



ISSN: 1646-9895

Revista Ibérica de Sistemas e Tecnologias de Informação
Revista Ibérica de Sistemas y Tecnologías de Información

M a r ç o 2 0 • M a r c h 2 0



©AISTI 2020 <http://www.aisti.eu>

Nº 36

Edição / Edición

Nº 36, 03/2020

Tiragem / Tirage: 1000

Preço por número / Precio por número: 17,5€

Subscrição anual / Suscripción anual: 30€ (2 números)

ISSN: 1646-9895

Depósito legal:

Indexação / Indexación

Academic Journals Database, CiteFactor, Dialnet, DOAJ, DOI, EBSCO, GALE, IndexCopernicus, Index of Information Systems Journals, ISI Web of Knowledge, Latindex, ProQuest, QUALIS, SciELO, SCImago, Scopus, SIS, Ulrich's.

Propriedade e Publicação / Propriedad y Publicación

AISTI – Associação Ibérica de Sistemas e Tecnologias de Informação

Rua Quinta do Roseiral 76, 4435-209 Rio Tinto, Portugal

E-mail: aistic@gmail.com

Web: <http://www.risti.xyz>

Director

Álvaro Rocha, Universidade de Coimbra

Coordenadores da Edição / Coordinadores de la Edición

Jezreel Mejia, Centro de Investigación en Matemáticas - Unidad Zacatecas, MX

Mirna Muñoz, Centro de Investigación en Matemáticas - Unidad Zacatecas, MX

Himer Avila-George, Universidad de Guadalajara, MX

Gabriel García-Mireles, Universidad de Sonora, MX

Conselho Editorial / Consejo Editorial

Carlos Ferrás Sexto, Universidad de Santiago de Compostela

Gonçalo Paiva Dias, Universidade de Aveiro

Jose Antonio Calvo-Manzano Villalón, Universidad Politécnica de Madrid

Luís Paulo Reis, Universidade do Porto

Manuel Pérez Cota, Universidad de Vigo

Ramiro Gonçalves, Universidade de Trás-os-Montes e Alto Douro

Conselho Científico / Consejo Científico

A. Augusto Sousa, FEUP, Universidade do Porto, PT

Adolfo Lozano-Tello, Universidad de Extremadura, ES

Adrián Hiebra Pardo, Universidad de Santiago de Compostela, ES

Adriano Pasqualotti, Universidade de Passo Fundo, BR

Alberto Fernández, Universidad Rey Juan Carlos, ES

Alberto Freitas, FMUP, Universidade do Porto, PT

Alcinia Zita Sampaio, IST, Universidade de Lisboa, PT

Alejandro Medina, Instituto Nacional de Astrofísica, Óptica y Electrónica, MX

Alejandro Peña, Escuela de Ingeniería de Antioquia, CO

Alexandre L'Erario, Universidade Tecnológica Federal do Paraná, BR

Alexandre Valle de Carvalho, FEUP, Universidade do Porto, PT

Alicia García-Holgado, Universidad de Salamanca, ES

Alma Gomez-Rodríguez, Universidade de Vigo, ES

Ana Amélia Carvalho, Universidade de Coimbra, PT

Ana Azevedo, Instituto Politécnico do Porto, PT

Ana Isabel Veloso, Universidade de Aveiro, PT

Ana Maria Correia, ISEGI, Universidade Nova de Lisboa, PT
Ana Paula Afonso, Instituto Politécnico do Porto, PT
Anabela Mesquita, Instituto Politécnico do Porto, PT
Anacleto Correia, Escola Naval, PT
Andreia Bos, Instituto Federal do Rio Grande do Sul, BR
Angelica Caro, Universidad del Bío-Bío, CL
Ana Calvão, Universidade de Aveiro, PT
Ana Carla Amaro, Universidade de Aveiro, PT
Ana Melro, Universidade de Aveiro, PT
Ania Cravero, Universidad de La Frontera, CL
Aníbal Zaldivar-Colado, Universidad Autonoma de Sinaloa, MX
Antoni Lluís Mesquida Calafat, Universidad de les Illes Balears, ES
António Abreu, ISCAP, Politécnico do Porto, PT
António Coelho, FEUP, Universidade do Porto, PT
Antonio Fernández-Caballero, Universidad de Castilla-La Mancha, ES
António Godinho, ISLA-Gaia, PT
Antonio Jesus Garcia Loureiro, Universidad de Santiago de Compostela, ES
Antonio Jiménez-Martín, Universidad Politécnica de Madrid, ES
António Manuel Andrade, Universidade Católica Portuguesa, PT
António Palma dos Reis, ISEG, Universidade de Lisboa, PT
António Pedro Costa, Universidade de Aveiro, PT
António Pereira, Instituto Politécnico de Leiria, PT
António Teixeira, Universidade de Aveiro, PT
Armando Mendes, Universidade dos Açores, PT
Arnaldo Martins, Universidade de Aveiro, PT
Arturo J. Méndez, Universidad de Vigo, ES
August Climent Ferrer, La Salle Open University, AD
Baltasar García Pérez-Schofield, Universidad de Vigo, ES
Beatriz Rodríguez, Universidad de la Republica, UY
Beatriz Sainz de Abajo, Universidad de Valladolid, ES
Bernabé Escobar-Pérez, Universidad de Sevilla, ES
Borga Bordel, Universidad Politécnica de Madrid, ES
Boris Almonacid, Global Change Science, CL
Bráulio Alturas, ISCTE - Instituto Universitário de Lisboa, PT

Brenda L. Flores-Rios, Universidad Autónoma de Baja California, MX
Brígida Mónica Faria, ESS, Instituto Politécnico do Porto, PT
Carlos Alexandre Silva, Instituto Federal de Minas Gerais, PT
Carlos Costa, Universidade de Aveiro, PT
Carlos Carreto, Instituto Politécnico da Guarda, PT
Carlos Morais, Instituto Politécnico de Bragança, PT
Carlos Vaz de Carvalho, Instituto Politécnico do Porto, PT
Carmen Galvez, Universidad de Granada, ES
Carlos Rabadão, Politécnico de Leiria, PT
Carlos Rompante Cunha, Politécnico de Bragança, PT
Ciro Martins, Universidade de Aveiro, PT
Claudio de la Riva, Universidade de Oviedo, ES
Cristina M.R. Caridade, ISEC, Politécnico de Coimbra, PT
Daniel Polónia, Universidade de Aveiro, PT
David Fonseca, Universitat Ramon Llull, ES
David Luis La Red Martínez, Universidad Nacional del Nordeste, AR
David Ramos Valcarcel, Universidad de Vigo, ES
Debora Paiva, Universidade Federal de Mato Grosso do Sul, BR
Diana Lancheros-Cuesta, Universidad de La Salle / Universidad Cooperativa de Colombia, CO
Diego Ordóñez-Camacho, Universidad Tecnológica Equinoccial, EC
Dora Simões, Universidade de Aveiro, PT
Edna Dias Canedo, Universidade de Brasília, BR
Eduardo Amadeu Dutra Moresi, Universidade Católica de Brasília, BR
Eduardo Sánchez Vila, Universidad de Santiago de Compostela, ES
Emiliano Reynares, CIDISI - UTN FRSF - CONICET, AR
Enric Mor, Universitat Oberta de Catalunya, ES
Erika Upegui, Universidad Distrital Francisco José de Caldas, CO
Eusébio Ferreira da Costa, Escola Superior de Tecnologias de Fafe, PT
Fábio Marques, Universidade de Aveiro, PT
Feliz Gouveia, Universidade Fernando Pessoa, PT
Fernando Bandeira, Universidade Fernando Pessoa, PT
Fernando Bobillo, Universidad de Zaragoza, ES
Fernando Moreira, Universidade Portucalense, PT

Fernando Paulo Belfo, ISCAC, Politécnico de Coimbra, PT
Fernando Ramos, Universidade de Aveiro, PT
Fernando Ribeiro, Politécnico de Castelo Branco, PT
Filipe Caldeira, Politécnico de Viseu, PT
Filipe Montargil, Politécnico de Lisboa, PT
Filipe Portela, Universidade do Minho, PT
Francisco Javier Lena-Acebo, Universidad de Cantabria, ES
Francisco Restivo, Universidade Católica Portuguesa, PT
Franyelit María Suárez, Universidad UTE, EC
Gabriel Alberto García-Mireles, Universidad de Sonora, MX
Gabriel Guerrero-Contreras, Universidade de Cádiz, ES
Gerardo Gonzalez Filgueira, Universidad da Coruña, ES
Germano Montejano, Universidad Nacional de San Luis, AR
Guilhermina Lobato Miranda, Universidade de Lisboa, PT
Hélder Gomes, Universidade de Aveiro, PT
Hélder Zagalo, Universidade de Aveiro, PT
Helena Garbarino, Universidad ORT Uruguay, UY
Hélia Guerra, Universidade dos Açores, PT
Henrique S. Mamede, Universidade Aberta, PT
Higino Ramos, Universidad de Salamanca, ES
Inês Domingues, CI-IPOP, PT
Isabel Pedrosa, Instituto Politécnico de Coimbra, PT
Isabel Lopes, Politécnico de Bragança, PT
Isaura Ribeiro, Universidade dos Açores, PT
Isidro Calvo, Universidad del País Vasco (UPV/EHU), ES
Ismael Etxeberria-Agiriano, Universidad del País Vasco (UPV/EHU), ES
Ivaldir de Farias Junior, Universidade de Pernambuco, BR
Ivan Garcia, Universidad Tecnológica de la Mixteca, MX
João Paulo Ferreira, ISEC, Politécnico de Coimbra, PT
João Reis, Universidade de Aveiro, PT
João Roberto de Toledo Quadro, CEFET/RJ, BR
Jacinto Estima, Universidade Europeia, PT
Javier Garcia Tobio, CESGA-Centro de Supercomputacion de Galicia, ES
Javier Medina, Universidad Distrital Francisco José de Caldas, CO

Jeimy Cano, Universidad de los Andes, CO
 Jezreel Mejia, Centro de Investigación en Matemática (CIMAT), MX
 João Balsa, FC, Universidade de Lisboa, PT
 João Paulo Costa, Universidade de Coimbra, PT
 João Rocha da Silva, FEUP, Universidade do Porto, PT
 João Tavares, FEUP, Universidade do Porto, PT
 João Vidal de Carvalho, ISCAP, Politécnico do Porto, PT
 Joaquim Ferreira, Universidade de Aveiro, PT
 Joaquim Madeira, Universidade de Aveiro, PT
 Joaquim Pinto, Universidade de Aveiro, PT
 Joaquim Reis, ISCTE - Instituto Universitário de Lisboa, PT
 Jorge Bernardino, ISEC, Politécnico de Coimbra, PT
 Jorge da Silva Correia-Neto, Universidade Federal Rural de Pernambuco, BR
 Jorge Pires, IBM R&D Labs - Prague, CZ
 Jose Alfonso Aguilar, Universidad Autonoma de Sinaloa, MX
 José Alvarez-Garcia, Universidad de Extremadura, ES
 José Borbinha, IST, Universidade de Lisboa, PT
 José Braga de Vasconcelos, Universidade Atlântica, PT
 José Carlos Ribeiro, Politécnico de Leiria, PT
 José Cascalho, Universidade dos Açores, PT
 José Felipe Cocón Juárez, Universidad Autónoma del Carmen, MX
 Jose J. Pazos-Arias, Universidad de Vigo, ES
 José Luis Pardo Díaz, Global School of Business. Empowement and Entrepreneursip, ES
 José Luís Pereira, Universidade do Minho, PT
 José Luís Silva, Instituto Universitário de Lisboa (ISCTE-IUL), PT
 José Paulo Lousado, Instituto Politécnico de Viseu, PT
 José Luis Pestrana Brincones, Universidad de Málaga, ES
 José Luís Reis, ISMAI - Instituto Superior da Maia, PT
 Jose M Molina, Universidad Carlos III de Madrid, ES
 José Machado, Universidade do Minho, PT
 Jose Maria Alvarez Rodríguez, Universidad Carlos III de Madrid, ES
 Jose Maria de Fuentes, Universidad Carlos III de Madrid, ES
 Jose R. R. Viqueira, Universidade de Santiago de Compostela, ES

José Silvestre Silva, Academia Militar, PT
José Torres, Universidade Fernando Pessoa, PT
Josep M. Marco-Simó, Universitat Oberta de Catalunya, ES
Juan Angel Contreras Vas, Universidad de Extremadura, ES
Juan D'Amato, PLADEMA-UNCPBA-CONICET, AR
Juan M. Santos Gago, Universidad de Vigo, ES
Juan Manuel Fernández-Luna, Universidad de Granada, ES
Jugurta Lisboa-Filho, Universidade Federal de Viçosa, BR
Leila Weitzel, Universidade Federal Fluminense, BR
Leonardo Bermon, Universidad Nacional de Colombia, CO
Leticia Morales Trujillo, Universidad de Sevilla, ES
Lucila Ishitani, PUC Minas, BR
Lucila Romero, Universidad Nacional del Litoral, AR
Luis Alvarez Sabucedo, Universidad de Vigo, ES
Luís Bruno, Instituto Politécnico de Beja, PT
Luis Camarinha-Matos, Universidade Nova de Lisboa, PT
Luís Cavique, Universidade Aberta, PT
Luis Chamba Eras, Universidad Nacional de Loja, EC
Luís Ferreira, Instituto Politécnico do Cávado e do Ave, PT
Luis Enrique Sánchez Crespo, Universidad de Castilla-La Mancha, ES
Luis Vilán-Crespo, Universidad de Vigo, ES
Luisa María Romero-Moreno, Universidad de Sevilla, ES
Luisa Miranda, Instituto Politécnico de Bragança, PT
Lus Sussy Bayona Ore, Universidad Nacional Mayor de San Marcos, PE
Magdalena Arcilla Cobián, Universidade Nacional de Educación a Distancia, ES
Manuel Fernández-Veiga, Universidad de Vigo, ES
Manuel Jose Fernandez Iglesias, Universidad de Vigo, ES
Marcelo Marciszack, Universidad Tecnológica Nacional, AR
Marcelo de Paiva Guimarães, Universidade Federal de São Paulo, BR
Marco Painho, ISEGI, Universidade Nova de Lisboa, PT
Mari de los Milagros Gutierrez, Universidad Tecnológica Nacional, AR
Maria Amelia Eliseu, Mackenzie Presbyterian University, BR
Maria Cecília Rosa, Politécnico da Guarda, PT
Maria Cristina Marcelino Bento, UNIFATEA, BR

Maria do Rosário Bernardo, Universidade Aberta, BR
Maria Helena Garcia Ruiz, Universidad de Cantabria, ES
María J. Lado, Universidad de Vigo, ES
Maria João Ferreira, Universidade Portucalense, PT
Maria João Gomes, Universidade do Minho, PT
Maria José Angélico, Instituto Politécnico do Porto, PT
Maria José Sousa, Universidade Europeia, PT
Marisol B. Correia, Universidade do Algarve, PT
Maristela Holanda, Universidade de Brasília, BR
Martín Llamas Nistal, Universidad de Vigo, ES
Martín López Nores, Universidad de Vigo, ES
Matías García Rivera, Universidad de Vigo, ES
Mercedes Ruiz, Universidad de Cádiz, ES
Miguel A. Brito, Universidade do Minho, PT
Miguel Bugalho, Universidade Europeia, PT
Miguel Casquilho, IST, Universidade de Lisboa, PT
Mirna Ariadna Muñoz Mata, Centro de Investigación en Matemáticas (CIMAT), MX
Nelson Rocha, Universidade de Aveiro, PT
Nuno Lau, Universidade de Aveiro, PT
Nuno Melão, Politécnico de Viseu, PT
Nuno Ribeiro, Universidade Fernando Pessoa, PT
Oscar Mealha, Universidade de Aveiro, PT
Patrícia Cabral de Azevedo Tedesco, Universidade Federal de Pernambuco, BR
Patricia Dias, Universidade do estado de Minas Gerais, BR
Patrícia Oliveira, Universidade de Aveiro, PT
Paula Prata, Universidade da Beira Interior, PT
Paulo Martins, Universidade de Trás-os-Montes e Alto Douro, PT
Paulo Pinto, FCT, Universidade Nova de Lisboa, PT
Paulo Rurato, Universidade Fernando Pessoa, PT
Paulo Urbano, FC, Universidade de Lisboa, PT
Pedro Araújo, Universidade da Beira Interior, PT
Pedro Fernandes, Instituto Superior Miguel Torga, PT
Pedro Sánchez Palma, Universidad Politécnica de Cartagena, ES
Pedro Sanz Angulo, Universidad de Valladolid, ES

Pedro Sobral, Universidade Fernando Pessoa, PT
Pedro Sousa, Universidade do Minho, PT
Pilar Mareca Lopez, Universidad Politécnica de Madrid, ES
Rui Rodrigues, Instituto Superior Miguel Torga, PT
Ramiro Delgado, Universidad de las Fuerzas Armadas, EC
Ramon Alcarria, Universidad Politécnica de Madrid, ES
Raul Laureano, ISCTE - Instituto Universitário de Lisboa, PT
Ricardo J. Rodríguez, Universidad de Zaragoza, ES
Ricardo Linden, FSMA, BR
Rita Oliveira, Universidade de Aveiro, PT
Rita Santos, Universidade de Aveiro, PT
Roberto Theron, Universidad de Salamanca, ES
Rodolfo Miranda Barros, Universidade Estadual de Londrina, BR
Rodrigo Rocha Silva, Universidade de Coimbra, PT
Rogerio Garcia, Universidade Estadual Paulista, BR
Román Lara, Universidad de las Fuerzas Armadas, EC
Rubén González Crespo, Universidad Internacional de La Rioja, ES
Rui Cruz, IST, Universidade de Lisboa, PT
Rui José, Universidade do Minho, PT
Rui Pedro Marques, Universidade de Aveiro, PT
Rui Pitarma, Politécnico da Guarda, PT
Rui S. Moreira, Universidade Fernando Pessoa, PT
Rute Abreu, Politécnico da Guarda, PT
Samuel Sepúlveda, Universidad de La Frontera, CL
Samuel Silva, Universidade de Aveiro, PT
Santiago Gonzales Sánchez, Universidad Inca Garcilaso de la Vega, PE
Sara Balderas-Díaz, Universidad de Cádiz, ES
Sara Paiva, Politécnico de Viana do Castelo, PT
Saulo Barbara de Oliveira, Universidade Federal Rural do Rio de Janeiro, BR
Sérgi F. Lopes, Universidade do Minho, PT
Sergio Gálvez Rojas, Universidad de Málaga, ES
Sérgio Guerreiro, Universidade Lusófona de Humanidades e Tecnologias, PT
Sérgio Nunes, ISEG, Universidade de Lisboa, PT
Silvia Fernandes, Universidade do Algarve, PT

Solange N Alves de Souza, Universidade de São Paulo, BR
Telmo Silva, Universidade de Aveiro, PT
Teresa Guarda, Universidad Estatal Península de Santa Elena, EC
Thiago Dias, CEFET-MG, BR
Valéria Farinazzo Martins, Universidade Presbiteriana Mackenzie, BR
Vania Baldi, Universidade de Aveiro, PT
Verónica Vasconcelos, ISEC, Politécnico de Coimbra, PT
Victor Flores, Universidad Católica del Norte, CL
Victor Hugo Medina Garcia, Universidad Distrital Francisco José de Caldas, CO
Vitor Carvalho, Instituto Politécnico do Cávado e do Ave, PT
Vitor Santos, ISEGI, Universidade Nova de Lisboa, PT
Wagner Tanaka Botelho, Universidade Federal do ABC, BR
Xose A. Vila, Universidad de Vigo, ES

Nuevos desafíos en Tecnologías de Información y Comunicación

New challenges in Information and Communication Technologies

Jezreel Mejia¹, Mirna Muñoz¹, Himer Avila-George², Gabriel García-Mireles³

{jmejia, mirna.munoz}@ciimat.mx, himer.avila@academicos.udg.mx,
gabriel.garcia@unison.mx

¹Centro de Investigación en Matemáticas- Unidad zacatecas, Parque Quantum, Ciudad del Conocimiento, Avenida Lasec, andador Galileo Galilei, manzana 3 lote 7, C. P. 98160, Zacatecas, México.

²Departamento de Ciencias Computacionales e Ingenierías, Universidad de Guadalajara. Carretera Guadalajara - Ameca Km. 45.5, C.P. 46600, Ameca, Jalisco, México.

³Universidad de Sonora, Blvd. Luis Encinas y Rosales s/n, Col. Centro, C.P.83000, Hermosillo, Sonora, México.

DOI: 10.17013/risti.36.0

Introducción

Los artículos presentados en este número fueron seleccionados de los artículos aceptados en el Octavo Congreso Internacional de Mejora de Procesos Software (CIMPS) 2019, realizado en la Ciudad de León, Guanajuato, México del 23 al 25 de Octubre del 2019. Este congreso ha tenido una tasa de aceptación del 30% de los artículos enviados.

Los artículos publicados en este número de la Revista Ibérica de Sistemas y Tecnologías de la Información (RISTI), aborda temas relacionados a “Nuevos Desafíos en Tecnologías de Información y Comunicación”. Los artículos presentan versiones extendidas y mejoradas de las versiones originales que han sido presentados en el congreso CIMPS 2019.

Estructura

El primer artículo presenta el estado actual de las empresas muy pequeñas con respecto a la implementación del Perfil básico de ISO/IEC 29110, las herramientas de software utilizadas para respaldarlo y las debilidades que surgen dentro de éstas con la finalidad de ayudar en evitar las barreras para la implementación de esta norma.

En el segundo artículo, los autores presentan una propuesta de herramientas que den soporte en la implementación a las actividades y tareas de los dos procesos que contiene la norma ISO/IEC 29110, ya que existe una carencia de herramientas que permitan automatizar o dar soporte en la implementación de estas nuevas normas.

Con respecto al tercer artículo, se presenta un estudio exploratorio donde se analiza la influencia de la Teoría de Belbin en la calidad del producto generado con el desarrollo de requisitos software. El estudio se desarrolla mediante un experimento controlado donde el factor manipulado por el investigador se corresponde con la forma en la que los equipos de desarrollo son integrados con estudiantes.

El cuarto artículo presenta una plataforma para analizar datos de pruebas de desempeño utilizando tecnología de Big Data para recolectar, almacenar, procesar y analizar resultados de desempeño de una manera integrada con la finalidad de analizar pruebas de regresión para verificar y validar la calidad de las nuevas versiones creadas durante el proceso de desarrollo.

