter[5] y Artillery[6] simulando múltiples usuarios accediendo a la aplicación, evaluando tiempos de respuesta, uso de CPU y memoria para asegurar la capacidad y se observó que el sistema soporta el tráfico esperado.
6. Se espera a futuro fortalece la especificación de la seguridad y la gestión de accesos, e incorporar elementos como AWS Lambda para configurar eventos que habiliten servicios como el procesamiento de imágenes o solicitudes web infraestructura subyacente y habilitar el API

4 Proceso de creación de un contenedor en Docker (https://profile.es/blog/que_es_docker/)
5 Herramienta de código abierto utilizada para realizar pruebas de rendimiento y carga en aplicaciones web y otros servicios. Permite simular múltiples usuarios que envían solicitudes simultáneamente a un servidor, evaluando cómo se comporta bajo diferentes cargas (https://jmeter.apache.org)
6 Herramienta de pruebas de carga y rendimiento para aplicaciones web, API y servicios basados en microservicios, altamente escalable y flexible, se enfoca en pruebas de sistemas distribuidos y puede manejar gran cantidad de usuarios concurrentes para simular cargas realistas (https://www.artillery.io)/

Gateway para brindar un punto de entrada para la API REST y gestionar las solicitudes y respuestas entre el Front-End y el microservicio.

5 Conclusiones

El proyecto en el que se enmarca el desarrollo de esta arquitectura de nube tiene por propósito apoyar con TI el sector rural por tanto posteriormente al tener las herramientas disponibles, procederá una estrategia de adopción que se apoya en el hecho de que el alcance de las Apps que, aunque ayudan en la gestión de diferentes eventos de una JAC y/o a sus integrantes, tiene la misma estructura, es de baja escala tecnológica y sigue los mismos pasos de operación, se espera que estos elementos favorezcan su adopción.

El trabajo favorece lineamientos para la educación en ingeniería en aspectos como el paralelo de la metodología con los niveles del TRL (en los que se deben incluir las fases de transferencia) y por la mostrar como la integración de tecnologías favorece el desarrollo social.
Se espera pasar de la disponibilidad de las herramientas a la adopción por parte de los campesinos con los respectivos ajustes que se requieran incorporando didácticas de apropiación, ya que el proyecto SARA continuará en desarrollo y en futuras versiones se espera:

1. Integrar el dispositivo IoT como un sensor de temperatura y un pluviómetro para apoyar el presidente de la JAC con el registro de variables climáticas que actualmente es manual de tal manera que sirva de motivador para el uso de la arquitectura.
2. Trabajar con la comunidad para que, con ayuda de los beneficiarios mejorar aspectos no considerados y se busque la adopción y apropiación de los usuarios, de manera similar a como fue realizado por Gómez, et al (2020) donde se usó gamificación.
3. Incluir elementos como clima, atención en salud o en educación, control de activos, cuidado de la naturaleza, consulta de normativas, entre otros.
4. Mejorar los balanceadores de carga y agregar los lineamientos de seguridad y gestión de accesos y hacer las pruebas de integración que quedaron pendientes.
5. Implementar nuevas apps con microservicios y en diferentes leguajes que permitan ampliar la cobertura y revisar otras alternativas.
6. Integrar al proyecto estrategias didácticas de clase o semillero o trabajos de grado para mostrar que el desarrollo guiado por modelos mejora los procesos (Gómez-Prada, 2022) y mostrar que el software genera desarrollo social.

6 Referencias

- Álvarez, E., & Bernal, C. (2017). Modelo de innovación abierta: Énfasis en el potencial humano, Información tecnológica, 28(1), 65-76, https://doi.org/10.4067/S0718-07642017000100007.
- Berman-Arévalo, E., & Ojeda, D. (2020). Ordinary geographies: Care, violence, and agrarian extractivism in "post-conflict" Colombia. Antipode, 52(6), 1583-1602, https://doi.org/10.1111/anti.12667
- Cíceri, M. (2019). Introducción a Laravel: Aplicaciones robustas y a gran escala. Routledge: Google Books approaches to resettlement: Policy, practice, and theory.