El quinto artículo propone un algoritmo para manipular un brazo antropomórfico utilizando una pantalla táctil LCD y una transformación lineal para obtener la trayectoria del brazo robótico, cinemática inversa y directa. Con la finalidad de que las personas con discapacidades físicas que tienen algún problema para mover sus cuerpos o simplemente no tienen la fuerza para moverlo puedan controlar el brazo robótico en las actividades cotidianas de una forma natural y fácil.

En el sexto artículo, se presenta una revisión sistemática de la literatura referente al uso de los vehículos aéreos no tripulados en aplicaciones relacionadas con la detección de vegetación y el inventario de plantas. Asimismo, se realiza un análisis de las principales técnicas computacionales y estadísticas utilizadas para el procesamiento de las imágenes tomadas desde vehículos aéreos no tripulados.

Los autores del séptimo artículo proponen una mejora en la interacción entre el sistema domótico inteligente y el usuario final utilizando un asistente de voz inteligente en lugar de aplicaciones móviles o una plataforma web local. Además, se exploran otras fuentes de información para pronósticos meteorológicos, como las interfaces de programación de aplicaciones (API) que proporcionan este servicio.

Finalmente, el octavo artículo presenta los resultados obtenidos de la implementación de tecnologías de internet de las cosas aplicadas a la mejora de la operación de un sistema de bombeo hidráulico en una comunidad rural. Como resultado, los autores presentan el desarrollo de una plataforma integral que incluye una aplicación web, una aplicación móvil y un módulo electrónico. Los resultados obtenidos muestran que la plataforma desarrollada así como los sensores utilizados son factibles y viables para controlar la bomba del suministro de agua de la comunidad de Huixtitla, Veracruz.

Agradecimientos

Agradecemos al Director de la Revista Álvaro Rocha por otorgar la oportunidad para dar a conocer los trabajos más relevantes presentados en el congreso CIMPS 2019. Asimismo, expresamos nuestra gratitud a los autores de los trabajos publicados, lo cual hizo posible este número en RISTI. Damos la enhorabuena a todos aquellos autores cuyos artículos han sido publicados en este número de RISTI.

Índice / Index

EDITORIAL

Nuevos desafíos en Tecnologías de Información y Comunicación.....	ix
<i>Jezreel Mejía, Mirna Muñoz, Himer Avila-George, Gabriel García-Mireles</i>	

ARTIGOS / ARTICULOS / ARTICLES

Situación actual sobre la implementación del perfil básico ISO/IEC 29110 en EMP: una revisión sistemática de la literatura	1
<i>Mirna Muñoz, Manuel Peralta</i>	
Identificación de herramientas como soporte en la implementación de la Norma ISO/IEC 29110 (perfil básico)	15
<i>Jezreel Mejía, Edgar Bonilla, Elizabeth Villanueva, Einar Serna, Israel Faustino, Alessandro Milán</i>	
Explorando la influencia de los roles de Belbin en la especificación de requisitos de software	34
<i>Raúl Aguilar Vera, Mirna Muñoz Mata, Julio Díaz Mendoza, Juan Ucán Pech</i>	
Plataforma de análisis de datos para la evaluación de desempeño de software.....	50
<i>Silvana de Gyves Avila, Patricia Ortegon Cano, Ayrton Mondragon Mejía, Ismael Solis Moreno, Arianne Navarro Lepe, Gloria Eva Zagal Domínguez</i>	
Algoritmo para controlar un brazo antropomórfico usando una transformación lineal.....	65
<i>Yadira Quiñonez, Oscar Zatarain, Carmen Lizárraga, Juan Peraza, Juan Peraza, Jezreel Mejía</i>	
Una revisión sistemática de la literatura enfocada al uso de vehículos aéreos no tripulados durante el proceso de detección de vegetación	82
<i>Enrique Ponce-Corona, María Guadalupe Sánchez, Daniel Fajardo-Delgado, Brenda Acevedo-Juárez, Miguel De-la-Torre, Himer Avila-George, Wilson Castro</i>	
Uso del asistente virtual Alexa como herramienta de interacción para el monitoreo de clima en hogares inteligentes por medio de Raspberry Pi y DarkSky API	102
<i>Adrián Calvopiña, Freddy Tapia, Luis Tello-Oquendo</i>	
Ameyali: Evaluación de la usabilidad de la plataforma Integral para el control y monitoreo del servicio de agua potable en comunidades rurales bajo la escala de Sus	116
<i>A. Montiel-De Jesús, C. Morales-Constantino, S. Ixmattlahua-Díaz, N. Hernández-Chaparro, H. Marín-Vega</i>	

Situación actual sobre la implementación del perfil básico ISO/IEC 29110 en EMP: una revisión sistemática de la literatura

Mirna Muñoz, Manuel Peralta

mirna.munoz@cimat.mx, manuel.peralta@cimat.mx

Centro de Investigación en Matemáticas, (CIMAT, A.C.), Unidad Zacatecas, México, Calle Lasec, Andador Galileo Galilei. Andador 3, Lote 7, CP 98160, Zacatecas, México.

DOI: 10.17013/risti.36.1-14

Resumen: Hoy en día, las empresas de desarrollo de software están en un contexto lleno de competitividad, así que, deben producir sistemas de software que cumplan con estándares de calidad que fomenten las buenas prácticas de desarrollo. Sin embargo, gran cantidad de estas empresas son Entidades Muy Pequeñas (EMP) con menos de 25 miembros, esto dificulta la implementación de grandes estándares como CMMI, ISO 12207, entre otros. Por esta razón, se creó el estándar ISO/IEC 29110, diseñado para este tipo de entidades que no desarrollan software crítico. En cambio, a pesar de que los procesos propuestos por ISO/IEC 29110 son ligeros, la mayoría de las EMP enfrentan barreras para su implementación. Este artículo presenta el estado actual de las EMP con respecto a la implementación del Perfil básico de ISO/IEC 29110, las herramientas de software utilizadas para respaldarlo y las debilidades que surgen dentro de estas EMP.

Palabras-clave: ISO/IEC 29110, Entidades Muy Pequeñas, implementación, barreras, herramientas de soporte.

Current State about ISO/IEC 29110 basic profile implementation in VSEs: A Systematic Literature Review

Abstract: Nowadays, software development enterprises are within a full of competitiveness context, therefore, these companies must produce software systems in compliance with quality standards that encourage good practices for development. Nevertheless, most of the software development enterprises are Very Small Entities (VSEs) with less than 25 members, this fact makes difficult to implement large standards such as CMMI, ISO 12207. For this reason, the ISO/IEC 29110 standard was created, which is specially designed for this type of entities that do not develop critical software. However, even although the processes proposed by ISO/IEC 29110 are very lightweight, most VSEs are facing barriers for their implementation. This paper aims to present the current state of VSEs regarding the implementation of the Basic Profile of ISO/IEC 29110 standard, the software tools used for supporting it, and the weaknesses that arise within these VSEs.

Keywords: ISO/IEC 29110, Very Small Entities, implementation, barriers, support tools.

1. Introducción

En los últimos años, el incremento de la importancia del software como núcleo de muchas organizaciones, brinda la oportunidad para que las Entidades Muy Pequeñas (EMP) generen productos y servicios de software capaces de proporcionar soluciones de alto rendimiento a problemas de diferentes dominios (industrial, agrícola, aeronáutico, Tecnologías de la Información y la Comunicación, entre otros). Por lo tanto, es importante garantizar la calidad de sus productos de software (Clarke & Yilmaz, 2017).

La introducción de mejoras en las EMP ayuda a implementar un conjunto de las mejores prácticas de ingeniería de software con el fin de lograr beneficios como el aumento de la calidad de su producto, reducir los tiempos de entrega, reducir los costos de producción y ampliar su cartera de clientes, entre otros (Nonoyama et al., 2016).

En este contexto, por una parte, modelos o normas como CMMI (SEI, 2010), MoProSoft (Oktaba & Martínez, 2005) e ISO/IEC 15504 (International Standard Organization, 2004), se han desarrollado como referencia para apoyar a las organizaciones en la implementación de Mejoras en los Procesos de Software (SPI por sus siglas en inglés). Sin embargo, estos modelos se centran en proporcionar un conjunto de prácticas recomendadas que cubran “qué prácticas hacer” sin cubrir “cómo implementarlas”. Además, estos modelos y estándares fueron desarrollados para las empresas más grandes sin tener a las EMP en mente (Maciaszek & Loucopoulos, 2011).

Con respecto a esta situación, se ha desarrollado el estándar ISO/IEC 29110, que presenta un conjunto de normas y directrices que se basan en subconjuntos de elementos estándares apropiados llamados perfiles para entidades muy pequeñas (EMP). La industria del software reconoce el valor de las EMP en la contribución de productos y servicios valiosos, de acuerdo con (Muñoz et al., 2018) este tipo de organizaciones representan un promedio de 90% en México.

Sin embargo, aunque ISO/IEC 29110 es un estándar diseñado para las EMP y adaptable a cualquier ciclo de desarrollo de software (Laporte & O'Connor, 2016). Las EMP encuentran a algunas debilidades durante la implementación del estándar.

El objetivo de este artículo, es mostrar los resultados de una Revisión Sistemática de la Literatura (RSL) realizada con el fin de identificar el estado actual de las EMP con respecto a la implementación del Perfil Básico del estándar ISO/IEC 29110, el uso de herramientas de software para apoyarlo y las principales debilidades que aparecen dentro de estas EMP. Después de la introducción, este documento se organiza de la siguiente manera: la sección 2 presenta la RSL llevada a cabo en esta investigación, así como los resultados obtenidos después de analizar los estudios primarios identificados, y la sección 3 incluye conclusiones y trabajo futuro.

2. Revisión Sistemática de la Literatura

Con el fin de establecer el estado del arte con respecto a estas herramientas de software y las debilidades que surgen dentro de las EMP durante la implementación del Perfil Básico del estándar ISO/IEC 29110, se llevó a cabo una Revisión Sistemática de la Literatura.

La Revisión Sistemática de la Literatura (RSL), es un método utilizado para identificar, evaluar e interpretar todas las evidencias disponibles, pertinentes sobre un tema o una pregunta de investigación (Kitchenham & Charters, 2007), este protocolo se ha utilizado en otros temas de investigación como en (Hernandez et al., 2017; Mejia et al., 2016; Mejía et al., 2017; Mejia Miranda et al., n.d.; Munoz et al., 2018). RSL reduce la posibilidad de sesgo en la búsqueda de estudios, ya que es necesario definir un protocolo que especifique los métodos utilizados para guiar la Revisión Sistemática de la Literatura (Selleri Silva et al., 2015).

La RSL consta de tres fases principales: 1) Planificación de la revisión, 2) Realización de la revisión y 3) Reporte de los resultados de la revisión. Estas fases pueden parecer secuenciales, pero es importante reconocer que estas fases pueden implicar iteración. La implementación del protocolo definido para la presente investigación se describe brevemente en las secciones siguientes.

2.1. Planificación de la Revisión Sistemática

Ésta es la etapa inicial de la Revisión Sistemática y debe ejecutarse de la siguiente manera: confirmar la necesidad de la revisión, especificar las preguntas de investigación, listar las fuentes de datos para realizar las búsquedas y formular la cadena de búsqueda adaptándola a los motores de búsqueda seleccionados (Kitchenham & Charters, 2007).

Confirmar la necesidad para la revisión

Para las organizaciones en el campo del desarrollo de software, es esencial implementar procesos de desarrollo basados en las mejores prácticas de modelos y estándares de calidad, sin embargo, la mayoría de las EMP en este campo no los implementan, esta situación lleva a la mayoría de sus proyectos a fallar y producir pérdidas financieras para estas organizaciones (O'Connor & Laporte, 2012). Con la finalidad de obtener una implementación exitosa del estándar ISO/IEC 29110, las EMP necesitan un entendimiento común sobre cómo implementar los procesos adaptados a su cultura particular, y también una herramienta de apoyo específica para implementar buenas prácticas y generar evidencia a partir de estas prácticas (Lohier & Rodríguez-Dapena, 2016), a su vez, necesitan conocer las principales debilidades que una EMP suele tener al implementar este estándar.

Es por eso por lo que esta investigación está orientada al estudio del estado actual y las principales debilidades de las EMP de desarrollo de software. Además, es importante conocer las herramientas de software que respaldan la implementación de ISO/IEC 29110, para que sea de ayuda al identificar las características que una herramienta de soporte debería cubrir para satisfacer sus necesidades.

Especificar las preguntas de investigación.

Una vez identificada la necesidad de la revisión, las preguntas de investigación se especificaron utilizando la técnica PICOC. Las preguntas de investigación se establecieron de la siguiente manera:

RQ1.- ¿Cuál es el estado actual de las EMP de desarrollo de software con respecto al uso del perfil básico del estándar ISO/IEC 29110?

RQ2.- ¿Cuáles son las principales debilidades de las EMP al implementar el perfil básico del estándar ISO/IEC 29110?

RQ3.- ¿Qué herramientas de soporte existen para la implementación del perfil básico del estándar ISO/IEC 29110?

El objetivo de estas preguntas es obtener información sobre el estado de las EMP e identificar las necesidades de las EMP con respecto a la implementación del Perfil Básico de ISO/IEC 29110.

Crear la cadena de búsqueda

Para responder a las preguntas de investigación, se seleccionaron las palabras clave identificadas en las mismas, luego, se enumeraron sinónimos y términos relacionados con las palabras clave. Después de eso, combinando esas palabras clave y los conectores lógicos tales como “AND” y “OR” (“Y” y “O” respectivamente), se creó la cadena de búsqueda. Las palabras clave identificadas y la cadena resultante se muestran en la Tabla 1.

Seleccionar las fuentes de datos

La búsqueda de estudios primarios se realizó en bases de datos indexadas consideradas relevantes para la Ingeniería de Software, específicamente para el área de Mejora de Procesos de Software, debido al alto nivel de su uso en el campo de la investigación de ingeniería de software, estas diferentes bibliotecas digitales fueron: a) SpringerLink, b) IEEE Xplore, c) Scopus, d) Web of Science, e) ScienceDirect y f) Biblioteca digital ACM.

Es importante mencionar, que estas palabras clave, sus sinónimos o términos relacionados, y la cadena de búsqueda fueron creados en el idioma inglés, para ampliar el alcance de la investigación.

Palabras clave	Sinónimos	Cadena de búsqueda
Implementación	Usage	(Implementation OR Usage OR Using OR Adoption) AND (ISO/IEC 29110 OR ISO 29110) AND (VSE OR VSEs OR Very Small Entities OR Very Small Entity OR Very Small Enterprise)
	Using	
	Adoption	
ISO/IEC 29110	ISO 29110	AND (Tool OR Application OR Software OR Support tool)
VSE	VSEs	
	Very Small Entity	
	Very Small Entities	
	Very Small Enterprise	
Tool	Application	
	Software	
	Support tool	

Tabla 1 – Palabras clave y Cadena de Búsqueda Resultante.

2.2. Ejecución la Revisión

La segunda fase de la RSL es la ejecución de la revisión, que se centra en la obtención de los estudios primarios. (Kitchenham & Charters, 2007). A continuación, se muestra el proceso de selección de estudios primarios y el número de estudios encontrados en cada paso del proceso. Esta fase incluye las siguientes actividades: establecer los criterios de inclusión y exclusión, establecer los criterios de calidad de los estudios, seleccionar los estudios primarios y, finalmente, extraer datos de los estudios primarios.

Establecer los criterios de inclusión y exclusión

Para comenzar con el proceso de selección de los estudios primarios, se establecieron los siguientes criterios de inclusión y exclusión:

Criterios de Inclusión:

1. Estudios escritos en inglés o español.
2. Estudios dentro del periodo 2015-2018.
3. Estudios que describen la situación de las EMP que implementan el estándar ISO/IEC 29110.
4. Estudios que hacen referencia al uso de herramientas de soporte para los procesos de las EMP o las debilidades de las EMP.

Criterios de Exclusión:

1. Estudios que aparecen en más de una fuente.
2. Estudios que no contienen información relevante sobre la implementación del estándar ISO/IEC 29110 en las EMP.

Establecer los criterios de calidad en los estudios.

Para garantizar que la información contenida en cada estudio primario sea relevante y valiosa para la presente investigación, se establecieron dos criterios de calidad en forma de las siguientes preguntas:

SQA1.- ¿El estudio describe las debilidades en al menos un estudio de caso sobre la implementación de ISO/IEC 29110 en una EMP?

SQA2.- ¿El estudio hace referencia al uso de al menos una herramienta de soporte para implementar el estándar ISO/IEC 29110?

La evaluación de los estudios basada en los criterios de calidad se llevó a cabo seleccionando aquellos estudios que cumplían al menos una de las dos preguntas.

Selección de estudios primarios

Esta actividad sugiere definir un proceso de selección de estudios para obtener los estudios primarios para esta investigación. El proceso para seleccionar estudios primarios consistió en seis pasos:

1. Usar y adaptar la cadena de búsqueda a los motores de búsqueda seleccionados.
2. Filtrar los estudios aplicando los dos primeros criterios de inclusión.

3. Leer los títulos y resúmenes de los estudios para identificar posibles estudios primarios.
4. Aplicar los criterios restantes de inclusión y exclusión al leer la introducción y la conclusión.
5. Aplicar criterios de calidad leyendo todo el estudio.
6. Seleccionar estudios primarios.

La Figura 1 muestra la implementación del proceso de selección de estudios descrito anteriormente. Como muestra la figura, se encontraron un total de 299 estudios durante la ejecución de la investigación en las fuentes de datos seleccionadas, pero solo 8 estudios cumplieron con los criterios de calidad, criterios de inclusión y exclusión y se consideraron como estudios primarios.

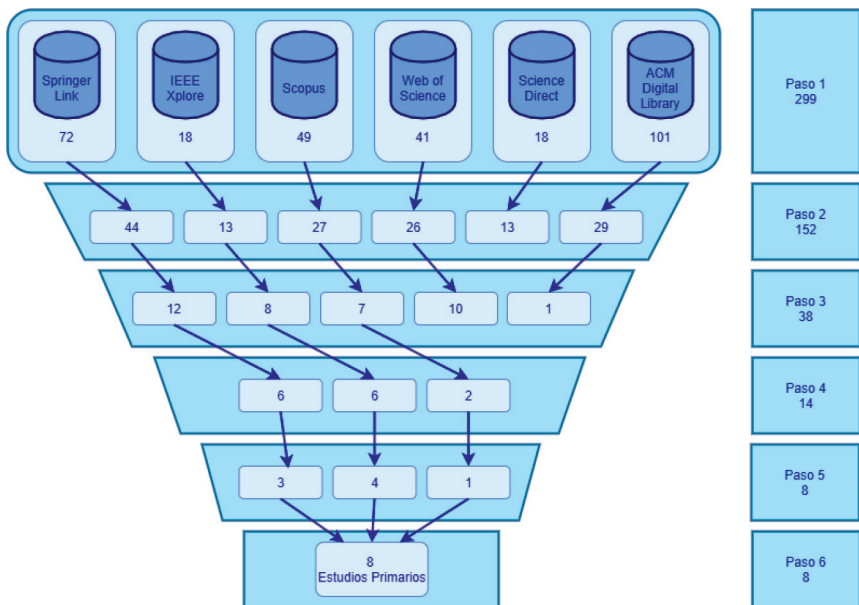


Figura 1 – Proceso de selección de estudios y resultados finales.

Extracción de información

Antes de realizar la extracción de datos, los estudios primarios se organizaron en la herramienta de gestión de archivos PDF Mendeley®, estos estudios se leyeron y analizaron. Posteriormente, la información obtenida de los estudios primarios se registró en una plantilla desarrollada en la herramienta Hojas de Cálculo de Google®, se diseñó un formato para la extracción de datos con las siguientes columnas: identificador, título, autores, año, palabras clave, fuente de datos, objetivo, problema, número de estudios de caso, principales debilidades, resultados obtenidos de la certificación y herramientas de soporte.

2.3. Resultados de la Revisión Sistemática

En esta sección, se muestran los principales resultados obtenidos de la RSL, estos resultados proporcionan el escenario que enfrentan las EMP con respecto a la implementación del estándar ISO/IEC 29110. La Tabla 2, muestra la cobertura de cada estudio primario relacionado con preguntas de investigación y los estudios de caso contenidos en el estudio primario. Los resultados se organizan en las siguientes subsecciones y los estudios primarios se enumeran en el Apéndice A.

Estudio Primario			Preguntas de investigación		
ID	Estudios de caso	Estudios de caso repetidos	RQ1	RQ2	RQ3
PSo1	3		X	X	X
PSo2	74			X	
PSo3	4		X	X	
PSo4	15			X	
PSo5	9	3	X	X	X
PSo6	11	4	X		X
PSo7	5			X	X
PSo8	4		X	X	
Total	125	7	31	114	23
Total, sin los estudios de caso repetidos	118		24	111	20

Tabla 2 – Trazabilidad entre preguntas de investigación y estudios primarios.

Es importante resaltar que durante el análisis de datos se detectaron un total de 7 estudios de casos repetidos, para obtener el conjunto de datos más claro, estos estudios de caso se eliminaron del análisis.

RQ1. ¿Cuál es el estado actual de las EMP de desarrollo de software con respecto al uso del estándar ISO/IEC 29110?

Las EMP que están implementando el estándar ISO/IEC 29110 reconocen que tuvo un factor de impacto positivo para su productividad. La mayoría de ellos está de acuerdo con los beneficios que la implementación de este estándar brinda a sus organizaciones. Los beneficios principales y más frecuentes se muestran en la Figura 2.

Como se muestra en la Figura 2, los beneficios más comunes están relacionados con la mejora de los procesos internos de las EMP, la reducción de las actividades de retrabajo, el incremento de la satisfacción del cliente, el acceso a nuevos clientes y el aumento de la calidad de sus productos.

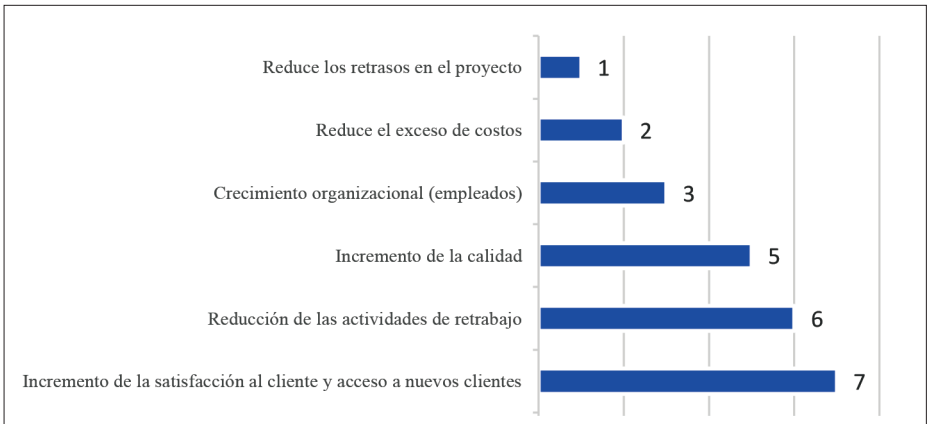


Figura 2 – Beneficios de implementar el estándar ISO/IEC 29110 (Frecuencia).

Los estudios de caso contenidos en estudios primarios como PS01 y PS05 informaron solo el 12% en actividades de retrabajo en contraste con el 30% en proyectos similares (Laporte & O'Connor, 2016). Además, el nivel de satisfacción sobre la implementación de ISO/IEC 29110 aumentó un promedio del 20% como se muestra en PS01.

Por otro lado, los beneficios menos comunes identificados están relacionados con la reducción de sobrecostos, el crecimiento organizacional y la reducción de demoras en los proyectos. Los estudios de caso contenidos en PS06 reportaron un crecimiento organizacional promedio de 18 empleados. Es importante resaltar que para responder la RQ1, solo 24 estudios de caso descritos en 5 estudios primarios contenían información relevante (PS01, PS03, PS05, PS06, PS08).

La reducción de sobrecostos y demoras en los proyectos son dos temas relacionados con la gestión del proyecto, que pueden tratarse con la ejecución de actividades correctas de monitoreo del proyecto, por lo que estos dos beneficios podrían lograrse mediante el uso de una herramienta de apoyo.

RQ2. ¿Cuáles son las principales debilidades de las EMP cuando se implementa el estándar ISO/IEC 29110?

Dada la RQ2, siete de los ocho estudios primarios contenían información relevante sobre las debilidades de las EMP con respecto a la implementación de ISO/IEC 29110 (PS01, PS02, PS03, PS04, PS05, PS07, PS08), que contenían 111 estudios de caso. La Figura 3 muestra las principales debilidades identificadas en estos estudios de caso.

La Figura 3, muestra que las debilidades más relevantes están relacionadas con: Diseño arquitectural y detallado, integración de componentes de software, aseguramiento de la calidad, verificación y validación, y casos y procedimientos de prueba. Por el contrario, las debilidades menos destacadas están relacionadas con: Proceso de gestión de proyectos, mala interpretación del estándar, falta de enfoque de mejora de procesos de software, falta de cultura de procesos organizacional y falta de capacitación adecuada de procesos.

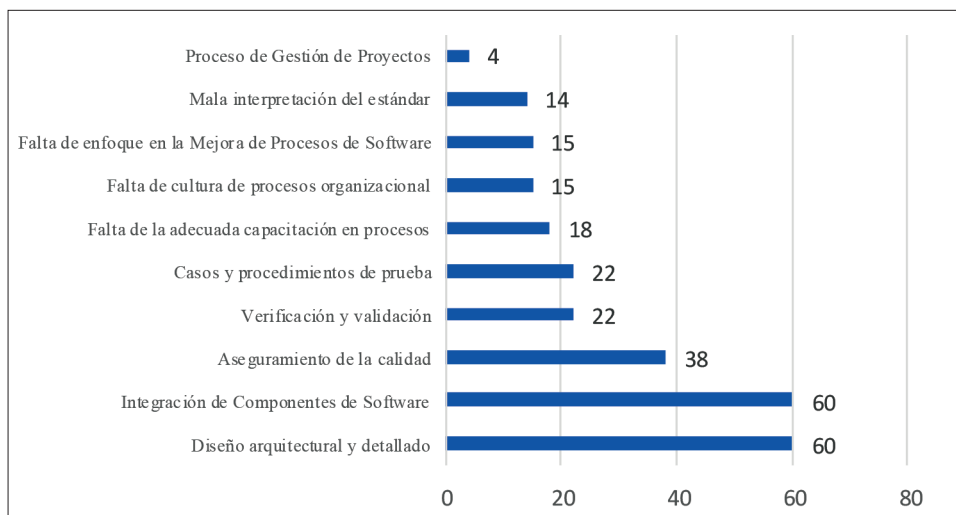


Figura 3 – Principales debilidades de las EMP con respecto a la implementación del Perfil básico de ISO/IEC 29110 (Frecuencia).

El diseño arquitectural y detallado, la integración de componentes de software y el aseguramiento de la calidad mediante verificación y validación son preocupaciones de implementación de software que, debido a las propiedades intrínsecas de estas actividades, podrían abordarse con la orientación y las recomendaciones correctas que ofrece automáticamente una herramienta de soporte.

RQ3. ¿Qué herramientas de soporte existen para la implementación del estándar ISO/IEC 29110?

De acuerdo con la RQ3, relacionada con las herramientas de soporte utilizadas para la implementación estándar ISO/IEC 29110, de los ocho estudios primarios, cuatro de ellos hacen referencia al uso de las herramientas de soporte (PS01, PS06, PS07, PS08) y también hacen referencia al uso de material de soporte. La Figura 4, muestra la frecuencia de uso de herramientas de soporte y material de soporte para implementar ISO/IEC 29110. Como se muestra en la Figura 4, las herramientas más utilizadas para apoyar la implementación de ISO/IEC 29110 son las herramientas internas y Bugzilla®.

Una herramienta interna es un software que una EMP desarrolla para soportar su propio proceso, por lo general, este software no es para uso público y se usa para abordar solo algunas áreas de proceso específicas como se menciona en PS01 y PS08.

Por otro lado, los materiales de soporte más utilizados son los paquetes de despliegue (DP por sus siglas en inglés); Los elementos de un DP típico son la descripción de procesos, actividades, tareas, pasos, roles, productos, plantillas, listas de verificación, ejemplos, referencias y mapeo de estándares y modelos, y una lista de herramientas que pueden documentar el proceso. Sin embargo, hay un inconveniente para los DP, estas

guías están destinadas a ser utilizables para cualquier EMP, sin particularizar ninguna solución específica. (Kreiner et al., 2016), y también, un DP no puede automatizar un proceso, por esta razón, este estudio solo se enfoca en herramientas de soporte.



Figura 4 – Frecuencia de uso de herramientas de soporte y material de soporte.

La Tabla 3, describe el porcentaje de cobertura que tiene cada herramienta para apoyar la implementación del Perfil Básico de ISO/IEC 29110. Este porcentaje de cobertura se calculó otorgando a cada tarea del Perfil Básico ISO/IEC 29110 un peso específico, de la siguiente manera: los procesos de Gestión de Proyectos (PM) e Implementación de Software (SI) tienen el mismo peso, por lo tanto, PM = 50% e SI = 50% de las tareas que debe cubrir una herramienta de soporte.

- PM tiene 26 tareas, por lo tanto, al completar una tarea de PM, una herramienta de soporte cubre un 1.9% del proceso.
- SI tiene 41 tareas, por lo tanto, al completar una tarea de SI, una herramienta de soporte cubre un 1.2% del proceso.

Tal como se muestra en la Tabla 3, las herramientas más completas para la implementación de ISO/IEC 29110 son: ISO 29110 in-a-box®, GForge® y Jira®, señalando ISO 29110 in-a-box® debido a que es la única herramienta cuya cobertura es del 100%. Sin embargo, hay un inconveniente para estas 3 herramientas de soporte, todas están bajo licencia privada e implican un costo de uso, incluso los costos de uso aumentan dependiendo de la cantidad de usuarios o la cantidad de proyectos, esto aumentaría el presupuesto que las EMP deben invertir para su programa de SPI, y podría crear la percepción para las EMP de que no hay un retorno de la inversión, incluso si mejoran sus procesos (Rory et al., 2015).

Herramienta de soporte	PM.1	PM.2	PM.3	PM.4	SI.1	SI.2	SI.3	SI.4	SI.5	SI.6	Cobertura
SVN (PS07)	1.9	3.8		1.9							7.6%
SharePoint (PS07)	1.9	3.8		1.9							7.6%
OpenOffice Calc (PS07)	15.2					1.2	2.4	1.2	4.8	1.2	26%
Microsoft Visio (PS07)	1.9					2.4					4.3%
Jira (PS07)	13.3	5.7			1.2	1.2		1.2	1.2		23.8%
ISO 29110 in a box (PS07)	28.5	11.4	5.7	3.8	2.4	8.4	9.6	8.4	13.2	7.2	100%
GForge (PS07)	15.2	9.5		1.9		2.4	2.4	2.4	2.4	1.2	37.4%
CVS (PS07)	1.9	3.8		1.9							7.6%
Herramienta interna (PS01)	1.9	3.8		1.9							7.6%
Herramienta interna (PS08)		1.9	1.9								3.8%
Bugzilla (PS01, PS07)								2.4	1.2		3.6%

Tabla 3 – Cobertura (en porcentaje) de herramientas identificadas para las actividades del perfil básico de ISO/IEC 29110.

3. Conclusiones y trabajo futuro

Se realizó una Revisión Sistemática de la Literatura para identificar el estado actual de las EMP y el uso de herramientas de soporte para implementar ISO/IEC 29110 centrado en tres elementos clave: resultados obtenidos después de la implementación, principales debilidades y herramientas de soporte. De un conjunto total de 299 estudios, solo 8 de ellos fueron seleccionados como estudios primarios para esta investigación. Los resultados más relevantes indican que la implementación de un programa de Mejora de Procesos de Software (SPI), específicamente sobre el estándar ISO/IEC 29110, brinda beneficios y oportunidades para el crecimiento de las EMP y, por lo tanto, un incremento en la calidad de sus productos de software, manteniendo la competitividad en el mercado y mejorando las relaciones con sus clientes.

Con la ejecución de la RSL también se identificaron las principales debilidades de las EMP cuando implementan ISO/IEC 29110, las debilidades más relevantes están relacionadas con: Diseño arquitectural y detallado, integración de componentes de software, aseguramiento de la calidad, verificación y validación, y casos y procedimientos de prueba, entre otros.

Otros resultados obtenidos de la RSL indican que la mayoría de estas EMP solo usan los Paquetes de Despliegue como material de soporte. En cambio, solo unas pocas EMP utilizan herramientas de soporte para facilitar la implementación de ISO/IEC 29110. Las

herramientas más completas para la implementación de ISO/IEC 29110 son ISO 29110 in-a-box[®], GForge[®] y Jira[®], pero solamente ISO 29110 in-a-box es la única herramienta que tiene una cobertura completa para una implementación correcta. Sin embargo, una gran desventaja para estas 3 herramientas de soporte es que todas están bajo licencia privada e implican un costo de uso que no es asequible para las EMP, lo que afecta negativamente su percepción de la implementación de un programa de SPI.

Con base en los hallazgos mencionados anteriormente, como trabajo futuro, se está creando un sistema de software destinado a brindar soporte para implementar el estándar ISO/IEC 29110, a fin de cubrir las principales debilidades detectadas y alentar la visión de las EMP para mejorar la calidad de sus productos de software mientras se facilita una implementación exitosa de ISO/IEC 29110. Para las EMP esto puede ser un factor para ganar competitividad en el mercado.

Referencias

- Clarke, P., & Yilmaz, M. (2017). Current Challenges and Proposed Software Improvement Process for VSEs in Developing Countries. *Computer Standards and Interfaces*, 54, 117–118. DOI: <https://doi.org/10.1016/j.csi.2017.03.004>
- Hernandez, L., Munoz, M., Mejia, J., & Pena, A. (2017). Gamification in software engineering teamworks: A systematic literature review. In: *Proceedings of 2016 International Conference on Software Process Improvement (CIMPS)*, pp. 1–8. Aguascalientes, Mexico: IEEE. DOI: <https://doi.org/10.1109/cimps.2016.7802799>
- International Standard Organization (2004). ISO/IEC 15504-1:2004, Information Technology – Process assessment – Part 1: Concepts and vocabulary. 2004. DOI: <https://doi.org/10.1007/s11367-011-0297-3>
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *Engineering*, 2, 1051. DOI: <https://doi.org/10.1145/1134285.1134500>
- Kreiner, C., O'Connor, R.V, Poth, A., & Messnarz, R. (2016). Self-assessment as a Path to Improvement and Recognition for Small Entities. In: *Proceedings of 23rd European Conference on Systems, Software and Services Process Improvement (EuroSPI 2016)*, pp. 514–526. DOI: <https://doi.org/10.1007/978-3-319-44817-6>
- Laporte, C. Y., & O'Connor, R. V. (2016). A Multi-case Study Analysis of Software Process Improvement in Very Small Companies Using ISO/IEC 29110. In: *Proceedings of 23rd European Conference on Systems, Software and Services Process Improvement (EuroSPI 2016)*, pp. 30–44. DOI: <https://doi.org/10.1007/978-3-319-44817-6>
- Lohier, P., & Rodríguez-Dapena, P. (2016). Self-assessment as a Path to Improvement and Recognition for Small Entities. In: *Proceedings of 23rd European Conference on Systems, Software and Services Process Improvement (EuroSPI 2016)*, pp. 514–526. DOI: <https://doi.org/10.1007/978-3-319-44817-6>

- Maciaszek, L. A., & Loucopoulos, P. (2011). Evaluation of Novel Approaches to Software Engineering: 5th International Conference, ENASE 2010 Athens, Greece, July 22-24, 2010 Revised Selected Papers. Communications in Computer and Information Science, 230, 163–177. DOI: <https://doi.org/10.1007/978-3-642-23391-3>
- Mejia, J., Íñiguez, F., & Muñoz, M. (2017). Data Analysis for Software Process Improvement: A Systematic Literature Review. Advances in Intelligent Systems and Computing, 3, 571. DOI: <https://doi.org/10.1007/978-3-319-56541-5>
- Mejia, J., Muñoz, M., Rocha, Á., & Calvo-Manzano, J. (2016). Establishing the State of the Art of Frameworks, Methods and Methodologies Focused on Lightening Software Process: A Systematic Literature Review. Advances in Intelligent Systems and Computing, 405. DOI: <https://doi.org/10.1007/978-3-319-26285-7>
- Mejia, J., Peralta, F. A., & Munoz, M. (2017). Next Generation Systems – Scope and Application of Intrusion Detection and Prevention Systems (IDPS) a Systematic Literature Review. In: Proceedings of 12th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-7. Lisbon: IEEE.
- Muñoz, M., Laporte, C. Y., & Mejia, J. (2018). Implementación del Estándar ISO/IEC 29110 en Centros de Desarrollo de Software de Universidades Mexicanas: Experiencia del Estado de Zacatecas. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, (29), 43–54. DOI: <https://doi.org/10.17013/risti.29.43-54>
- Munoz, M., Mejia, J., & Lagunas, A. (2018). Implementation of the ISO/IEC 29110 standard in agile environments: A systematic literature review. In: Iberian Conference on Information Systems and Technologies, CISTI, 2018-June 1–6. DOI: <https://doi.org/10.23919/CISTI.2018.8399332>
- Nonoyama, T., Kabaale, E., Wen, L., David, T., & Wang, Z. (2016). Integrating Culture Awareness and Formalisation in Software Process Assessment and Improvement for Very Small Entities (VSEs) Tatsuya. In: Porceedings of 23rd European Conference on Systems, Software and Services Process Improvement (EuroSPI 2016) (Vol. 2). Springer International Publishing. DOI: <https://doi.org/10.1007/978-3-319-44817-6>
- O'Connor, R. V., & Laporte, C. Y. (2012). Software project management in very small entities with ISO/IEC 29110. Communications in Computer and Information Science, 301, 330–341. DOI: https://doi.org/10.1007/978-3-642-31199-4_29
- Oktaba, H., & Martínez, A. (2005). Modelo de Procesos para la Industria de Software: MoProSoft Brain Mapping View project An Autonomic Software Architecture For Knowledge Management In Medical Diagnosis View project. DOI: <https://doi.org/10.13140/2.1.2229.5043>
- Rory, T. R., O'Connor, R. V., & Dorling, A. (2015). Evaluating VSEs Viewpoint and Sentiment Towards the ISO/IEC 29110 Standard: A Two Country Grounded Theory Study. Communications in Computer and Information Science, 526, 114–127. DOI: <https://doi.org/10.1007/978-3-319-19860-6>

SEI (2010). CMMI® para Desarrollo, Versión 1.3. Equipo del Producto CMMI. DOI: <http://www.sei.cmu.edu>

Selleri Silva, F., Soares, F. S. F., Peres, A. L., Azevedo, I. M. De, Vasconcelos, A. P. L. F., Kamei, F. K., & Meira, S. R. D. L. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, 58, 20–43. DOI: <https://doi.org/10.1016/j.infsof.2014.09.012>

Apéndice A: Lista de Estudios Primarios

C. Y. Laporte and R. V. O'Connor, "A Multi-case Study Analysis of Software Process Improvement in Very Small Companies Using ISO/IEC 29110," *23rd Eur. Conf. Syst. Softw. Serv. Process Improv. (EuroSPI 2016)*, vol. 1, pp. 30–44, 2016. <https://doi.org/10.1007/978-3-319-44817-6>

X. Larrucea and I. Santamaria, "Towards a survival Analysis of Very Small Organisations," *23rd European Conference on Systems, Software and Services Process Improvement (EuroSPI 2016)*, pp. 599–609, 2016 <https://doi.org/10.1007/978-3-319-44817-6>

M. Muñoz, J. Mejía, and C. Y. Laporte, "Reinforcing Very Small Entities Using Agile Methodologies with the ISO/IEC 29110", pp. 865, 88–98, 2019. <https://doi.org/10.1007/978-3-030-01171-0>

N. Wongsai, V. Siddoo and R. Wetprasit, "Factors of influence in software process improvement: An ISO/IEC 29110 for very-small entities," *Proceedings - 2015 7th International Conference on Information Technology and Electrical Engineering: Envisioning the Trend of Computer, Information and Engineering, ICITEE 2015*, 12–17. 2015. <https://doi.org/10.1109/ICITEED.2015.7408904>

C. Y. Laporte and R. V. O'Connor, "Implementing process improvement in very small enterprises with ISO/IEC 29110: A multiple case study analysis. *Proceedings - 2016 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016*, 125–130. 2017. <https://doi.org/10.1109/QUATIC.2016.033>

C. Y. Laporte, M. Muñoz, J. Mejía and R. V. O'Connor, "Applying Software Engineering Standards in Very Small Entities: From Startups to Grownups," *IEEE Software*, 35(1), 99–103. 2017. <https://doi.org/10.1109/MS.2017.4541041>

M. Muñoz, J. Mejía and A. Lagunas, "Implementation of the ISO/IEC 29110 standard in agile environments: A systematic literature review," *Iberian Conference on Information Systems and Technologies, CISTI, 2018–June* 1–6. 2018. <https://doi.org/10.23919/CISTI.2018.8399332>

M. Muñoz, C. Y. Laporte and J. Mejía, "Implementation of ISO/IEC 29110 in Software Development Centers from Mexican Universities: An experience of the Zacatecas State," *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (29), 43–54. 2019. <https://doi.org/10.17013/risti.29.43-54>

Identificación de herramientas como soporte en la implementación de la Norma ISO/IEC 29110 (perfil básico)

Jezreel Mejía¹, Edgar Bonilla¹, Elizabeth Villanueva¹, Einar Serna¹, Israel Faustino¹, Alessandro Milán¹.

{jmejia, edgar.bonilla, elizabeth.villanueva, einar.serna, israel.faustino, hector.milan}@cimat.mx

¹Centro de Investigación en Matemáticas, (CIMAT, A.C.), Unidad Zacatecas, México, Calle Lasec, Andador Galileo Galilei. Andador 3, Lote 7, CP 98160, Zacatecas, México.

DOI: 10.17013/risti.36.15-33

Resumen: Actualmente, se están dando a conocer normas orientadas a las denominadas Entidades Muy Pequeñas (EMPs) como lo es la norma ISO/IEC 29110. Aun cuando se están creando y dando a conocer nuevas normas orientadas a las EMPs, existe una carencia de herramientas que permitan automatizar o dar soporte en la implementación de estas nuevas normas. Por lo tanto, este artículo, presenta una propuesta de herramientas que den soporte en la implementación a las actividades y tareas de los dos procesos que contiene la norma ISO/IEC 29110.

Palabras-clave: ISO/IEC 29110; Herramientas; Gestión de Proyectos; Implementación del Software.

Identification of tools to support ISO/IEC 29110 implementation (basic profile)

Abstract: Currently, standards oriented to the so-called Very Small Entities (EMPs) such as the ISO / IEC 29110 standard are becoming known. However, although new standards oriented to EMPs are being created, there is a lack of tools that allow automating or supporting the implementation of these new standards. Therefore, this article presents a tools proposal that support the implementation of the activities and tasks of the two processes of the ISO / IEC 29110 standard.

Keywords: ISO/IEC 29110; Tools; Project Management; Software Implementation.

1. Introducción

En la actualidad, existen en México poco más de 4.1 millones de entidades, de las cuales el 97.3% son consideradas como micro y el 2.7% están dentro del rango de pequeña y mediana esto según datos obtenidos de la encuesta ENAPROSE 2018 llevada a cabo por el INEGI (INEGI, 2020). Las muy pequeñas empresas (VSEs por sus siglas en inglés)

es una entidad que esta definida como “una empresa”, organización, departamento o proyecto que tiene a su cargo de 1 a 25 personas (ISO/IEC, 2018), (NYCE, 2020). En este contexto, se estan dando a conocer normas orientadas a las denominadas Entidades Muy Pequeñas (EMPs) de desarrollo de software como lo es la norma ISO/IEC 29110. El uso de este tipo de normas en organizaciones de tipo VSEs les ayuda a incrementar su competitividad, asi como la calidad de sus desarrollos de software (O'Connor, 2017).

La norma ISO/IEC 29110 es un estándar internacional de ingeniería de software que define los perfiles del ciclo de vida para las EMPs que desarrollan software no crítico. Su objetivo, es satisfacer las necesidades específicas de las EMPs y abordar el problema de la baja adopción de normas por parte de estas organizaciones (NYCE. 2020), (ISO/IEC 29110, 2012). Sin embargo, factores como falta de cultura de procesos, conocimiento de normas o metodologías de desarrollo, falta de personal y tiempo para llevar de una manera adecuada la implementacion de esta norma, es complicado para las EMPs implementar de manera adecuada la norma. Además, la falta de herramientas que permitan automatizar las actividades o tareas indicadas en las normas es un obstaculo difícil de eliminar y que propicia una barrera para una buena implementacion de este tipo de normas.

Por lo tanto, el objetivo de este trabajo es facilitar una propuesta de implementación de la norma ISO/IEC 29110, proporcionando un ciclo de desarrollo de software con base a los dos procesos Gestión de Proyectos (GP) e Implementación de Software (IS) del perfil básico de esta norma. Además de proponer el uso de herramientas para estos procesos en apoyo a la automatizacion o soporte al registro de informacion o datos que generan o requiere cada tarea y/o actividad de los procesos GP e IS.