- Condori, J. (2012). Ventajas y desventajas de cloud computing. Inform., tecnología y sociedad, 86, 86-87, http://revistasbolivianas.umsa.bo/pdf/rits/n7/n7a35.pdf.
- Daza, L., Hernández, C., Quijano, A., y Serna, J. (2017). Plan estratégico del Santander. Tesis de Maestría en administración de empresas - Pontificia Universidad Católica del Perú. https://tesis.pucp.edu.pe/repositorio//handle/20.500.12404/8864
- Diga, K., & May, J. (2016). The ICT ecosystem: The application, usefulness, and future of an evolving concept. Information Technology for Development, 22(sup1), 1-6, https://doi.org/10.1080/02681102.2016.1168218
- Dirven, M., & Candia, D. (2020). Medición de lo rural para el diseño e implementación de políticas de desarrollo rural. https://repositorio.cepal.org/handle/11362/45317.
- Embarcadero Tech. (2022). REST Debugger. https://docwiki.embarcadero.com/RADStudio/Sydney/en/REST_Debugger.
- Gómez-Prada, U. (2022). Propuesta de formación guiada con Dinámica de Sistemas para promover el desarrollo de software orientado por modelos en sistemas productivos rurales que requieren asociatividad. Formación Universitaria, 15(1), 153-162, http://dx.doi.org/10.4067/S0718-50062022000100153
- Autor, A. (2025). System for the support of rural community management supported by cloud architecture. Documento Interno.
- Gómez-Prada, U., Orellana, M., y Salinas, J. (2020). Strategy for promoting the Appropriation of Information Systems by Small Cattle Farmers based on Simulation Models and Serious Games. Information, 11(12), 566, https://doi.org/10.3390/info11120566
- Guo, D.; & Zhang, Y. (2012). Research on Software Architecture Base on Data Snap Technology. In Software Engineering and Knowledge Engineering: Theory and Practice: Volume 2 (pp. 459-464). Springer Berlin Heidelberg.
- Hevner, A.; March, S.; Park, J.; Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105, https://doi.org/10.2307/25148625
- Hoyos, A., & Verhelst, K. (2019). La innovación social como herramienta para la transformación social de comunidades rurales. Revista Virtual Universidad Católica del Norte, (57), 87-99, https://doi.org/10.35575/rvucn.n57a7
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). El Proceso Unificado de Desarrollo de Software (RUP). Madrid, España: Pearson.
- Minciencias (2016). Anexo 9 -Niveles de madurez tecnológica. [Online]: https://shre.ink/MnuJ
- Ministerio de TIC de Colombia. (2018). Plan TIC. Agosto 16, 2024. [Online]: https://lc.cx/X_26lc.
- Molina, N., & Tapia, J. (2020). Estudio de factibilidad técnica y financiera, para la migración a Cloud Computing, de los servicios tecnológicos del municipio de Gualaceo. Revista Polo del Conocimiento 5(1), 521-541, https://dialnet.unirioja.es/servlet/articulo?codigo=7436073.
- Natarajan, N., Newsham, A., Rigg, J., & Suhardiman, D. (2022). A sustainable livelihoods framework for the 21st century. World Development, 155, 105898. https://doi.org/10.1016/j.worlddev.2022.105898
- Ndlela, S., & Worth, S. (2021). Creating self-reliance and sustainable livelihoods amongst small-scale sugarcane farmers. The Journal of Agricultural Education and Extension, 27(3), pp. 325–339. https://doi.org/10.1080/1389224X.2020.1851268.
- Pressman, R. (2014). Ingeniería del Software. México: Mc Graw Hill, Séptima Edición.
- Tambotoh, J., Manuputty, A., & Banunaek, F. (2015). Socio-economics factors and information technology adoption in rural area. Procedia Computer Science, 72, 178-185, https://doi.org/10.1016/j.procs.2015.12.119.
- Wittig, A., & Wittig, M. (2023). Amazon Web Services in Action: An in-depth guide to AWS. Simon and Schuster.

Sobre el autor

- **Urbano Eliécer Gómez Prada**, Ingeniero de Sistemas, Magíster en Ingeniería en el área de Informática y Ciencias de la Computación, Doctor en Tecnología Educativa. Profesor Titular de Escuela de Ingeniería de Sistemas. Universidad Industrial de Santander, Bucaramanga, Colombia, Email: uegomezp@uis.edu.co

---

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2025 Asociación Colombiana de Facultades de Ingeniería (ACOFI)

ANEXO 1. Comparación de la metodología de SARA con los niveles del TRL

La metodología se presenta en paralelo con los nueve niveles del Nivel de Preparación Tecnológica (TRL). Esta reflexión y comparación, ilustrada en la Figura 2, debe analizarse junto con la Figura 1 para apreciar la coherencia entre los niveles TRL y la metodología propuesta. En cuanto a estos niveles, SARA puede considerarse a nivel de laboratorio (TRL 5) al completar la Fase 2; actualmente, se encuentra en la Fase 3, cuyo objetivo es alcanzar la demostración en un entorno relevante (TRL 6). Al contemplar el modelo TRL es necesaria la inclusión de fases específicas necesarias para la adopción y apropiación de la tecnología. Estas dos fases requieren iteraciones tras la etapa de demostración (representada en áreas verdes en la figura), ya que es difícil que la tecnología progrese a los niveles 7, 8 o 9 sin adopción y apropiación (áreas grises).