El contenido de este documento está estructurado de la siguiente forma: en la sección 2, se presenta una breve introducción acerca de la norma ISO/IEC 29110; en la sección 3, se presenta la propuesta que incluye: el ciclo de desarrollo de software propuesto y las herramientas que dan soporte a las actividades y tareas de los dos procesos GP e IS de esta norma; en la sección 4 muestra el estudio de caso y por último en la sección 5, se muestran las conclusiones y trabajo futuro.

2. Descripción general del ISO/IEC 29110

La ISO (Organización Internacional para la Normalización) y la IEC (Comisión Electrotécnica Internacional) forman el sistema especializado para la normalización mundial. Los organismos nacionales miembros de ISO e IEC participan en el desarrollo de las Normas Internacionales por medio de comités técnicos establecidos por la organización respectiva, para atender campos particulares de la actividad técnica (ISO/IEC). La ISO/IEC se divide en cinco partes (ISO/IEC, 2018): Parte 1: Visión general, Parte 2: Marco de trabajo y taxonomía, Parte 3: Guía de evaluación, Parte 4: Especificaciones de perfil, Parte 5: Guía de gestión e ingeniería: Provee una guía de implementación sobre gestión e ingeniería para el Perfil Básico del Grupo del Perfil Genérico especificado en la ISO/IEC 29110. La Parte 5-1 (ISO/IEC, 2012) define el Perfil Básico como el desarrollo de software de una sola aplicación por un solo equipo de proyecto sin ningún riesgo especial o factores situacionales. Esta ultima parte, es la utilizada para ser analizada.

2.1. Análisis de la estructura de ISO/IEC 29110

La norma ISO/IEC 29110, está estructurada mediante dos procesos que son Gestión de Proyectos (GP) e Implementación del Software (IS), además, estos cuentan con objetivos y actividades.

En la Tabla 1, se muestra los roles identificados en la norma ISO/IEC 29110, tanto para GP como para el proceso de Implementación de Software IS.

Rol	Proceso
Cliente (CL)	GP e IS
Gestor del Proyecto (GP)	GP e IS
Líder Técnico (LT)	GP e IS
Equipo de Trabajo (ET)	GP e IS
Analista (AN)	IS
Diseñador (DIS)	IS
Programador (PR)	IS

Tabla 1 – Roles en GP e IS.

2.2. Proceso GP

PMBOK define la gestión de proyecto de la siguiente manera “es la aplicación de conocimientos, habilidades, herramientas y técnicas para proyectar actividades que cumplan con los requisitos del proyecto” (Project Management Institute, 2013).

El propósito del proceso GP es establecer y llevar a cabo de manera sistemática las tareas del proyecto de implementación de Software, las cuales permiten cumplir con los Objetivos del proyecto en calidad, tiempo y costo esperado (ISO/IEC 29110, 2012).

El proceso GP, a través de la realización de cada una de sus actividades, pretende cumplir 7 objetivos. Además, este proceso consta de cuatro actividades y cada una de ellas con un número específico de tareas:

- GP.1. Planificación del Proyecto: 15 tareas.
- GP.2. Ejecución del Plan del Proyecto: 6 tareas.
- GP.3. Evaluación y control del Proyecto: 3 tareas.
- GP.4. Cierre del Proyecto: 2 tareas.

En la Tabla 2, se muestra los productos que la ISO/IEC 29110 señala como necesarios para realizar las tareas del proceso de GP, así como también se muestra su origen correspondiente.

Nombre	Origen
Enunciado del Trabajo	• Cliente
Configuración del Software	• Implementación de software
Solicitud de Cambio	• Cliente • Implementación de software

Tabla 2 – Productos de entrada GP.

En la Tabla 3, se muestra los productos que la ISO/IEC 29110 señala como aquellos que se generan al realizar las tareas del proceso de GP, así como también se muestra su destino correspondiente. Los productos internos y los que deben ser guardados en el repositorio del proyecto.

Nombre	Destino
Plan de Proyecto	• Implementación de Software
Acta de Aceptación	• Alta Dirección
Repositorio del Proyecto	• Implementación de Software
Acta de Reunión	• Cliente
Configuración de Software	• Cliente

Tabla 3 – Productos de salida GP.

2.3. Proceso IS

El propósito del proceso de Implementación de Software es la realización sistemática de las actividades de Análisis, Diseño, Construcción, Integración y Pruebas para los productos de Software, nuevos o modificados, de acuerdo con los requisitos especificados ISO/IEC 29110, 2012).

El proceso IS, a través de la realización de cada una de sus actividades, pretende cumplir 7 objetivos, además, este proceso de IS consta de seis actividades y cada una de ellas con un numero específico de tareas:

- IS.1. Inicio de la Implementación de Software: 2 tareas.
- IS.2. Análisis de Requisitos del Software: 7 tareas.
- IS.3. Arquitectura y Diseño Detallado del Software: 8 tareas.
- IS.4. Construcción de Software: 7 tareas.
- IS.5. Integración y Pruebas del Software: 11 tareas.
- IS.6. Entrega del Producto: 8 tareas.

En la Tabla 4, se muestra los productos que la ISO/IEC 29110 señala como necesarios para realizar las tareas del proceso de IS, así como también se muestra su origen correspondiente.

Nombre	Origen
Enunciado del Trabajo	• Cliente
Plan del Proyecto Gestión de Proyectos	• Plan del Proyecto Gestión de Proyectos
Repositorio del Proyecto Gestión de Proyectos	• Repositorio del Proyecto Gestión de Proyectos

Tabla 4 – Productos de entrada IS.

En la Tabla 5, se muestra los productos que la ISO/IEC 29110 señala como aquellos que se generan al realizar las tareas del proceso de IS, así como también se muestra su destino correspondiente.

Nombre	Destino
Configuración de Software: • Especificación de Requisitos • Diseño de Software • Registro de Trazabilidad • Componente de Software • Software • Casos de Prueba y Procedimientos de Prueba • Reporte de Pruebas • Manual de Operación • Manual de Usuario • Manual de Mantenimiento	Gestión de Proyectos
Solicitud de Cambio	Gestión de Proyectos

Tabla 5 – Productos de salida IS.

3. Propuesta

Para identificar las herramientas necesarias como soporte en la implementación de la Norma ISO/IEC 29110, en el siguiente apartado, se muestra la propuesta de ciclo de desarrollo, con base en un análisis de la misma norma.

3.1. Ciclo de Desarrollo

Se analizaron las actividades y tareas de la ISO/IEC 29110, como resultado se observa un flujo de trabajo combinado. En la Figura 1, se muestra el ciclo de desarrollo propuesto a partir del análisis realizado.

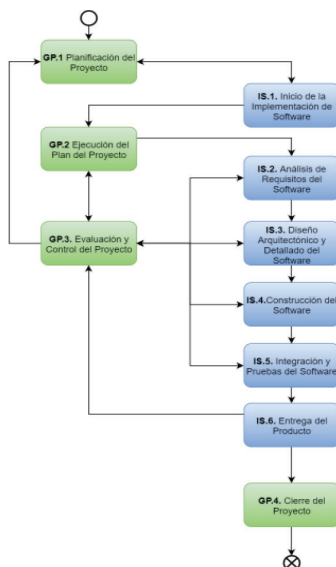


Figura 1 – Propuesta de ciclo de desarrollo.

Como puede observarse en la Figura 1, el ciclo inicia en GP.1 Planificación del Proyecto, en donde se documenta los detalles de la planificación necesarios para la gestión del proyecto.

Seguido de GP.1, le sigue la actividad IS.1 Inicio de la Implementación de Software, en donde se asegura que el Plan de Proyecto establecido en la actividad anterior, es comprendido por el equipo de desarrollo y se establece el ambiente de trabajo. De existir algún desacuerdo identificado por el equipo de trabajo, se puede regresar a la actividad anterior para la modificación o adaptación del plan de proyecto.

La siguiente actividad es GP.2. Ejecución del Plan del Proyecto, en donde se implementa el plan documentado en el proyecto, esta actividad inicia la actividad IS.2. Análisis de Requisitos del Software, donde se analiza los requisitos acordados con el Cliente y establece los requisitos del proyecto validados.

Desde la actividad IS.2, le sigue de manera secuencial las actividades IS.3. Diseño Arquitectónico y Detallado del Software, donde se transforman los requisitos del Software en la arquitectura de software del sistema y en el diseño detallado del Software; IS.4. Construcción del Software, donde se desarrolla el código y datos del Software a partir del Diseño de Software; IS.5. Integración y Pruebas del Software, donde se asegura que los componentes de software integrados satisfacen los requisitos del software.

Las actividades IS.2, IS.3, IS.4 e IS.5 son monitoreadas en paralelo por la actividad GP.3. Evaluación y Control del Proyecto, para evaluar el desempeño del plan contra los compromisos documentados. En caso de existir algún cambio aprobado, GP.3 regresa a GP.1 para realizar las modificaciones necesarias en el plan de trabajo. En caso de que el cambio no afecte el plan de trabajo, regresa a GP.2 para seguir con la secuencia de actividades de IS (IS.2, IS.3, IS.4 e IS.5). De no existir algún cambio, simplemente regresa a monitorear la actividad de IS en que se quedó.

Al llegar a la actividad IS.6. Entrega del Producto, se integra el producto de software para el Cliente. Ya que el desarrollo de este producto ha sido monitoreado desde el inicio de su construcción hasta este punto, en esta actividad se hace la entrega del producto al Cliente de acuerdo con las instrucciones de entrega. De existir alguna irregularidad con el producto, se regresa a GP.3 a evaluar y hacer los cambios correspondientes, en caso contrario se pasa a la actividad GP.4. Cierre del Proyecto, donde se documenta los detalles de planificación necesarios para gestionar el proyecto.

Establecido el ciclo de desarrollo propuesto y la interacción de las actividades tanto de GP como de IS, es necesario dar soporte con herramientas para llevar a cabo algunas de las actividades y tareas en ambos procesos de esta norma.

3.2. Herramientas

Las herramientas son una parte importante de cualquier proceso de desarrollo de software. Permiten la automatización, y por lo tanto, una mejor adaptación del proceso entre los miembros del equipo de desarrollo. Por ello, es importante definir y estandarizar entre el equipo, aquellas herramientas que permitan la mejor adaptación de trabajo que implica el proceso de desarrollo de software.

Para proponer las herramientas que ayuden a cada tarea en las actividades que conforman el proceso de GP e IS, se analizaron cada una de las tareas y la relación que tienen con los *productos de entrada, internos y salida* que se generan. Como resultado de este análisis se identificaron las siguientes herramientas.

- **Podio:** Es una herramienta para la gestión de tareas y proyectos, dispone de una versión gratuita de hasta 10 usuarios. Con esa herramienta se pueden organizar reuniones, llamadas o emails para tener controlado los proyectos. Cada proyecto hace referencia a un cliente, así que es fácil buscar conversaciones y resultados anteriores relacionados con un cliente actual, además se puede integrar con otras herramientas como Google Drive, Dropbox, OneDrive, Google Calendar, Excel, Microsoft Exchange (Podio, 2020).
- **Suite de Google:** ofrece de manera gratuita diferentes servicios de oficina que podrían ser de utilidad en el proceso de implementación de la ISO/IEC 29110, tales como el correo electrónico, docs, slides, hojas de cálculo, formularios, servicio de almacenamiento, y más.
 - **Documentos Google:** es un procesador de textos online que permite crear documentos, darles formato y edición en tiempo real por parte de los colaboradores registrados. Google Docs cuenta también con un historial de versiones ordenadas por fecha y por autor del cambio realizado (Google Docs, 202).
 - **Hojas de cálculo Google:** es un tipo de procesador de documentos que permiten la manipulación de datos dispuestos en formas de tablas. Google permite, mediante su arquitectura basada en la nube, la colaboración entre las personas del equipo a través de la edición en tiempo real (Google Suite, 2020).
 - **Draw.io:** es una herramienta que facilita la elaboración de diagramas en línea. Draw.io tiene una interfaz que es bastante sencilla y fácil de usar, además dispone de una gran variedad de formas y diseños predeterminados que pueden ser modelados de acuerdo con la estandarización de diagramas seleccionada (Chrome web store, 2020).
 - **Drive:** es un sistema de almacenamiento en la nube, que además integra funciones colaborativas en Microsoft Office, permitiendo crear, almacenar y compartir cualquier documento. Actualmente Google Drive ofrece, en su versión gratuita, 15 GB de almacenamiento (Google Drive, 2020).
- **GitLab:** es una aplicación para todo el ciclo de vida del desarrollo de software. Desde la planificación del proyecto y la gestión del código fuente hasta la Configuración de Integración/Despliegue Continuo (CI/CD), la supervisión y la seguridad. GitLab es una única aplicación para todo el ciclo de vida de DevOps (GitLab, 2020).
- **Git:** es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta muy grandes, con velocidad y eficiencia. Git es fácil de aprender y tiene una huella pequeña con un rendimiento increíblemente rápido. Supera las herramientas de SCM como Subversion, CVS, Perforce y ClearCase con características como ramificación local barata, áreas de preparación convenientes y múltiples flujos de trabajo (Git, 2019).

- Jest es un framework de prueba de JavaScript, con un enfoque en la simplicidad. Funciona con proyectos utilizando: Babel, TypeScript, Node, React, Angular y Vue. El objetivo de Jest es trabajar fuera de la caja, configurando libremente en la mayoría de los proyectos de JavaScript (Jest, 2020).

Cabe mencionar que como resultado del análisis de las tareas y las herramientas antes descritas, algunos productos de salida deben combinarse al momento de elaborar un producto de entrada, interno o de salida.

3.2.1. Herramientas para el Proceso GP

Tras el análisis del conjunto de las 26 tareas que se indican en el proceso GP, las herramientas propuestas para este proceso son: Google documento (Doc), Google Hoja de cálculo (HC), Google Drive (D), Podio (), Git (G).

A continuación, se muestran en la Tabla 6 a la 9, las tareas (columna 1) de las cuatro actividades respectivamente, que son cubiertas por las herramientas propuestas (columnas 3 – 7) con una “X” y que en algunas tareas se deben complementar las diferentes herramientas que se proponen.

La actividad GP 1. Planificación del Proyecto. La Tabla 6, muestra las herramientas que cubren las 15 tareas, las cuales son:

Tarea	Producto	Google			P	G
		Doc	HC	D		
GP.1.1 Revisar al Enunciado de Trabajo.	Enunciado de trabajo	X				
GP.1.2 Definir con el cliente las instrucciones de entrega.	Enunciado de trabajo/ Plan de proyecto.	X				
GP.1.3 Identificar las tareas específicas a realizar.	Enunciado de trabajo/ Plan de proyecto.				X	
GP.1.4 Establecer la duración estimada para realizar cada tarea.	Plan de proyecto				X	
GP.1.5 Identificar y documentar los recursos.	Enunciado de trabajo/ Plan de proyecto.	X	X			
GP.1.6 Establecer la composición del equipo de trabajo.	Plan de Proyecto				X	
GP.1.7 Asignar las fechas de inicio y fin estimadas para cada tarea.	Plan de proyecto				X	
GP.1.8 Calcular y documentar el esfuerzo y costo estimado del proyecto.	Plan de proyecto		X		X	
GP.1.9 Identificar y documentar los riesgos.	Plan de proyecto		X			
GP.1.10 Documentar la estrategia de control de versiones.	Enunciado de trabajo/ plan de proyecto	X				X
GP.1.11 Generar el plan de proyecto integrando los elementos identificados.	Enunciado de trabajo/ plan de proyecto	X				

Tarea	Producto	Google			P	G
		Doc	HC	D		
GP.1.12 Incluir la descripción del producto, alcance, objetivos y entregables en el plan de proyecto.	Enunciado de trabajo/ plan de proyecto	X				
GP.1.13 Verificar y obtener aprobación del plan del proyecto.	Plan de proyecto/ resultados de verificación.	X				
GP.1.14 Revisar y obtener la aprobación del plan del proyecto.	Plan de proyecto/Acta de reunión.	X				
GP.1.15 Establecer el repositorio del proyecto usando la estrategia de control de versiones.	Plan de proyecto/ Repositorio del proyecto			X		X

Tabla 6 – Propuesta de herramientas para GP.1.

La Actividad 2. Ejecución del proyecto. La Tabla 7, muestra las herramientas que cubren las 6 tareas, las cuales son:

Tarea	Producto	Google			P	G
		Doc	HC	D		
GP.2.1 Monitorear la ejecución del Plan del Proyecto y registrar la información actual en el Reporte de Avance.	Plan de proyecto/Reporte de Avance				X	
GP.2.2 Analizar y evaluar el impacto en costo, tiempo e impacto técnico de la Solicitud de Cambio.	Solicitud de cambio/plan de proyecto	X		X	X	
GP.2.3 Conducir reuniones de revisión con el Equipo de Trabajo.	Plan de proyecto/Reporte de avance/acciones correctivas/ acta de reunión	X		X	X	
GP.2.4 Realizar reuniones con el Cliente, de las cuales se registran acuerdos y se dará seguimiento hasta su conclusión.	Plan de proyecto/Reporte de avance/solicitud de cambio/ correctivas/acta de reunión	X		X	X	
GP.2.5 Realizar el Respaldo del Repositorio del Proyecto de acuerdo con la Estrategia de Control de Versiones.	Respaldo del repositorio del proyecto			X		X
GP.2.6 Realizar la recuperación del Repositorio del Proyecto utilizando el Respaldo del Repositorio del Proyecto, en caso de ser necesario.	Respaldo del repositorio del proyecto/Repositorio del proyecto			X		X

Tabla 7 – Propuesta de herramientas para GP.2.

La Actividad 3. Evaluación y control del proyecto. La Tabla 8, muestra las herramientas que cubren las 3 tareas a desarrollar:

Tarea	Producto	Google			P	G
		Doc	HC	D		
GP.3.1 Evaluar el progreso del proyecto con respecto al Plan del Proyecto.	Plan de proyecto/ Reporte de Avance				X	
GP.3.2 Establecer acciones para corregir desviaciones o problemas e identificar riesgos que amenacen el cumplimiento del plan.	Reporte de avance/ acciones correctivas	X			X	
GP.3.3 Identificar cambios a requisitos y/o al Plan del Proyecto para hacer frente a desviaciones importantes, potenciales riesgos o problemas relativos al cumplimiento del plan.	Reporte de avance/ Solicitud de cambio	X	X		X	

Tabla 8 – Propuesta de herramientas para GP.3.

Actividad 4. Cierre del proyecto. La Tabla 9, muestra las herramientas que cubren las 2 tareas a desarrollar:

Tarea	Producto	Google			P	G
		Doc	HC	D		
GP.4.1 Formalizar la conclusión del proyecto de acuerdo con las Instrucciones de Entrega establecidas en el Plan del Proyecto.	Plan de proyecto/ Configuración de Software/acta de Aceptación	X				
GP.4.2 Actualizar el Repositorio del Proyecto.	Configuración de Software/ Repositorio del proyecto	X		X		X

Tabla 9 – Propuesta de herramientas para GP.4.

3.2.2. Herramientas para el Proceso IS

Tras el análisis del conjunto de las 42 tareas que se indican en el proceso IS, las herramientas propuestas para este proceso son: Documentos (doc), Draw.io (DI), Podio (P), GitLab (GL) y Jest.

A continuación, se muestran en las Tablas de la 10 a la 15, las tareas (columna 1) de las cuatro actividades respectivamente, que son cubiertas por las herramientas propuestas (columnas 3 – 7) con una “X” y que en algunas tareas se deben complementar las diferentes herramientas que se proponen.

Actividad IS 1. Inicio de la Implementación de Software. La Tabla 10, muestra las herramientas que cubren las 2 tareas a desarrollar:

Actividad	Producto	Google		P	GL	Jest
		Doc	DI			
IS.1.1 Revisar el Plan del Proyecto actual con los miembros del Equipo de Trabajo.	Plan de Proyecto			X		
IS.1.2 Establecer o actualizar el ambiente de implementación.	Plan de proyecto/ Ambiente de implementación	X		X		

Tabla 10 – Propuesta de herramientas para IS.1.

Actividad IS 2. Análisis de Requisitos del Software. La Tabla 11, muestra las herramientas que cubren las 7 tareas a desarrollar:

Actividad	Producto	Google		P	GL	Jest
		Doc	DI			
IS.2.1 Asignar las fechas de inicio y fin estimadas para cada tarea.	Plan de proyecto/Reporte de avance			X		
IS.2.2 Documentar o actualizar la Especificación de Requisitos.	Plan de proyecto/ Especificación de requisitos	X				
IS.2.3 Verificar y obtener la aprobación de la Especificación de Requisitos.	Especificación de requisitos/Resultados de verificación/Solicitud de cambio.	X				
IS.2.4 Validar y obtener la aprobación de la Especificación de Requisitos.	Especificación de requisitos/Resultados de verificación	X				
IS.2.5 Documentar la versión preliminar del *Manual de Usuario o actualizar el manual existente.	Especificación de requisitos/Manual de usuario	X				
IS.2.6 Verificar y obtener la aprobación del *Manual de Usuario, si es apropiado.	Manual de usuario/ Especificación de requisitos/resultados de verificación/solicitud de cambio.	X				
IS.2.7 Incorporar la Especificación de Requisitos y el *Manual de Usuario a la Configuración de Software en la línea base.	Especificación de requisitos/ Manual de usuario/Configuración del software.	X				

Tabla 11 – Propuesta de herramientas para IS.2.

Actividad IS 3. Arquitectura y Diseño Detallado del Software. La Tabla 12, muestra las herramientas que cubren las 8 tareas a desarrollar:

Actividad	Producto	Google		P	GL	jestjs. io
		Doc	DI			
IS.3.1 Monitorear la ejecución del Plan del Proyecto y registrar la información actual en el Reporte de Avance.	Plan de proyecto			X		
IS.3.2 Comprender la Especificación de Requisitos.	Especificación de requisitos	X				
IS.3.3 Documentar o actualizar el Diseño de Software.	Especificación de requisitos/Diseño de software/Registro de Trazabilidad	X				
IS.3.4 Verificar y obtener la aprobación del Diseño de Software.	Diseño de software/ registro de trazabilidad/ especificación de requisitos/Resultados de verificación	X				
IS.3.5 Establecer o actualizar los Casos de Prueba y Procedimientos de Prueba.	Especificación de requisitos/diseño de software/Casos y procedimientos de prueba	X			X	X
IS.3.6 Verificar y obtener la aprobación de los Casos de Prueba y Procedimientos de Prueba.	Casos y procedimientos de prueba/Especificación de requisitos/Resultados de Verificación/Diseño de software	X				X
IS.3.7 Actualizar el Registro de Trazabilidad incorporando los Casos de Prueba y Procedimientos de Prueba.	Casos y procedimientos de prueba/Registro de trazabilidad	X				X
IS.3.8 Incorporar el Diseño de Software, y el Registro de Trazabilidad a la Configuración de Software como parte de la línea base.	Diseño de Software/ Casos y procedimientos de prueba/Registro de Trazabilidad/configuración del software	X			X	X

Tabla 12 – Propuesta de herramientas para IS.3.

Actividad IS 4. Construcción de Software. La Tabla 13, muestra las herramientas que cubren las 7 tareas a desarrollar:

Actividad	Producto	Google		P	GL	Jest
		Doc	DI			
IS.4.1 Asignar Tareas a los miembros del Equipo de Trabajo de acuerdo con cada rol, basado en el Plan del Proyecto actual.	Plan de proyecto			X		
IS.4.2 Entender el Diseño de Software.	Diseño de Software	X	X	X		
IS.4.3 Construir o actualizar los Componente de Software basados en la parte detallada del Diseño de Software.	Diseño de software/ Registro de Trazabilidad/ componente de software	X	X		X	

Actividad	Producto	Google		P	GL	Jest
		Doc	DI			
IS.4.4 Diseñar o actualizar los Casos de Pruebas unitarias y aplicarlos para verificar que los Componente de Software implementan la parte detallada de Diseño de Software.	Componente de software				X	X
IS.4.5 Corregir los defectos encontrados hasta lograr la prueba unitaria exitosa (logrando el criterio de salida).	Componente de software				X	X
IS.4.6 Actualizar el Registro de Trazabilidad incorporando Componente de Software contruidos o modificados.	Componente de software/ Registro de Trazabilidad	X			X	
IS.4.7 Incorporar Componente de Software y Registro de Trazabilidad para la Configuración de Software como parte de la línea base.	Componente de software/ Registro de trazabilidad/ configuración de software	X			X	

Tabla 13 – Propuesta de herramientas para IS.4.

Actividad IS 5. Integración y Pruebas del Software. La Tabla 14, muestra las herramientas que cubren las 11 tareas a desarrollar las cuales son:

Actividad	Producto	Google		P	GL	jestjs. io
		Doc	DI			
IS.5.1 Asignar Tareas a los miembros del Equipo de Trabajo de acuerdo con cada rol, basado en el Plan del Proyecto actual.	Plan de proyecto			X		
IS.5.2 Entender los Casos de Prueba y Procedimientos de Prueba. Establecer o actualizar el ambiente de prueba.	Casos y procedimientos de prueba/ Ambiente de prueba	X				X
IS.5.3 Integrar el Software usando los Componente de Software y actualizar los Casos de Prueba y Procedimientos de Prueba para las pruebas de integración, conforme sea necesario.	Componente de software/Casos y procedimientos de prueba/registro de trazabilidad/software	X			X	X
IS.5.4 Realizar pruebas de Software usando Casos de Prueba y Procedimientos de Prueba para la integración y documentar los resultados en el Reporte de Pruebas.	Software/Casos y procedimientos de prueba/Reporte de pruebas	X			X	X
IS.5.5 Corregir los defectos encontrados y realizar una prueba de regresión hasta satisfacer el criterio de finalización.	Software/Reporte de pruebas/Casos y procedimientos de prueba/Registro de trazabilidad	X			X	X
IS.5.6 Actualizar el Registro de Trazabilidad en caso de ser necesario.	Software/Registro de trazabilidad/	X			X	
IS.5.7 Documentar el *Manual de Operación o actualizar el manual actual, en caso de ser apropiado.	Software/Manual de operación	X				

Actividad	Producto	Google		P	GL	jestjs. io
		Doc	DI			
IS.5.8. Verificar y obtener la aprobación del *Manual de Operación, en caso de ser necesario.	Manual de Operación/ Software/Resultados de verificación	X			X	
IS.5.9 Documentar el *Manual de Usuario o actualizar el actual, si aplica.	Software/Manual de Usuario	X			X	
IS.5.10 Verificar y obtener la aprobación del *Manual de Usuario, en caso de ser necesario.	Software/Manual de usuario/resultados de verificación	X				
IS.5.11 Incorporar los Casos de Prueba y Procedimientos de Prueba, Software, Registro de Trazabilidad, Reporte de Pruebas, *Manual de Operación y *Manual de Usuario a la Configuración de Software como parte de la línea base.	Casos y procedimientos de prueba/Software/ Reporte de pruebas/ Registro de trazabilidad/Manual de operación/Manual de Usuario/Configuración de software	X			X	X

Tabla 14 – Propuesta de herramientas para IS.5.

Actividad IS 6. Entrega del Producto. La Tabla 15, muestras las herramientas que cubren las 8 tareas a desarrollar:

Actividad	Producto	Google		P	GL	Jest
		Docs	DI			
IS.6.1 Asignar Tareas a los miembros del Equipo de Trabajo de acuerdo con cada rol, basado en el Plan del Proyecto actual.	Plan de proyecto			X		
IS.6.2 Comprender la Configuración de Software.	Configuración del software	X			X	
IS.6.3 Documentar el Manual de Mantenimiento o actualizar el actual.	Configuración del Software/ Manual de Mantenimiento	X			X	
IS.6.4 Verificar y obtener la aprobación del Manual de Mantenimiento.	Manual de mantenimiento/ Configuración del software/ Resultado de verificación	X			X	
IS.6.5 Incorporar el Manual de Mantenimiento a la línea base de la Configuración de Software.	Configuración del software/ Manual de Mantenimiento	X			X	
IS.6.6 Llevar a cabo la entrega de acuerdo con las Instrucciones de Entrega.	Plan de proyecto/ Configuración del software	X		X	X	

Tabla 15 – Propuesta de herramientas para IS.6.

3.2.1. Cobertura global de Herramientas para el proceso GP e IS.

En la Tabla 16, se muestra el resumen de las tareas cubiertas por actividad por parte de las herramientas (columnas 3 – 10). En la última fila se encuentra el porcentaje de la cobertura por parte de la herramienta, teniendo como base las 68 (100%) tareas de los dos procesos.

Actividad	Tareas	Google							
		Docs	Hoja de Calculo	Drive	Draow.io	Podio	GitLab	Git	Jest
GP. 1.	15	8	3	1	0	5	0	2	0
GP. 2.	6	3	0	5	0	4	0	2	0
GP. 3.	3	2	1	0	0	3	0	0	0
GP. 4.	2	2	0	1	0	0	0	1	0
IS. 1.	2	1	0	0	0	2	0	0	0
IS. 2.	7	6	0	0	0	1	0	0	0
IS. 3.	8	7	0	0	0	1	2	0	4
IS. 4.	8	4	0	0	2	2	5	0	2
IS. 5.	11	10	0	0	0	1	7	0	5
IS. 6.	6	5	0	0	0	2	5	0	0
Total	68	48	4	7	2	21	19	5	11
%		70.59	5.88	10.29	2.94	30.88	27.94	7.35	16.18

Tabla 16 – Resumen de cobertura de herramientas propuestas para GP e IS

Con respecto a las actividades de GP, puede observarse, en la Tabla 16, la herramienta que cubre mas tareas de las 4 actividades de este proceso es la suite de google teniendo en primer lugar: Documentos, Drive, Hoja de Calculo y Draow.io, sin embargo, en esta herramienta se deben definir formatos que permitan automatizar las tareas. En este contexto, la herramienta con campos definidos y que tiene una cobertura amplia es Podio para definir estimaciones de tiempo y que ayudan a dar seguimiento al proyecto.

Con respecto a las actividades de IS, puede observarse en la Tabla 16, la herramienta que cubre mas tareas de las 6 actividades del proceso IS es nuevamente la suite de google, siendo el uso de documentos el que mas incidencia tiene. Sin embargo, el proceso de IS esta enfocado a la Implementacion y/o desarrollo del software por lo que las herramientas que mas incidencia tienen durante el desarrollo de actividades relacionadas al desarrollo son: Podio, GitLab, Jest, este ultimo da soporte para el establecimiento de pruebas que solicita la norma.

4. Estudio de caso

Tomando como base un cuestionario guía para la extracción del conocimiento de una organización, tanto para GP como para IS, se realizaron entrevistas a miembros del

centro de desarrollo del Instituto de Tecnológico de Loreto, Zacatecas, México de los tres niveles básicos: operativo, administrativo y gerencial. De esta manera se entrevistaron a dos desarrolladores y un líder técnico.

De cada entrevista se obtuvieron diagramas para GP e IS. El gestor de proyecto proporcionó sus diagramas de manera directa, cumpliendo así con la diagramación de los tres niveles básicos de una organización.

De esta manera, se obtuvieron diagramas de cada una de las entrevistas realizadas y se validaron los procesos identificados con cada entrevistado.

Tomando los diagramas como base, se analizaron y se validaron con cada una de las cuatro personas entrevistadas, para verificar que la información proporcionada concuerda con su forma de visualizar su trabajo dentro del equipo de desarrollo.

Una vez que los entrevistados aprobaron sus respectivos diagramas de forma de trabajo, se realizó una matriz para identificar las actividades en común del equipo de trabajo y visualizar aquellas actividades diferentes entre ellos. En la Figura 2, se muestra la primera parte de la matriz realizada.

	A	B	C	D	E	F	G	H	I	J	K		
1		Raul Antonio		Victor Rodríguez		Alvaro Ulises		Alberto Romo		Proceso Genérico			
2	Proceso	Tarea	Rol	Tarea	Rol	Tarea	Rol	Tarea	Rol	Tarea	Rol		
3		Realizar reunión	Cliente	Realizar reunión	Cliente	Tener primera reunión con el cliente	Cliente	Plantea necesidad	Cliente	Reunion para plantear la necesidad	Cliente		
4			Líder Técnico		Líder de proyecto				Líder de proyecto		Líder de proyecto		
5						Dar a conocer las necesidades generales	Cliente	Comunica necesidad en reunión de academia	Cliente	Comunicar la necesidad de reunión de academia	Cliente		
6									Líder de proyecto		Líder de proyecto		
7									Jefe de Carrera ISC		Jefe de carrera ISC		
8									Líder de proyecto		Líder de proyecto		
9								Elabora Enunciado de trabajo	Líder de proyecto	Elabora Enunciado de trabajo	Líder de proyecto		
10													
11													
12								Forma el equipo de trabajo	Jefe de Carrera ISC	Forma el equipo de trabajo	Jefe de Carrera ISC		
13												Líder de proyecto	Líder de proyecto
14												Líder de proyecto	Líder de proyecto
15								Expone enunciado de trabajo	Equipo de Trabajo	Expone enunciado de trabajo	Equipo de Trabajo		
16												Líder de proyecto	Líder de proyecto
17												Líder de proyecto	Líder de proyecto
18					Verificar viabilidad	Líder Técnico			Analiza el enunciado de trabajo	Líder Técnico	Analizar el enunciado de trabajo. Verificar viabilidad		
19												Líder de proyecto	Líder de proyecto
20											Equipo de Trabajo		

Figura 2 – Matriz de trazabilidad de actividades entre entrevistas.

Tomando como base la Matriz de trazabilidad de las actividades, se realizó la diagramación de los procesos GP e IS, identificados a partir de las actividades mencionadas en las entrevistas anteriores a los miembros del equipo de trabajo del centro de desarrollo de software del ITS Loreto. Una vez que se han identificado los procesos genéricos de GP e IS del centro de desarrollo del ITS Loreto, se identificaron aquellas tareas de ISO/IEC 29110 que son cubiertas por los procesos genéricos de ITS Loreto (Figura 3).

La Figura 3, muestra a manera de ejemplo la trazabilidad de las actividades identificadas hacia la norma.

Una vez realizada la trazabilidad por parte de la organización de sus procesos hacia los procesos de la norma ISO/IEC 29110, la empresa pudo identificar que herramientas le daban soporte para la realización de sus tareas orientadas a esta norma. Asimismo, esta organización detecto que un punto critico es la automatizacion de las pruebas que se estaban realizando de manera manual y con muy poca informacion para realizar acciones correctivas durante el desarrollo del software.

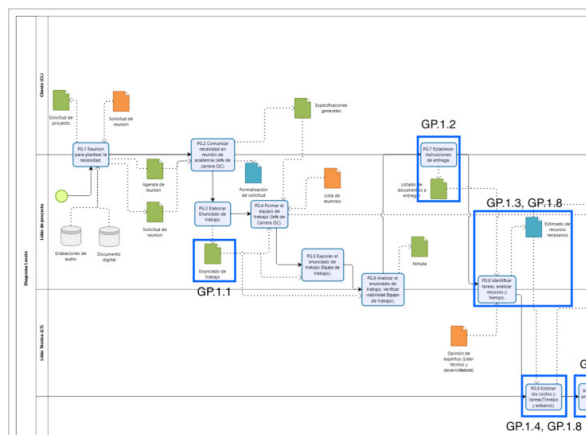


Figura 3 – Trazabilidad de proceso genérico con ISO/IEC 29110 (ejemplo).

5. Conclusiones

Gracias a la observación del estudio de caso, se logró detectar dos de las principales complicaciones que presentan las organizaciones al implementar por primera vez el estándar ISO/IEC 29110:

- La resistencia al cambio en su forma de trabajar, aun más, si una norma como la ISO/IEC 29110 no tiene un soporte de herramientas que la acompañen para la implementación de algunas actividades o tareas.
- No conocen los términos ni los entregables que requiere la ISO/IEC 29110. Al facilitar y familiarizarse con las herramientas de manera inferencial, se familiarizan con la terminología y los entregables que deben cumplir de acuerdo a la norma.
- Automatización de pruebas, un punto alarmante fue identificar que este tipo de organizaciones desconoce como establecer procesos, procedimiento y técnicas de pruebas, en este sentido, la herramienta Jest da un soporte a las pruebas que especifica esta norma.

El trabajo presentado propone una guía de herramientas que dan soporte a las tareas de los procesos de GP e IS para desarrollar los productos de Entrada, Internos y de Salida. Con base en la trazabilidad presentada entre sus procesos y los de la norma, la organización pueda identificar que herramientas pueden ser utilizadas para desarrollar los entregables que pide la norma ISO/IEC 29110 e iniciar una mejora continua.

Como trabajo futuro se esta proponiendo realizar una herramientas global que pueda gestionar la creación de formatos y poder automatizar la captura y generación de entregables de acuerdo a los productos indicados en los procesos de GP e IS de la Norma ISO/IEC 29110.

Agradecimientos

Este trabajo se pudo culminar gracias al trabajo previo y apoyo del equipo de trabajo conformado por la Dra. Mirna Muñoz, y los alumnos: Patricia Montoya-Méndez, Héctor Girón-Bobadilla, Isaac Rodríguez-Maldonado y de la participación del Centro de Desarrollo del Instituto Tecnológico Superior de Loreto (ITS Loreto), perteneciente al Ecosistema de Calidad en Tics del Estado de Zacatecas, México.

Referencias

- Chrome web store. (2020). [En línea]. Available: <https://chrome.google.com/webstore/detail/drawio-diagrams/onlkggianjhjenigcpigpjehhplldkc>.
- Git. (2020). [En línea]. Available: <https://git-scm.com/>. (Accessed: 23- Mar- 2020).
- GitLab. (2020). [En línea]. Available: <https://about.gitlab.com>.
- Google Docs. (2020). [En línea]. Available: https://www.google.com/intl/es_mx/docs/about/#start.
- Google Drive. (2020). [En línea]. Available: https://www.google.com/intl/es-419_mx/drive/.
- Google Suite. (2020). [En línea]. Available: https://gsuite.google.com.mx/intl/es-419_mx/products/sheets/.
- INEGI. (2020). Resultados de la encuesta nacional sobre productividad y competitividad de las micro, pequeñas y medianas empresas (enaproce) 2018. [Online]. Available: <https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2019/especiales/ENAPROCE2018.pdf>. (Accessed: 23- Mar- 2020).
- ISO/IEC 29110. (2012). Perfiles del ciclo de vida para las pequeñas organizaciones (PO). Parte 5-1-2: Guía de gestión e ingeniería: Grupo de perfil genérico, Perfil básico, 2012. Lima: INDECOPI.
- ISO/IEC. (2018). Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 4-1: Software engineering - Profile specifications: Generic profile group [En línea]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec:29110:-4-1:ed-2:v1:en>.
- Jest. (2020). [En línea] <https://jestjs.io/>
- NYCE. (2020). Certificación, Verificación, Normalización. [Online]. Available: <https://www.nyce.org.mx/>. (Accessed: 23- Mar- 2020).

- O'Connor, R., & Laporte, C. (2017). The Evolution of the ISO/IEC 29110 Set of Standards and Guides. *International Journal of Information Technologies and Systems Approach*, 10(1), 1-21.
- Podio. (2020). Project Management and Collaboration Software, Podio, 2020. [Online]. Available: https://podio.com/site/?gclid=CjwKCAiA7t3yBRADEiwA4GFI9YfFmI9noqzwwauyV2nj8_j-R95jENJq1-2Ee8xF4AKCIfyAYOYDxoCXrsQAvD_BwE. (Accessed: 23- Mar- 2020).
- Project Management Institute. (2013). *Guía de los Fundamentos para la Dirección de Proyectos PMBOK*, PMI, Quinta Edición. Newtown Square: Project Management Institute.

Explorando la influencia de los roles de Belbin en la especificación de requisitos de software

Raúl Aguilar Vera¹, Mirna Muñoz Mata², Julio Díaz Mendoza¹, Juan Uacán Pech¹

avera@correo.uady.mx, mirna.munoz@cimat.mx, julio.diaz@correo.uady.mx,
juan.ucan@correo.uady.mx

¹ Facultad de Matemáticas, Universidad Autónoma de Yucatán, C.P. 97000, Mérida, Yucatán, México.

² Centro de Investigación en Matemáticas, Unidad Zacatecas, C.P. 98068, Zacatecas, Zacatecas, México.

DOI: 10.17013/risti.36.34-49

Resumen: Se presenta un estudio exploratorio en el que se analiza la influencia de la Teoría de Belbin en la calidad del producto generado con el desarrollo de requisitos software. El estudio se desarrolla mediante un experimento controlado en el que el factor manipulado por el investigador se corresponde con la forma en la que los equipos de desarrollo son integrados con estudiantes. Los resultados indican que existe una diferencia estadísticamente significativa entre la calidad de los productos generados por los equipos con roles compatibles, en comparación con la calidad de los productos generados por equipo integrados en forma aleatoria.

Palabras-clave: Especificación de Requisitos Software; Experimentación en Ingeniería de Software; Roles de Belbin.

Exploring the Influence of Belbin's Roles on Software Requirements Specification.

Abstract: An exploratory study is presented in which the influence of Belbin's Theory on the quality of the product generated with the development of software requirements is analyzed. The study is developed through a controlled experiment in which the factor manipulated by the researcher is the way in which the development teams are integrated with students. The results indicate that there is a statistically significant difference in the quality of the products generated by teams with compatible roles, compared to the quality of the products generated by randomly integrated teams.

Keywords: Belbin Roles; Experimentation in Software Engineering; Software Requirements Specification.

1. Introducción

Transcurrido medio siglo de su concepción, la Ingeniería de Software (IS) ha logrado acumular un cuerpo de conocimientos que es aceptado por profesionistas y académicos

vinculados con la joven disciplina, dicho cuerpo integra un conjunto de áreas relacionadas con procesos de desarrollo —Requisitos, Diseño, Construcción, Pruebas y Mantenimiento— y con procesos de gestión —Calidad, Configuración, Proceso de Software— asociados al primer grupo (Bourque & Fairley, 2014).

A nuestros días, existe una búsqueda constante de nuevos y mejores métodos, técnicas, herramientas, así como de buenas prácticas, que le permitan a la industria del software contar con mejores prácticas para ofrecer un Proceso Software menos complejo y más eficiente (Cuevas, 2002). Por otro lado, es conocido que muchas de las tareas vinculadas con el proceso software, se realizan en el contexto de equipos de desarrollo, y también es bien sabido que los principales problemas o causas de fracaso —en los proyectos software— no son de naturaleza tecnológica, sino que se deben a factores de naturaleza sociológica (De Marco & Lister, 1999); por lo anterior, uno de los factores que ha sido estudiado en los últimos años, es el factor humano, en particular, su incidencia en la mejora del proceso software. En un estudio reciente, Morales y Vega (2018) propusieron un catálogo de factores humanos críticos para el éxito de propuestas de mejora al proceso software, uno de dichos factores se relaciona con el rol que desempeña un Ingeniero de Software al interior del equipo de desarrollo.

El presente estudio, tiene como propósito explorar si el uso de la teoría de roles de Belbin (1993) en la integración de equipos de proyecto, tiene alguna influencia en una de las principales tareas vinculadas con el proceso de desarrollo software, la especificación del software.

2. Trabajos relacionados con este estudio

Diversos investigadores (Belbin, 1981; Briggs-Myers & McCaulley, 1985; Margerison & McCan, 1985; Mumma, 2005) afirman haber identificado roles que definen el comportamiento de los integrantes al interior de un equipo de trabajo, a dichos roles se les conoce como roles de equipo, y aunque no existe evidencia de alguna relación funcional con alguna tarea o proceso particular, su ausencia o presencia, al parecer tiene una influencia significativa en los logros del equipo (Aritzeta, Swailes & Senior, 2005).

Entre los estudios sobre roles de equipo, el trabajo de Belbin (1981, 1993) es posiblemente el más utilizado entre consultores e investigadores; Belbin mantiene que un rol de equipo hace referencia a la forma de comportarse, contribuir y relacionarse con otras personas en el trabajo, y aunque algunos de los roles son naturales, otros roles pudieran ser adoptados por el propio individuo, e incluso algunos pueden llegar a ser descubiertos luego de ser adoptados.

Los roles propuestos por Belbin pueden ser agrupados en torno al tipo de conducta en tres diferentes categorías (ver tabla 1): roles orientados a la acción (A), roles orientados a las personas o roles sociales (S), y roles orientados a los procesos cognitivos o roles mentales (M).