Figura 5. Metodología propuesta para el desarrollo de SARA

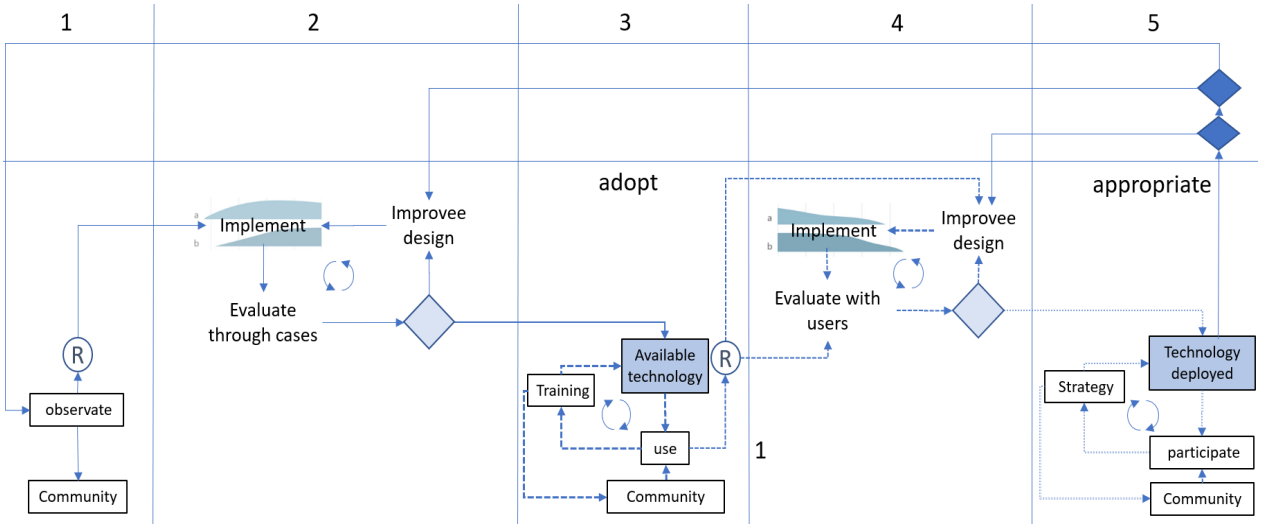


Figure 6. Paralelo de la Metodología con el modelo TRL

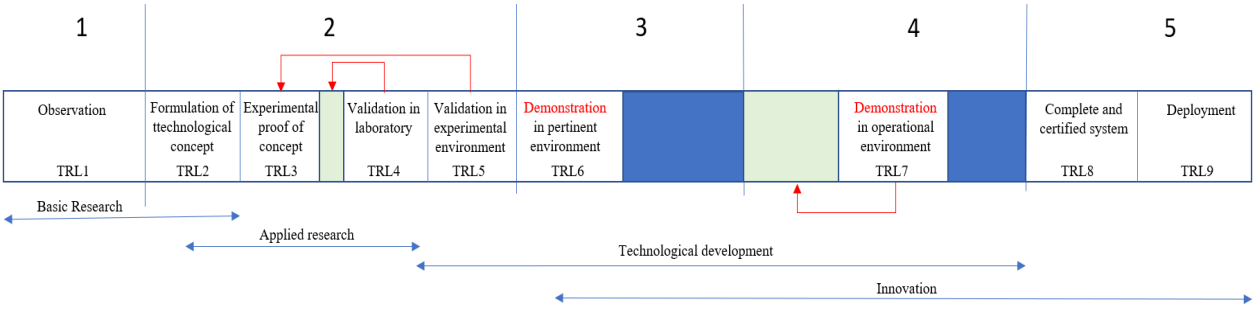
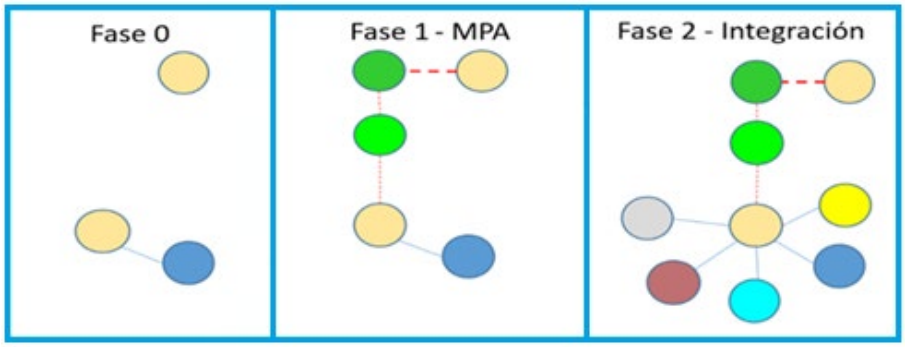



Figura 7. Esquema de cobertura de la arquitectura y de los beneficiarios