Tipo	Rol	Cracterísticas
Acción	Impulsor (<i>Sharper: S</i>)	Retador, dinámico, trabaja bien bajo presión. Tiene iniciativa y coraje para superar obstáculos.
	Implementador (<i>Implementer: I</i>)	Disciplinado, leal, conservador y eficiente. Transforma las ideas en acciones.
	Finalizador (<i>Completer-Finisher: CF</i>)	Esmerado, concienzudo, ansioso. Busca los errores y las omisiones. Realiza las tareas en el plazo establecido.
Mental	Cerebro (<i>Plant: P</i>)	Creativo, imaginativo, poco ortodoxo. Resuelve problemas difíciles.
	Monitor Evaluador (<i>Monitor-Evaluator: ME</i>)	Serio, perspicaz y estrategia. Percibe todas las opciones. Juzga con exactitud.
	Especialista (<i>Specialist: SP</i>)	Solo le interesa una cosa a un tiempo, cumplidor del deber. Aporta cualidades y conocimientos específicos.
Social	Coordinador (<i>Chairman: CH</i>)	Maduro, seguro de sí mismo. Aclara las metas. Tiene iniciativa y coraje para superar obstáculos.
	Investigador de Recursos (<i>Resouce Investigator: RI</i>)	Extrovertido, entusiasta, comunicativo. Busca nuevas oportunidades. Desarrolla contactos.
	Cohesionador (<i>Teamworker: TW</i>)	Cooperador, apacible, perceptivo y diplomático. Escucha y evita enfrentamientos.

Tabla 1 – Roles de Equipo propuestos por Belbin

El trabajo de Belbin (1993) ofrece también un conjunto de recomendaciones que permiten integrar equipos de trabajo con base en roles compatibles, así como mecanismos para la identificación del rol primario que un individuo puede asumir en un equipo de trabajo, en función de su comportamiento.

Entre los estudios reportados sobre el uso de la teoría de Roles de Belbin en el contexto de la IS, se encuentra el trabajo de Henry y Stevens (1999) quienes exploraron la utilidad de los roles de Belbin en la mejora de la efectividad de los equipos, particularmente, la influencia del liderazgo; dichos autores llevaron a cabo un experimento controlado con estudiantes, y entre sus hallazgos concluyen que la inclusión de un solo líder en el equipo, mejora su rendimiento, en comparación con equipos que tienen varios líderes o que no incluyen dicho rol.

Pollock (2009) exploró la influencia de contar con una diversidad de roles y personalidades, como estrategia para mejorar la efectividad del equipo, dicho autor comenta que aunque no encontró evidencia de que dicha diversidad influya de alguna manera, pudo observar que la presencia de ciertos roles, como son el caso del Impulsor, Coordinador o Finalizador, pueden incrementar la efectividad.

Estrada y Peña (2013) reportan un experimento controlado con estudiantes desempeñando actividades vinculadas con las fases de requisitos, diseño y codificación; las autoras concluyen que algunos roles presentan mayor aportación en ciertas actividades, particularmente señalan al rol Implementador en la tarea de codificación.

En el caso de nuestro grupo de investigación, en un estudio previo comparamos la calidad de la legibilidad del código generado por equipos integrados con roles compatibles —

con base en la Teoría de Belbin— con equipos formados con integrantes asignados aleatoriamente; los resultados obtenidos mediante un experimento controlado con estudiantes, nos permitieron concluir que los equipos que usaron como referencia la Teoría de Belbin, presentan resultados significativamente mejores que aquellos integrados mediante un criterio de aleatoriedad (Aguileta, Ucán y Aguilar, 2017). En un segundo estudio, los autores seleccionaron la tarea de medición software, sin embargo, los resultados obtenidos entre el primer experimento y su réplica experimental —similar interna— no permitieron concluir la existencia de diferencias significativas entre las variables vinculadas ni con el producto (número de puntos de función) ni con el proceso (tiempo utilizado); no obstante, con el estudio reflexionamos sobre la alternativa de explorar la influencia de los diferentes roles de equipo en tareas individuales vinculadas con el proceso software (Aguilar, Díaz y Ucán, 2019).

3. El Proceso de Especificación de Requisitos Software

El propósito de la especificación de requisitos software, es definir el conjunto de funcionalidades, atributos de calidad y restricciones que describen en forma detallada el producto software que se desea desarrollar para un cliente u organización (Bourque y Fairley, 2014). El producto final de dicho proceso se conoce como Documento de Especificación de Requisitos Software; el estándar IEEE Std 830-1998 establece el contenido y las cualidades de un buen documento de especificación de requisitos de software (IEEE CS-SES, 1998).

El proceso de desarrollo de requisitos software consta de cuatro fases (Aguilar, *et al*, 2017). La primera etapa, denominada Elicitación de requisitos, se refiere a la identificación de las necesidades reales y las restricciones operativas expresadas por las partes interesadas de una organización; para obtener dicha información, el Ingeniero de Software diseña una estrategia en la que combina un conjunto de técnicas tradicionales, contextuales, grupales y/o cognitivas (Yousuf & Asger, 2015). La segunda etapa del proceso, el de Análisis, tiene como objetivo garantizar la calidad de los requisitos antes de incorporarlos al documento de especificación; en esta etapa, se definen los límites del sistema de software y su interacción con el entorno. Las solicitudes o requerimientos de los interesados en el nuevo sistema se traducen en requisitos de software. La etapa de Especificación consiste en documentar los requisitos de software acordados, con un nivel de detalle apropiado, y generalmente se usa un modelo de documento, el cual se redacta en términos comprensibles para las partes interesadas. Finalmente, en la última fase del proceso, la de Validación, se realiza una revisión cuidadosa de un conjunto de atributos de calidad vinculados tanto a los requisitos individuales, como al documento de manera integral; el objetivo es identificar problemas en el documento de especificación de requisitos software antes de que se pueda utilizar como base para el diseño del nuevo Sistema. En ocasiones, se utiliza la técnica de casos de uso para documentar requisitos funcionales; dicha técnica permite especificar el comportamiento de un sistema, mediante la secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Es importante destacar que los casos de uso son documentos de texto, no diagramas, y el modelado de casos de uso es principalmente un acto de redactar texto, no de dibujar; no obstante, el Lenguaje de Modelado Unificado, incorpora un tipo de diagrama que permite ilustrar para un sistema, los diferentes casos de uso, actores, así como sus relaciones.

La calidad es difícil de definir, es subjetiva y depende de los diferentes puntos de vista y del contexto sobre el constructo al que se aplica el concepto (Kitchenham & Pfleeger, 1996). En el caso del producto generado con el proceso de especificación de requisitos software, podemos utilizar el estándar IEEE 830-1998 como punto de partida para definir la calidad de los requisitos (ver Tabla 2).

Atributos de Calidad	Definición
Compleitud	Deben considerarse todos los elementos importantes que son relevantes para cumplir con las diferentes tareas del usuario. Esto incluye requisitos funcionales y no funcionales.
Consistencia	Los requisitos establecidos deben ser consistentes con todos los demás requisitos y otras restricciones importantes
Correctitud	Los requisitos implementados deben reflejar el comportamiento esperado de los usuarios y clientes.
Trazabilidad	Debería ser posible referirse al requisito de una manera fácil, tanto para los artefactos derivados del requisito como para aquellos de los que se deriva.
Inambigüedad	Los requisitos solo deben tener una interpretación posible
Verificabilidad	Debe haber un proceso para que una máquina o un ser humano verifique si el requisito se cumple o no

Tabla 2 – Ejemplos de atributos de calidad para el DER

El proceso de verificación llevado a cabo por el ingeniero de software debe realizarse en dos niveles. En un primer nivel, los atributos de calidad deben evaluarse individualmente para los requisitos de software —p.e. trazabilidad, inambigüedad, verificabilidad— y en el segundo nivel, los atributos de calidad se verifican en el documento de especificación de requisitos de manera integral —p.e. completitud, consistencia, correctitud.

4. Planificación del Experimento Controlado

Aunque en la propia definición propuesta para Ingeniería de Software por parte de la IEEE (1990) considera en su segundo inciso, el estudio —investigación— de los enfoques vinculados con el quehacer del proceso software, en el siglo pasado no se le concedió la importancia debida (Juristo y Moreno, 2001); por fortuna, a nuestros días existe una comunidad que utiliza métodos de investigación, fundamentalmente empíricos, que buscan acrecentar el cuerpo de conocimientos disponible para la disciplina (Glass, Vessey, & Ramesh, 2002).

Una de las metodologías empíricas más recurridas en el campo de la IS, es la experimentación, particularmente, la experimentación en entornos controlados (Basili, Selby & Hutchens, 1986); dicha metodología tiene como objetivo identificar las causas por las cuales se producen ciertos resultados, su aplicación nos ayuda a identificar y en su caso, comprender las posibles relaciones entre factores y variables dependientes, ambos parámetros inmersos en quehacer del proceso software. Entre los elementos característicos de los estudios controlados encontrados en la literatura, se destaca el uso de grupos de estudiantes como sujetos experimentales; Genero, Cruz-Lemus y Piattini (2014) indican al respecto, que dicha muestra académica permite al investigador obtener

evidencia preliminar para confirmar o refutar hipótesis que pueden ser contrastadas posteriormente en contextos industriales.

El presente estudio tiene como propósito explorar, mediante la ejecución de un experimento controlado con estudiantes, la influencia del uso de la Teoría de Belbin en la integración de equipos de desarrollo con integrantes que presentan roles compatibles, en la calidad del producto de una de las fases más importantes del proceso de desarrollo, el documento de especificación de requisitos software.

4.1. Factor y Tratamientos

El factor considerado en el experimento, se corresponde con la estrategia utilizada para la integración de los equipos de trabajo; se consideraron dos tratamientos:

- *Equipos Compatibles (EC)*: Equipos de trabajo de tres integrantes, conformados con roles compatibles, de acuerdo con la Teoría de Belbin.
- *Equipos Tradicionales (ET)*: Equipos de trabajo de tres integrantes, seleccionados en forma aleatoria.

4.2. Hipótesis y Variables

Con el objetivo de explorar si los equipos integrados con base en la Teoría de Roles de Belbin —a los que llamaremos Compatibles— generan Documentos de Especificación de Requisitos Software (DERS) significativamente diferentes, en términos de su calidad, de aquellos obtenidos por equipos integrados en forma aleatoria —denominados Tradicionales— se propuso un par de hipótesis:

- H₀₁: La calidad del DERS generado por los equipos tradicionales es igual a la calidad del DERS generado por los equipos compatibles.
- H₁₁: La calidad del DERS generado por los equipos tradicionales difiere de la calidad del DERS generado por los equipos compatibles

La variable dependiente, Calidad del documento ERS, se obtuvo a través de la validación —por parte de un analista experto— de cada uno de los productos generados por los equipos de desarrollo participantes.

4.3. Instrumento para Evaluación de la Calidad del DERS

Para evaluar los productos generados por los equipos de desarrollo, se diseñó un instrumento que consta de 27 ítems que utilizan una escala Likert para obtener la evaluación del DERS por parte de un analista. El instrumento diseñado consideró atributos de calidad vinculados con requisitos individuales (8 ítems) así como con el documento de especificación en su conjunto (16 ítems). Adicionalmente, se incluyen tres ítems con los que se verifica la correspondencia entre los casos de uso y los requisitos funcionales.

4.4. Diseño Experimental

El diseño experimental más apropiado para nuestro estudio, es el diseño factorial con una fuente de variación y dos tratamientos (ver tabla 3), la variable dependiente es una

métrica numérica que se obtiene a través del instrumento de evaluación de los productos generados por los equipos de desarrollo.

	Tratamientos	Equipos
Factor	Equipos Belbin	Métrica sobre la Calidad del DERS
	Equipos Tradicionales	Métrica sobre la Calidad del DERS

Tabla 3 – Diseño factorial con una fuente de variación

4.5. Propuesta del Análisis Estadístico

Para el análisis de datos, se generará una sección de estadística descriptiva con tablas de resumen de los valores obtenidos, así como gráficos de caja y bigotes para analizar visualmente el comportamiento de los mismos; en el caso del análisis inferencial, elegimos el análisis de varianza (ANOVA) de una vía, debido a que nos permite realizar pruebas de hipótesis para determinar si existe o no diferencias significativas entre las medias de los valores recogidos en la variable dependiente, para los diferentes niveles del factor (tratamientos). El ANOVA (Gutierrez y de la Vara, 2012) es una técnica que permite construir con los datos, un modelo estadístico que describe el impacto de un solo factor categórico sobre una variable dependiente.

5. Ejecución del Experimento Controlado

Con base en el plan del experimento, se trabajó con estudiantes de la Licenciatura en Ingeniería de Software que se oferta en la Universidad Autónoma de Yucatán, en particular alumnos inscritos a un curso de la asignatura Desarrollo de Requisitos de Software. A lo largo de período escolar que dura el curso, el profesor —autor principal— impartió al grupo los temas relacionados con el proceso de desarrollo de requisitos de software, y desde la conclusión de la unidad donde se abordan las técnicas de educación de requisitos, los alumnos integrados en equipos de trabajo, formados por el profesor, comenzaron a aplicar el proceso de requisitos para un problema del mundo real seleccionado por los propios estudiantes. El producto final de la asignación consistió en un DERS basado en el IEEE Std 830-1998, e incluye, como sección adicional, los casos de uso que se requieren, de acuerdo con los servicios especificados para el proyecto.

5.1. Participantes

La muestra por conveniencia utilizada para en el experimento estuvo conformada por 34 estudiantes que cursaban la asignatura denominada “Desarrollo de Requisitos de Software” que se encuentra ubicada en el quinto semestre del Plan de Estudios 2009 de la carrera antes citada. Con 32 de dichos estudiantes se integraron 8 equipos de trabajo de cuatro integrantes cada uno; para lo anterior se utilizó la información obtenida de cada estudiante —rol primario— luego de administrar el instrumento de autopercepción de Belbin (instrumento propuesto por Belbin para identificar el papel que puede desempeñar el sujeto que lo responde en un equipo de trabajo). Se integraron cuatro equipos con roles compatibles (EC), es decir, roles que no tienen conflicto entre ellos y

otros cuatro equipos como grupo de control, integrados con estudiantes asignados de forma aleatoria (Equipos tradicionales: ET).

Los dos estudiantes restantes integraron un noveno equipo para efectos de la asignatura, sin embargo, no fueron incluidos en el estudio en virtud del número diferenciado de estudiantes. Por otro lado, por motivos personales, uno de los estudiantes que integraba a uno de los EC (equipo 3) abandonó la asignatura cuando estaban trabajando en la primera fase del proceso de requisitos, por ello fue necesario hacer un ajuste en el equipo, y también fue descartado para el estudio. La tabla 4 ilustra la asignación de los equipos a los tratamientos.

Tratamientos	Equipos
EC	I, II, IV
ET	V, VI, VII, VIII

Tabla 4 – Asignación de sujetos experimentales por tratamiento

5.2. Sujetos Experimentales

Dado que las mediciones se obtendrían de los productos generados por los equipos de desarrollo (DERS), los sujetos experimentales en este caso fueron finalmente los siete equipos de trabajo integrados por el profesor. La conformación de los tres equipos basados en la teoría de Belbin se ilustra en la Figura 1; las letras dentro de los círculos indican el identificador utilizado para los roles de equipo desempeñados por cada participante (consultar tabla 1), y así se puede observar la conformación de los tres equipos compatibles (EC). El número romano al centro los cuatro círculos (que representan a los integrantes) es un identificador para el equipo –sujeto experimental.

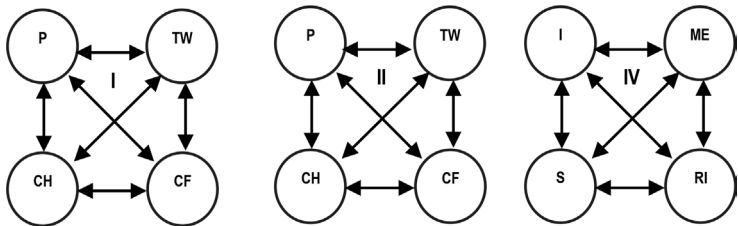


Figura 1 – Equipos integrados con la Teoría de Belbin (Equipos Compatibles: EC)

Durante el desarrollo del curso, los equipos realizaron presentaciones parciales del avance de su proyecto, no obstante, la entrega del producto por evaluar en la asignatura, se programó para el final del curso; dichos DERS generados por los equipos, son los que se usaron posteriormente para evaluar y obtener las métricas de calidad que serían utilizadas como variable respuesta en el experimento controlado; es importante destacar que la evaluación de los productos, para efectos de evaluación en el curso, fue independiente de la evaluación realizada a través del instrumento diseñado para nuestro experimento.

5.3. Proceso de generación de los Objetos Experimentales

El experimento se realizó en cuatro momentos diferentes; en una primera sesión, se administró el instrumento de autopercepción para identificar los roles primarios que podrían desempeñar los participantes; en un segundo momento, el maestro trabajó con los estudiantes a lo largo del curso sobre los temas de la asignatura, y a partir dela tercera unidad los siete equipos desarrollaron las actividades del proceso de requisitos. Posteriormente, los equipos entregaron el DERS —objetos experimentales. Finalmente, con los DERS generados por los equipos, se invitó a dos analistas expertos a evaluar la calidad de los productos generados, utilizando para ello el instrumento citado en la sección 4.3.

La tabla 5 resume las métricas obtenidas con doble evaluación.

Equipo	Evaluación 1	Evaluación 2
I	106	107
II	119	126
IV	121	106
V	99	95
VI	108	89
VII	107	105
VIII	91	89

Tabla 5 – Asignación de sujetos experimentales por tratamiento

6. Análisis de Datos del Experimento

En esta sección se presenta el análisis estadístico de los dos conjuntos de métricas obtenidas por los expertos invitados al experimento.

6.1. Análisis Estadístico de la Evaluación 1

Esta sección presenta tanto el análisis estadístico descriptivo como el análisis estadístico inferencial de las mediciones generadas por el evaluador 1. La Tabla 6 presenta algunas de las medidas más importantes de tendencia central y variabilidad.

Factor	#	μ	Mediana	σ^2	Min	Max	Rango
EC	3	115.333	119.0	66.3333	106.0	121.0	15.0
ET	4	101.25	103.0	62.9167	91.0	108.0	17.0

Tabla 6 – Resumen Estadístico de la Evaluación 1

Podemos observar que tanto la media como la mediana de los EC es mayor que la del ET, y que los EC presentan un menor rango; no obstante, los EC presentan una mayor varianza.

Para comparar visualmente los dos tratamientos, generamos un diagrama de caja y bigotes, dicho gráfico podemos nos permite observar la dispersión y la simetría de ambos conjuntos de datos (ver Fig. 2).

En la Figura 2 podemos ver que presentan ligeras superposiciones en los tratamientos, por lo que es probable que presenten diferencias significativas entre ellos. También podemos observar que no existen grandes diferencias en la dispersión de los tratamientos, aunque los ET tienen mayor simetría. Con el propósito de evaluar si las diferencias observadas en el análisis descriptivo previo, resultaban significativas, se plantearon las siguientes hipótesis estadísticas en torno a la Calidad del DERS.

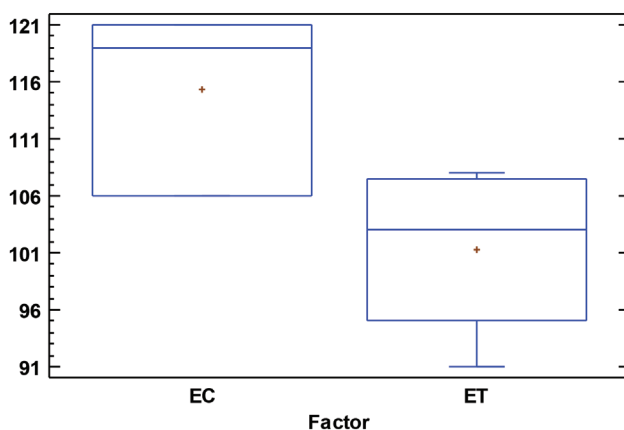


Figura 2 – Diagrama de Caja y bigotes para los tratamientos de la Métrica Calidad del DERS (Evaluación 1)

$$H_{0_{C-DERS}}: \mu_{EC} = \mu_{ET} ; H_{1_{C-DERS}}: \mu_{EC} <> \mu_{ET} \quad (I)$$

El resultado de evaluar con ANOVA dichas hipótesis estadísticas, se ilustra en la tabla 7.

Métrica	F	p-valor
C-DERS	5.29	0.0698

*p-valor < 0.10

Tabla 7 – ANOVA para la variable Calidad del DERS (Evaluación 1)

La tabla ANOVA descompone la varianza de la variable en estudio en dos componentes: un componente intergrupo y un componente en grupos; dado que el valor p de la prueba F —en la variable Calidad del DERS— es menor que 0.1 (0.0698) la hipótesis de nulidad puede ser rechazada, es decir, la hipótesis alternativa permite afirmar que existe diferencias entre las medias de los dos tratamientos con un 10% de significancia.

Es importante mencionar que el modelo ANOVA tiene asociado tres supuestos que es necesario validar antes de utilizar la información que nos ofrece; los supuestos del modelo son: (1) Los errores experimentales de sus datos se distribuyen normalmente,

(2) existe independencia entre las muestras, y (3) No existe diferencia entre la varianza de los tratamientos (Homocedasticidad)

Para validar el primer supuesto, usaremos el gráfico de probabilidad normal de los residuos. Es una técnica gráfica para evaluar si un conjunto de datos se distribuye de acuerdo con la distribución normal. Como se puede ver en la gráfica que se ilustra en la figura 3a, los puntos no muestran desviaciones de la diagonal, por lo que es posible asumir que los residuos tienen una distribución normal.

En cuanto al supuesto de independencia de los datos, generamos un gráfico de residuales versus secuencia; en este caso en la figura 3b podemos observar que no existe una tendencia en los datos, por lo que es posible suponer que los datos provienen de poblaciones independientes.

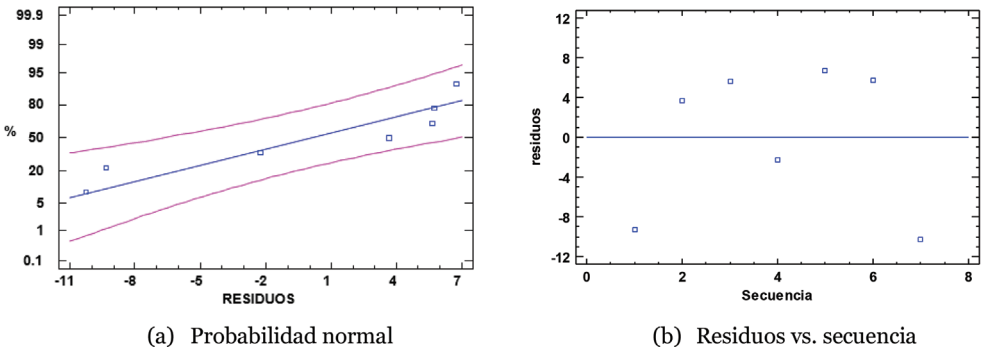


Figura 3 – Gráficos generados con los residuos (Evaluación 1)

Finalmente, la prueba de Levene permite evaluar diferencias significativas entre las varianzas de los dos conjuntos de datos (homocedasticidad) y puesto que el p valor es mayor que 0.05 (ver tabla 8), no existe una diferencia estadísticamente significativa entre las desviaciones estándar, con un nivel del 95.0% de confianza.

Prueba	p-valor
0.000134924	0.9912

Tabla 8 – Prueba de Levene (Evaluación 1)

6.2.Análisis Estadístico de la Evaluación 2

De manera similar a la sección 6.1, en esta sección se presenta el análisis estadístico descriptivo e inferencial de las mediciones generadas por el evaluador 2. La Tabla 9 presenta el resumen estadístico de la evaluación 2, podemos identificar que tanto la media como la mediana de los EC es mayor que la de los ET, aunque el rango en este caso es mayor que el de los ET, así mismo los EC presentan mayor variabilidad en los datos.

Factor	#	μ	Mediana	σ^2	Min	Max	Rango
EC	3	113.0	107.0	127.0	106.0	126.0	20.0
ET	4	94.5	92.0	57.0	89.0	105.0	16.0

Tabla 9 – Resumen Estadístico de la Evaluación 2

En la Figura 4 podemos ver que los conjuntos de datos se encuentran desfasados, por lo que es muy probable que presenten diferencias significativas entre ellos. También podemos observar que tanto los EC como los ET presentan asimetría a su derecha, aunque es ligeramente menor la de los ET.

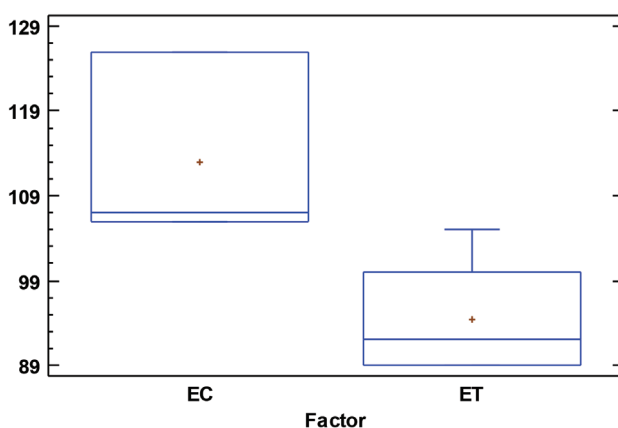


Figura 4 – Diagrama de Caja y bigotes para los tratamientos de la Métrica Calidad del DERS (Evaluación 2)

Para evaluar si las diferencias observadas previamente resultan significativas, se plantearon las siguientes hipótesis estadísticas en torno a la Calidad del DERS.

$$H_{0_{C-DERS}}: \mu_{EC} = \mu_{ET} ; H_{1_{C-DERS}}: \mu_{EC} < \mu_{ET} \quad (II)$$

El resultado de evaluar con ANOVA dichas hipótesis estadísticas, se ilustra en la tabla 10.

Métrica	F	p-valor
C-DERS	6.90	0.0467

*p-valor < 0.05

Tabla 10 – ANOVA para la variable Calidad del DERS (Evaluación 2)

Dado que el valor p de la prueba F —en la variable Calidad del DERS— es menor que 0.05 (0.0467) la hipótesis de nulidad puede ser rechazada, es decir, la hipótesis alternativa permite afirmar que existe diferencias entre las medias de los dos tratamientos incluso con un 5% de significancia. Al igual que con la evaluación 1, procedimos a evaluar los supuestos del modelo ANOVA en forma similar a la sección previa.

En el caso del primer supuesto (normalidad) el gráfico de probabilidad normal (ver figura 5a) nos indica que los residuos se presentan dentro del intervalo de confianza de la diagonal, por lo que es posible asumir que los residuos presentan una distribución normal.

En cuanto al supuesto de independencia de los datos, en el gráfico de residuales versus secuencia (figura 5b) no se identifica tendencia alguna en los datos, por lo que es posible suponer que los datos provienen de poblaciones independientes.

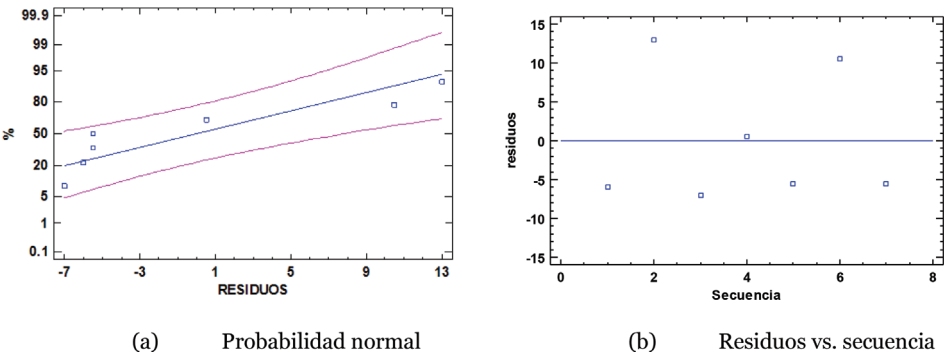


Figura 5 – Gráficos generados con los residuos (Evaluación 2)

Por su parte, con la prueba de Levene podemos concluir que no existe una diferencia estadísticamente significativa entre las desviaciones estándar de ambas poblaciones — con un nivel del 95.0% de confianza— lo anterior debido a que el p valor es mayor que 0.05 (ver tabla 11).

Prueba	p-valor
1.09262	0.3438

Tabla 11 – prueba de Levene (Evaluación 2)

7. Conclusiones y Trabajos en Curso

El proceso de requisitos busca satisfacer las necesidades de las partes interesadas mediante la especificación de las funcionalidades de una aplicación software, para lo cual el Ingeniero de Software interactúa a través de las técnicas de obtención con las partes interesadas de manera individual o grupal, y el proceso de análisis y especificación de requisitos se realiza con varias iteraciones.

En este artículo presentamos un experimento controlado en el que comparamos la calidad de un producto generado por los equipos de trabajo en tareas relacionadas con el desarrollo de software, particularmente, con la tarea del desarrollo de requisitos software. Los tratamientos a comparar se vincularon con la forma de integrar los equipos de trabajo; en primer lugar, la propuesta de integrar equipos utilizando la teoría de roles de Belbin (Equipos compatibles: RC) y en segundo lugar, la forma tradicional de asignar aleatoriamente a sus miembros (equipos tradicionales: ET).

Para llevar a cabo el experimento controlado, utilizamos un diseño de comparación de medias, y para el análisis estadístico de los datos, utilizamos el modelo ANOVA de una vía, el cual sirve para comparar dos medias de dos grupos independientes (no relacionados) que utilizan la distribución F. Con el propósito de obtener una doble evaluación, el experimento incorporó la evaluación de los productos por parte de dos expertos en requisitos y se procedió al análisis independiente de los datos. Los resultados sobre la calidad del DERS mostraron diferencias significativas entre los tratamientos, con mejores resultados para los EC; en el caso de la evaluación 1 las diferencias pueden ser explicadas con un 10% de significancia, y en la evaluación 2 con un 0.05 %.

Los obtenidos en este experimento coinciden con lo encontrado y descrito en (Aguileta, Ucán y Aguilar, 2017), en ambos casos el producto fue desarrollado a lo largo de un período aproximado de tres meses, plazo en el que posiblemente el equipo logro evolucionar de un simple grupo a un equipo efectivo de trabajo; no obstante, el resultado contrasta con lo reportado en (Aguilar, Díaz y Ucán, 2019), y es que posiblemente la tarea de medir el software, no requiere el desarrollo de altos niveles de interacción entre los miembros del equipo, tampoco un período largo de trabajo, pero sobre todo, es probable que deba ser realizada en forma individual.

Es importante mencionar que cuando usamos la experimentación como una metodología empírica, dos de las circunstancias que encontramos son las relacionadas con: (1) el tamaño de muestra no grande, (2) la necesidad de un muestreo conveniente; Sin embargo, lo ideal en las estadísticas para tener una muestra representativa de la población sería: (1) usar un tipo de muestreo apropiado y (2) seleccionar un tamaño de muestra representativo, es prácticamente imposible realizarlo en nuestro campo. Por otro lado, el desarrollo de una serie de réplicas experimentales, o en su caso, la obtención de datos para la variable dependiente por procesos alternos, como fue nuestro caso de utilizar la validación por parte de más de un experto, nos permite corroborar las hipótesis de investigación y así obtener un conocimiento más específico sobre el fenómeno.

Como parte de los trabajos en curso, el grupo de investigación ha realizado experimentos controlados en tareas como el Diseño de Bases de Datos, Desarrollo de Aplicaciones Web, y solo nos resta por realizar la validación de los productos, obtención de las métricas correspondiente y el análisis estadístico de los datos.

Referencias

- Aguilar, R., Oktaba, H., Juarez, R., Aguilar, J., Fernández, C., Rodríguez, O., & Ucán, J. (2017). Ingeniería de Software. En: Pineda, L. (ed.) *La computación en México por especialidades académicas*. Academia Mexicana de Computación, Capítulo V. Mexico: Academia Mexicana de Computación.

- Aguilar, R., Díaz, J. & Ucán, J. (2019). Influencia de la Teoría de Roles de Belbin en la Medición de Software: Un estudio exploratorio. *RISTI - Revista Ibérica de Sistemas y Tecnologías de la Información*, (31), 50-65.
- Aguileta, A. Ucán, J. & Aguilar, R. (2017). Explorando la influencia de los roles de Belbin en la calidad del código generado por estudiantes en un curso de ingeniería de software. *Revista Educación en Ingeniería*, 12(23), 93-100.
- Aritzeta, A., Swailes, S., & Senior, B. (2005). *Team roles: Psychometric evidence, construct validity and team building*. Hull, UK: University Hull.
- Basili, V., Selby, R., & Hutchens, D. (1986). Experimentation in Software Engineering. *IEEE Transactions on Software Engineering*, 12(7), 733-743.
- Belbin, M. (1981). *Management teams: Why they succeed or fail*. New York, USA: John Wiley & Sons.
- Belbin, R.M. (1993). *Team roles at Work*. Oxford: Elsevier Butterworth Heinemann.
- Bourque, P., & Fairley, R. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK V3.0)*. IEEE Computer Society.
- Briggs-Myers, I., & McCaulley, M. (1985). *Manual, A Guide to the Development and Use of the Myers-Briggs Type Indicator*. Palo Alto, USA: Consulting Psychologists Press.
- Cuevas, G. (2002). *Gestión del Proceso Software*. Madrid, España: Centro de Estudios Ramón Areces.
- De Marco, T., & Lister, T. (1999). *Peopleware Productive Projects and Teams*. 2nd Ed. New York, USA: Dorset House Publishing Co.
- Estrada, E., & Peña, A. (2013). Influencia de los roles de equipo en las actividades del desarrollador de software. *Revista Electrónica de Computación, Informática, Biomédica y Electrónica*, 2(1), 1-19.
- Genero, M., Cruz-Lemus, J., & Piattini, M. (2014). *Métodos de investigación en ingeniería de software*. Madrid, España: Ra-Ma.
- Glass, R., Vessey, I., & Ramesh, V. (2002). Research in Software Engineering: An analysis of the literature. *Information and Software Technology*, 44(8), 491-506.
- Gutiérrez, H., & De la Vara, R. (2012). *Análisis y Diseño de Experimentos*. 3ª Ed. Ciudad de México: México. McGraw Hill.
- IEEE (1990). *IEEE Standard Glossary of Software Engineering Terminology* (Std. 610.12-1990). IEE.
- IEEE CS-SES (1998). *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Std 830-1998. IEE.
- Henry, S., & Stevens, K. (1999). Using Belbin's leadership role to improve team effectiveness: An empirical investigation. *The Journal of Systems and Software*, 44, 241-250.

- Juristo, N., & Moreno, A. (2001). *Basics of Software Engineering Experimentation*. Boston, USA: Kluwer Academic Publishers.
- Kitchenham, B., & Pfleeger, S. (1996). *Software quality: the elusive target*. IEEE Software, 13(1), pp. 12-21.
- Margerison, C.J., & McCann, D.J. (1985). Team Management Profiles: Their use in Managerial Development. *Journal of Management Development*, 4(2), 34-37. DOI: 10.1108/eb051580
- Morales, N., & Vega, V. (2018). Factores Humanos y la Mejora de Procesos de Software: Propuesta inicial de un catálogo que guíe su gestión. *RISTI - Revista Ibérica de Sistemas y Tecnologías de Información*, (29), 30-42.
- Mumma, F.S. (2005). *Team-work & team-roles: what makes your team tick?: facilitator guide*. King of Prussia, PA: HRDQ Press.
- Pollock, M. (2009). Investigating the relationship between team role diversity and team performance in information systems teams. *Journal of Information Technology Management*, 20(1), 42-55.
- Senior, B. (1997). Team roles & Team performance: Is there 'really' a link?. *Journal of Occupational and Organizational Psychology*, (70), 85-94.
- Yousuf, M., Asger, M. (2015). Comparison of various requirements elicitation techniques. *Int. J. Comput. Appl*, 116(4): 0.

Plataforma de análisis de datos para la evaluación de desempeño de software

Silvana de Gyves Avila¹, Patricia Ortegon Cano¹, Ayrton Mondragon Mejia¹,
Ismael Solis Moreno¹, Arianne Navarro Lepe¹, Gloria Eva Zagal Dominguez¹

silvana.degyves@ibm.com, patricia.ortegon@ibm.com, ayrton.mondragon1@ibm.com,
ismael@mx1.ibm.com, arianne.navarro@ibm.com, gloria.eva.zagal@ibm.com

¹IBM, México Software Lab, Carretera al Castillo 2200, 45680, el Salto Jalisco, México

DOI: 10.17013/risti.36.50-64

Resumen: El desempeño es un parámetro importante en los procesos de evaluación de software. Es usado como punto de diferenciación entre competidores. El aseguramiento del desempeño no es trivial, ya que requiere la ejecución de pruebas exhaustivas y el análisis de grandes volúmenes de datos. Normalmente, se usan soluciones comerciales para producir métricas de desempeño. Sin embargo, estas son de propósito general y requieren de esfuerzo para ser adaptadas a casos particulares. En este artículo presentamos DDP, una plataforma para analizar datos de pruebas de desempeño. DDP utiliza tecnología de Big Data para recolectar, almacenar, procesar y analizar resultados de desempeño de una manera integrada. Demostramos el uso exitoso de DDP en la evaluación de “Spectrum Scale”, una solución de almacenamiento definido por software. Ilustramos el uso DDP en el análisis de pruebas de regresión para verificar y validar la calidad de las nuevas versiones creadas durante el proceso de desarrollo.

Palabras-clave: calidad de software; desempeño; big data; ciencia de datos.

A Data Analysis Platform to Evaluate Performance During Software Development Process

Abstract: Performance is one of the most important parameters to consider during the software development process. It is used as competitive advantage among similar solutions. Assuring expected performance levels is not trivial since it requires to run exhaustive tests and the analysis of Big Data. Normally, companies rely on commercial solutions to produce performance analytics. However, these require a significant effort to be adapted to particular performance use cases. In this paper, we describe DDP, a platform to analyze performance testing data. DDP uses Big Data technology to collect, store, process and analyze performance results in an integrated way. We demonstrated the successful use of DDP evaluating the performance of “Spectrum Scale”, a software defined storage solution. We illustrate the use of DDP analyzing data from performance regression tests to verify and validate the quality of new versions build during the development process.

Keywords: software quality; performance; big data; data science.

1. Introducción

El desempeño se considera como una de las dimensiones no-funcionales más importantes en cualquier proceso de desarrollo de software (Kurian, 2018). Mantener o mejorar los niveles de desempeño, como ventaja competitiva, es fundamental para lograr la satisfacción del cliente y los objetivos del mercado. Sin embargo, garantizar que un componente de software ofrezca el desempeño esperado se ha convertido en una tarea compleja y desafiante, junto con la complejidad de los entornos en los que está destinado a ejecutarse (Aleti, 2018). Los sistemas ciber-físicos han evolucionados de servidores stand-alone a entornos distribuidos masivos, compuestos por varios clústeres y cargas de trabajo complejas, que interactúan entre sí. Para que las compañías de software evalúen un subconjunto representativo de escenarios realistas, se requiere de herramientas avanzadas que no solo automaticen la ejecución de las pruebas de desempeño, sino que además manejen grandes volúmenes de datos producidos durante el proceso de evaluación (Kurian, 2018). Los datos de desempeño son complejos y provienen de diversas fuentes, por ejemplo: logs, benchmarks y archivos de configuración, cada uno con distintos niveles de estructura. Por lo tanto, analizar estos datos y producir observaciones, requiere de un conjunto de tecnologías y esfuerzo significativo, para llevar los datos de desempeño a través de su ciclo de vida. Comúnmente, se utilizan paquetes estadísticos como, R-Studio (Dessau, 2008), SPSS (Aldrich, 2018) y Cognos Analytics (Huijgens, 2018). Sin embargo, las etapas que se encuentran entre la recolección, almacenamiento, pre-procesamiento y visualización de datos, se llevan a cabo en su mayoría de forma manual, creando silos donde se desperdician datos, y en muchas ocasiones, se pierden. Esto crea la necesidad de confiar en un marco holístico que pueda ayudar a los analistas de desempeño a simplificar la gestión de datos y la visualización de resultados. En este artículo, se introduce la Plataforma de Desempeño Orientada a Datos (Data Driven Performance) DDP, un enfoque que permite integrar la tecnología necesaria para manejar datos provenientes de las pruebas de desempeño realizadas para soluciones de almacenamiento definido por software. DDP está compuesto por ETLs y motores de procesamiento de datos que trabajan con un repositorio de datos altamente escalable e interfaces de usuario interactivas. DDP ha sido implementado para mejorar el proceso de desarrollo de software de Spectrum Scale, un sistema de gestión de archivos de alto rendimiento que se utiliza ampliamente en múltiples compañías a nivel mundial. DDP utiliza Big Data para generar reportes y dashboards sobre las tendencias de desempeño, y así proporcionar soporte visual en la toma de decisiones durante los procesos de desarrollo y pruebas. Nuestro caso de prueba de éxito muestra la flexibilidad de DDP para manejar una amplia variedad de datos de pruebas de desempeño. DDP complementa exitosamente los procesos de desarrollo y pruebas, para así mejorar la calidad del producto ofrecida a los clientes.

La mayor contribución de este artículo es la descripción de DDP, una plataforma para mejorar los procesos de pruebas de desarrollo. También se presenta un caso de uso donde la aplicabilidad de DDP ha sido demostrada y se discuten los principales analíticos implementados. El resto de este artículo se encuentra organizado de la siguiente forma: la sección 2 presenta el marco conceptual, la sección 3 describe la plataforma de desempeño orientada a datos, la sección 4 describe el caso de uso, la sección 5 presenta los analíticos implementados, y la sección 6 incluye nuestras conclusiones y trabajo futuro.

2. Marco Conceptual

2.1. Dashboard, Estadísticos de Desempeño y Benchmark

A continuación, se presentan los conceptos de dashboard, estadísticos de desempeño y benchmark, los cuales serán utilizados a lo largo de este artículo.

- **Dashboard.** Puede definirse como “una herramienta visual e interactiva que muestra en una sola pantalla la información más importante necesaria para lograr uno o más objetivos individuales y/u organizacionales, permitiendo al usuario identificar, explorar y comunicar áreas que necesiten medidas correctivas” (Yigitbasioglu & Velcu, 2012). En el contexto de desempeño, los dashboards han sido ampliamente utilizados por diversas organizaciones y poco a poco han ido reemplazando a los reportes tradicionales (Velcu-Laitinen & Yigitbasioglu, 2012).
- **Estadísticos de desempeño.** Para identificar y analizar un problema de desempeño, es necesario contar con datos estadísticos, los cuales se obtienen a partir de información histórica. Un estadístico de desempeño se define en base al sistema que está siendo evaluado y las dimensiones de interés. Los estadísticos de desempeño pueden incluir elementos de estadística descriptiva y estadística predictiva.
- **Benchmark.** Es un estándar que se utiliza para evaluar el desempeño relativo de un sistema. Es relativo, ya que los resultados de una prueba generalmente cubren un solo aspecto o dimensión. Existen benchmarks orientados a evaluar velocidad de transferencia, procesamiento, lectura y escritura, etc. (Price, 1989).

2.2. IBM Spectrum Scale

Las características de desempeño de un sistema de software, tales como el tiempo de respuesta, rendimiento de la red, escalabilidad y disponibilidad, son atributos de calidad que se vuelven clave en las aplicaciones distribuidas usadas para realizar el análisis de grandes volúmenes de datos. A fin de cumplir con todos estos requerimientos, la forma en que realizamos el almacenamiento de datos se vuelve un factor crítico (Denaro, 2015).

IBM tiene un portafolio robusto de soluciones de almacenamiento definido por software, y uno de los componentes clave de este portafolio es *Spectrum Scale*, un sistema de archivos de clúster que provee acceso concurrente a un solo sistema de archivos o a un conjunto de ellos, desde múltiples nodos (Coyne, 2018). Esto permite acceso de alto desempeño al conjunto común de datos para mantener una solución escalable o proveer una plataforma de alta disponibilidad. *Spectrum Scale* es usado en muchas de las supercomputadoras científicas más grandes del mundo y en aplicaciones comerciales que requieren acceso de alta velocidad a grandes volúmenes de datos (Quintero, 2015).

Cuando una nueva aplicación paralela es desarrollada, una etapa de calibración y ajuste es requerida para evaluar su comportamiento, con el propósito de mejorar el desempeño. Algunos cuellos de botella en el código paralelo solamente pueden ser descubiertos durante el modo de ejecución. Durante este proceso de evaluación se utilizan las métricas tradicionales de desempeño para sistemas distribuidos que son: latencia, descripción del retardo entre una solicitud y la terminación de una operación; rendimiento de la

red, que denota el número de operaciones que pueden ser terminadas en un periodo de tiempo dado; y escalabilidad, que identifica la dependencia entre el número de recursos de sistemas distribuidos que pueden ser usados por una aplicación distribuida. Otras métricas relevantes son actividades de I/O y tiempo de respuesta (Denaro, 2015).

2.3. Aplicaciones Orientadas a Datos en el Desarrollo de Software

En los últimos años, el análisis de datos ha sido aplicado a diferentes campos, desde la ciencia hasta los negocios, y la ingeniería de software no es la excepción. Como podemos ver en diferentes trabajos (Menzies, 2013; Mockus, 2014), las metodologías orientadas a datos han sido aplicadas a todas las fases en el ciclo de desarrollo de software. Y el rol del científico de datos en un equipo de desarrollo de software comienza a ser más popular (Kim, 2016).

Las compañías de software están impulsando un enfoque incesante en la calidad y las pruebas de software, podemos decir que estas áreas están enfrentando un proceso de disrupción en estos momentos. Las tareas más comunes y repetitivas están siendo reemplazadas por frameworks automatizados de pruebas más frecuentemente, y nuevos frameworks de automatización están siendo creados (Yandrapally, 2014).

Un enfoque orientado a los datos en el campo de las pruebas funciona de la siguiente manera: las entradas para las pruebas son almacenadas en una base de datos, un controlador de pruebas toma múltiples conjuntos de datos de entrada y ejecuta los casos de prueba. Cuando las entradas son agregadas o removidas de la base de datos, las pruebas no son alteradas. Una revisión de diferentes trabajos que siguen este enfoque, principalmente orientado a las pruebas de requerimientos funcionales puede encontrarse en (Anbunathan, 2015).

Cuando estamos verificando los requerimientos funcionales de cualquier aplicación, frecuentemente los problemas se pueden rastrear hacia el código o el diseño. Pero cuando se trata de los requerimientos no funcionales, donde el desempeño juega un rol principal, encontrar el problema puede ser más complejo debido a que se involucran diferentes aspectos fuera de la implementación del software. Por ejemplo, la red y las configuraciones de hardware que requieren la participación de diferentes expertos en dichas áreas.

Durante el ciclo de desarrollo, las variaciones en el desempeño pueden ocurrir en muchas ocasiones, pero las pruebas de desempeño son la herramienta para evitar el impacto que dichas variaciones negativas podrían tener en el siguiente entregable del producto de software (Mostafa, 2017). Los desarrolladores requieren detectar regresiones en el desempeño tan pronto como sea posible, para reducir el impacto negativo y el costo que conlleva arreglar dichos problemas. Entonces, existe una necesidad para integrar este tipo de pruebas al ciclo de desarrollo y no dejarlas hasta el final como se realiza comúnmente.

La evaluación del desempeño es usualmente cara de realizar debido a que requiere ejecuciones frecuentes que pueden llevar horas o incluso días. Además, la cantidad de métricas y datos que se generan frecuentemente son complejos y se producen en grandes volúmenes.

La mayoría del trabajo reciente que podemos encontrar relacionado con la automatización de pruebas de desempeño está enfocado en el análisis de los datos recolectados. En (Shang, 2015), se propone un enfoque basado en modelos para automatizar el análisis de las métricas de desempeño, construyendo modelos de regresión para encontrar retrocesos en el desempeño del software. En (Foo, 2015), se presenta otro trabajo basado en el impacto de los diferentes ambientes de ejecución donde las pruebas de desempeño son realizadas, usando modelos estadísticos. En (Berral, 2017), una propuesta de aprendizaje automático es usada para interpretar benchmarks de datos de desempeño en aplicaciones de Big Data.

También podemos encontrar algunos trabajos que tienen como objetivo otros pasos en el proceso de las pruebas de desempeño. En (Mostafa, 2017), se implementa un método para priorizar los casos de pruebas de desempeño, a fin de permitir más ejecuciones en etapas tempranas del desarrollo. En (Kroß, 2016), se presenta una propuesta de arquitectura de DevOps para pruebas de desempeño. En (Okanović, 2019), se presenta una herramienta para crear reportes de análisis de desempeño basados en diferentes intereses.

Recolectar, analizar, modelar y visualizar datos son pasos dentro del proceso de evaluación de desempeño. Sin embargo, ninguna de las soluciones existentes provee la integración de todas las fases de dicho proceso. Herramientas que permitan el análisis de todos estos datos y los presenten a los diferentes tomadores de decisiones, de una manera sencilla son una necesidad primordial.

Las diferentes propuestas descritas en esta sección se enfocan en uno o dos de los pasos del proceso, mientras los pasos restantes son cubiertos usando algunas herramientas comerciales que no siempre son ad-hoc para el sistema que se está probando, o en otros casos, estos pasos son ejecutados de manera manual. Integrar todas las diferentes soluciones existentes puede ser muy complejo y llevar a la pérdida de datos. Por lo tanto, aún existe una necesidad de herramientas que permitan integrar todas las etapas del proceso de evaluación de desempeño en un solo marco de trabajo. En las siguientes secciones describimos DDP, una plataforma para analizar y explotar datos de desempeño en todas las fases de evaluación.

3. Plataforma DDP

La plataforma DDP proporciona a los usuarios un repositorio centralizado que les permite almacenar, organizar y visualizar resultados de desempeño, hacia y desde una sola fuente. Esto ayuda a resolver el problema que se genera cuando se consumen datos de desempeño (tales como: resultados de benchmarks, datos de cargas de trabajo, perfiles de sistemas, configuración de hardware y logs, entre otros), que tienen diversos orígenes y formatos, y que no se encuentran relacionados de forma explícita.

Los componentes que integran la arquitectura de la plataforma de DDP se muestran en la figura 1 y se describen a continuación:

- **ETL (Extract, Transform and Load).** Módulo que se encarga de la lectura, procesamiento y almacenamiento de información relevante obtenida a partir de diversos archivos que incluyen resultados de pruebas de desempeño. Cada

archivo incluye información que debe ser filtrada y estructurada, antes de ser almacenada en el repositorio. Esto facilita el análisis numérico y textual que realizan los otros módulos de la plataforma. Este módulo se encuentra implementado en Python. Para cada uno de los diferentes benchmarks que se ejecutan durante una evaluación de desempeño, existe un analizador que utiliza expresiones regulares (importadas de la librería re de Python), para extraer toda la información relevante. Los valores obtenidos se procesan y almacenan en las tablas correspondientes definidas utilizando modelos de Cassandra.

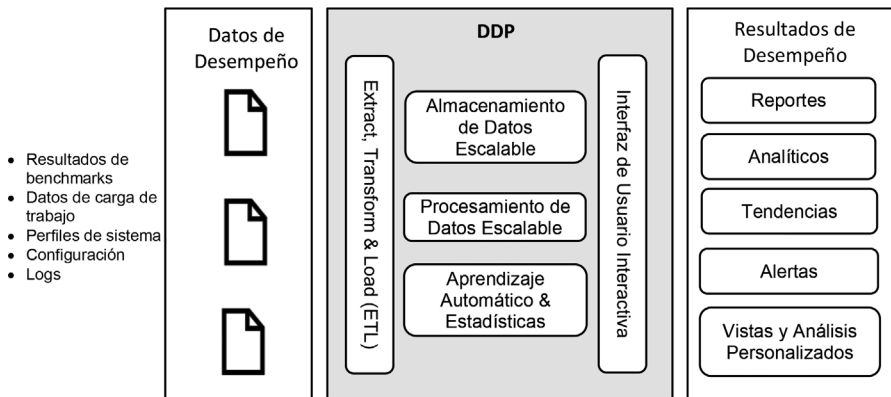


Figura 1 – Arquitectura de la Plataforma DDP

- **Almacenamiento de Datos Escalable.** Repositorio en el cual se almacenan los diferentes datos de desempeño. Este módulo se encuentra formado por dos bases de datos NoSQL: Cassandra (The Apache Software Foundation, 2016) y MongoDB (MongoDB, Inc., 2020). Los datos estructurados se almacenan en las tablas de Cassandra, mientras que los archivos que contienen los datos de desempeño (con el detalle de cada prueba), se almacenan como documentos en MongoDB. La información estructurada se utiliza para la etapa de análisis, mientras que los documentos se utilizan para consulta. En algunos casos, puede ser necesario que un experto humano realice una revisión de detalles contenidos en esos documentos, esto con fines de rastreo de procedencia.
- **Procesamiento de Datos Escalable.** Este módulo procesa los datos de desempeño que se encuentran almacenados en las tablas de Cassandra. El procesamiento de los datos se ejecuta utilizando trabajos de Scala (École Polytechnique Fédérale, 2002-2020) y un clúster de Spark (The Apache Software Foundation, 2018). Para evitar tiempos de respuesta elevados al momento de consultar gráficos y reportes (comparativas de desempeño), todos los resultados obtenidos por el módulo de procesamiento también son almacenados en estructuras de Cassandra.
- **Aprendizaje Automático & Estadísticas.** Conjunto de algoritmos y heurísticas utilizados para determinar regresiones de desempeño, perfiles de configuración y toda la información a desplegar en la interfaz de usuario.

Este módulo se implementó en Python y debido a la naturaleza, no es posible proporcionar más detalles, ya que se encuentra bajo un acuerdo de confidencialidad.

- **Interfaz de Usuario Interactiva.** Este módulo incluye todos los elementos necesarios que permiten no solamente visualizar los datos, sino también importar los archivos hacia el repositorio y coordinar la ejecución de los demás módulos del sistema. Se encuentra implementado en Django, con Python y Bootstrap. Las vistas desarrolladas para este módulo siguen principios de UX (experiencia de usuario), manteniendo un diseño intuitivo que permita una fácil interacción con la plataforma, además de cuidar la responsividad de sus elementos.

Esta plataforma se encuentra diseñada para ser utilizada por analistas de desempeño y equipos de desarrollo. Siendo los analistas los encargados de ejecutar diferentes benchmarks y cargar todos los datos generados en la plataforma. Por otro lado, los equipos de desarrollo se encuentran más enfocados en los resultados proporcionados por la herramienta. La información que se despliega incluye reportes, analíticos, tendencias y alertas que pueden ser utilizadas en el proceso de toma de decisiones mientras se encuentran en desarrollo los componentes del sistema en evaluación. Los resultados de desempeño muestran el comportamiento de nuevos componentes/versiones del producto, comparado con versiones anteriores. Al tener acceso a tendencias y analíticos, los desarrolladores pueden evaluar la calidad del producto.

Para validar los beneficios que aporta la plataforma DDP en el ciclo de desarrollo de software, esta ha sido aplicada a un caso de uso, el cual se describe en la siguiente sección.

4. Caso de Uso

DDP ha sido desplegado en un ambiente de desarrollo para cubrir los requerimientos de datos en el proceso de la evaluación de regresión de desempeño de Spectrum Scale. El equipo de desarrollo realiza la evaluación de regresión para validar que la nueva versión mantiene o mejora el desempeño en comparación con versiones previas. La evaluación de regresión de desempeño se compone de múltiples casos de pruebas para datos, metadatos y operaciones, donde benchmarks son usados para estresar el sistema. Estos benchmarks incluyen IOR (Lawrence Livermore National Laboratory, 2018), MDTEST (Lawrence Livermore National Laboratory, 2017), e IOZONE (IOzone, 2018). Cada caso de prueba tiene variaciones y el objetivo principal es crear cargas de trabajo de lectura y escritura para evaluar el desempeño de diferentes componentes de la configuración de Spectrum Scale. Durante la ejecución de los casos de pruebas, múltiples archivos de datos son creados, incluyendo los resultados de los benchmarks, bitácoras de los sistemas de archivos, archivos de configuración y datos de monitoreo de recursos. Después, todos estos datos tienen que ser recolectados, procesados y presentados de forma que muestren tendencias en el desempeño para el equipo de toma de decisiones del área de desarrollo. El ciclo de vida de los datos de este proceso es complejo. Se requiere lidiar con grandes volúmenes de datos, con diferentes formatos y gestionar una considerable cantidad de preprocesamiento para la combinación, agregación y visualización de datos. Para hacer frente a esta complejidad, DDP ha sido desarrollado en tres componentes primarios, como se muestra en la figura 2. Primero, la capa de aplicación aloja todos los submódulos de la interfaz Web que se compone de tableros

interactivos para presentar los resultados al usuario final. Además, la capa de aplicación también aloja todos los algoritmos para los procesos de ETL, combinación y agregación de datos. El componente de almacenamiento de Big Data se integra con una base de datos de Cassandra y con colecciones de MongoDB para agregar todos los datos previamente procesados y almacenarlos de manera no estructurada como una colección de datos de alta flexibilidad. Este desarrollo permite su fácil escalabilidad en cuanto a su capacidad de almacenamiento de acuerdo con los requerimientos de operación. Finalmente, el motor de procesamiento integrado por Spark y Scala se conecta directamente con el componente de almacenamiento de Big Data, procesa los datos y almacena estos nuevos resultados, para que puedan ser utilizados por la interfaz Web. El motor de procesamiento es también altamente escalable para permitir tiempos de procesamiento más eficientes dependiendo del tamaño de las cargas de trabajo.

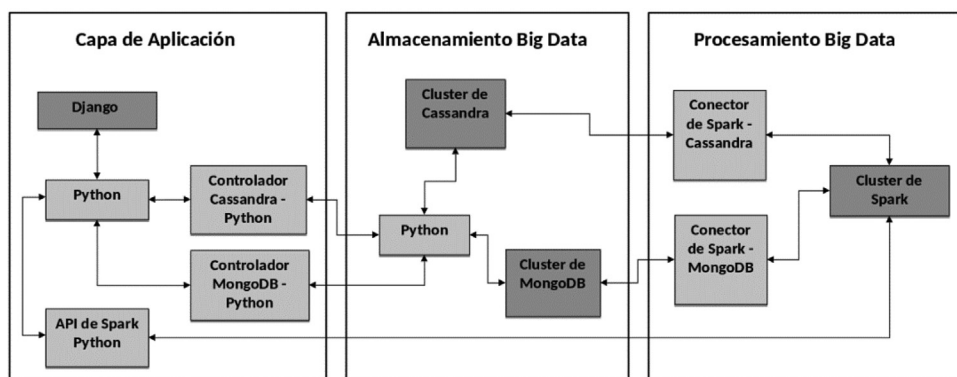


Figura 2 – Estructura general de la plataforma DDP

Esta aplicación mejora de manera significativa el proceso de evaluación de desempeño. Una vez que los datos han sido generados, el analista de desempeño a través de la interfaz interactiva puede, en tan solo unos cuantos clicks, subir los datos. Internamente, el módulo de ETL pre-procesa los datos, los transforma en estructuras apropiadas para su almacenamiento y los carga en el componente de almacenamiento de Big Data. Una vez que los datos han sido almacenados exitosamente, se encuentran disponibles para el motor de procesamiento y para ser explotados. Por lo tanto, en el caso del proceso de evaluación de regresión de desempeño, el analista puede seleccionar las versiones a comparar y obtener los resultados para su evaluación. Esta mejora del proceso permite transparentar toda la complejidad del manejo de datos, reduciendo los tiempos de procesamiento y dejando los datos disponibles para análisis futuros más allá del proceso de evaluación de regresión de desempeño.

Las figuras siguientes muestran algunas interfaces de la aplicación. La figura 3 muestra la ventana donde el analista puede subir los archivos correspondientes a las pruebas de regresión. Estos archivos pueblan la base de datos que se utiliza para generar los reportes de regresión. Todos los datos mostrados en las diferentes interfaces del artículo son ejemplos y no representan resultados reales del proceso de evaluación de desempeño.

The screenshot shows the 'Data Driven Performance' application interface. At the top, there is a navigation bar with 'Home' and 'Upload data' links. The main section is titled 'Version' and contains several input fields: 'Choose a version' (a dropdown menu), 'Build Date' (a date input field showing '02/28/2020'), and 'Cluster' (a dropdown menu). Below these is a link that says 'Register a new version, cluster or test.' and a 'Regression files:' label above a large text area for file paths. At the bottom, there are two buttons: 'Browse files' and 'Upload'.

Figura 3 – Interfaz para subir los datos de las pruebas de regresión

En la figura 4 se muestra la interfaz en donde el analista puede seleccionar dos diferentes versiones para comparar y generar un reporte de regresión, seleccionando también el clúster en el cual se corrieron las pruebas, así como la visibilidad o permisos del reporte, es decir, si desea que el reporte sea público o privado.

The screenshot shows the 'Data Driven Performance' application interface for generating a report. The navigation bar includes 'Home' and 'Generate Report' links. The main section is titled 'Cluster:' and contains several input fields: 'Choose a cluster' (a dropdown menu), 'Baseline execution:' (a dropdown menu with 'Choose a baseline execution' selected), 'New execution:' (a dropdown menu with 'Choose a new execution' selected), and 'Visibility:' (a dropdown menu with 'Choose the visibility of the report' selected). Below these is a 'Report Notes:' label above a large text area. At the bottom, there are two buttons: 'Setup details' and 'Generate'.

Figura 4 – Interfaz para seleccionar las versiones a comparar para generar el reporte de regresión

En la figura 5 se puede observar un reporte de regresión, aquí el analista puede observar si ha habido alguna mejora o regresión en cuanto al desempeño de la aplicación comparando dos versiones diferentes. Con un click en el botón de “Setup Details”, el analista puede obtener la configuración del clúster que fue utilizado para correr este reporte de regresión.

Con base en este reporte, el equipo de desarrollo puede determinar si existen acciones necesarias que deban ser concretadas antes de liberar la nueva versión del producto, debido a alguna regresión presentada durante las pruebas de desempeño.

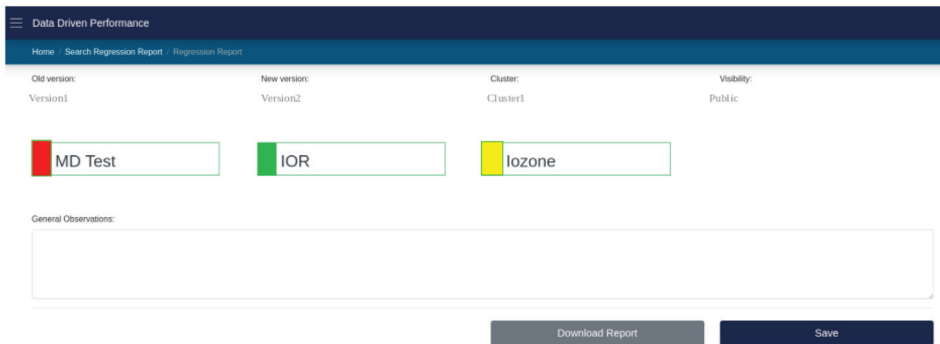


Figura 5 – Reporte de regresión

5. Analíticos de Desempeño en DDP

En cada proceso de desarrollo de software la etapa de pruebas juega un papel primordial. El objetivo de dicha etapa es verificar y validar la calidad del producto que se liberará a los clientes o usuarios finales. Un subconjunto de estas pruebas está relacionado con la evaluación del desempeño del software. El objetivo es detectar problemas que afecten el rendimiento del o los componentes y su interacción con elementos específicos de hardware y aplicaciones de terceros. Los analíticos de desempeño son métricas que permiten observar el rendimiento de componentes de software de forma puntual a través de cifras concretas o elementos visuales.

Algunas de las versiones internas que se generan durante el proceso de desarrollo son elegidas para realizar las pruebas de desempeño. Así tenemos diferentes fechas de pruebas para una misma versión liberada. Cada prueba incluye diferentes benchmarks (MDTest, IOR, IOZONE) y es ejecutada al menos 5 veces. Medimos el número de operaciones por segundo que se pueden realizar, en cada una de las ejecuciones de las pruebas, y tomamos la media para analizar.

Para DDP se eligió realizar dos tipos de análisis de desempeño: un análisis de estadísticos descriptivos y otro de estadísticos predictivos.

En el caso del análisis descriptivo tenemos:

- **Análisis de datos históricos.** Comparamos el desempeño de cada ejecución de las pruebas a través del tiempo.
- **Deltas de desempeño.** Comparación del desempeño de dos versiones internas específicas. Se usa un mapa de calor para representar las deltas del desempeño. El color rojo indica una baja en el desempeño y el color verde una mejora. Se inició esta comparativa únicamente con la versión interna actual y la anterior, pero esto no reflejaba el efecto acumulativo de múltiples decrementos en el desempeño. Así que se decidió calcular esta delta para cualesquiera dos versiones internas, lo que permite obtener una matriz de deltas que es usada para crear el mapa de calor.

- **Resumen de estadísticos descriptivos.** Finalmente tenemos una tabla que nos muestra las tendencias centrales, la dispersión y la forma de la distribución de los datos.

Para el análisis predictivo se eligió un estadístico que mostrará una proyección de las tendencias para predecir los resultados de desempeños en las siguientes versiones.

Las siguientes figuras son algunos ejemplos de los gráficos que se utilizan en el dashboard de análisis de desempeño de la herramienta DDP, esto con la finalidad de encontrar tendencias en los datos para tomar alguna acción necesaria en caso de observar una regresión en el desempeño.

En la figura 6.a (izquierda) se puede observar el desempeño de una operación para un benchmark específico (MDTest) a lo largo de sus diferentes build dates, con esta gráfica el analista puede detectar alguna tendencia en el desempeño para cada una de las operaciones y dar aviso al equipo de desarrollo. Por su parte, en la figura 6.b (derecha) se observa la gráfica de la distribución del desempeño de una operación específica (Directory creation mean) para comparar el comportamiento de la misma en sus diferentes build dates.

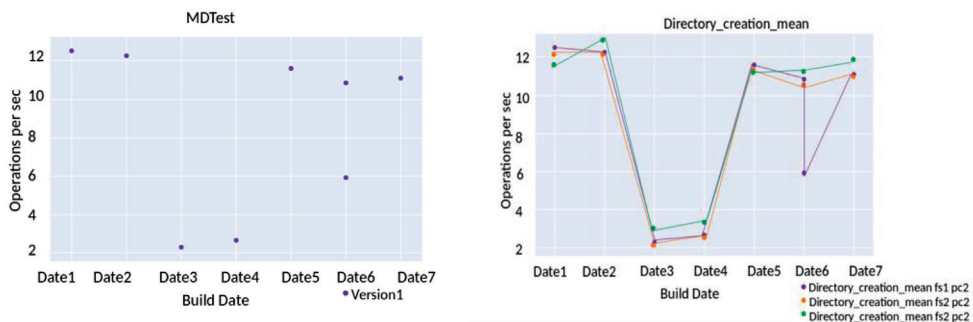


Figura 6.a – Gráfica de dispersión de una versión con respecto a sus build dates.

Figura 6.b – Gráfica de dispersión de una operación específica, para los diferentes build dates

En la figura 7.a (izquierda) podemos encontrar una comparación entre el desempeño de todas las operaciones de una versión en específico para todos los build dates de la misma, en donde un color verde fuerte indica una mejora y un color rojo intenso indica una regresión. En la Figura 7.b (derecha) podemos observar un ejemplo de los estadísticos que se pueden mostrar en el dashboard para ser analizados por el analista de desempeño.

En la actualidad se pueden encontrar muchos ejemplos a nivel mundial sobre los beneficios de analíticos en el proceso de evaluación del desempeño orientado a datos. Un caso interesante es el de Siemens (Siemens Deutschland, 1996 –2020), una fábrica inteligente de la cuarta revolución industrial, donde los productos dirigen su propia fabricación con una producción altamente automatizada, gracias a la analítica de sus datos (euskadi.eus, 2015).

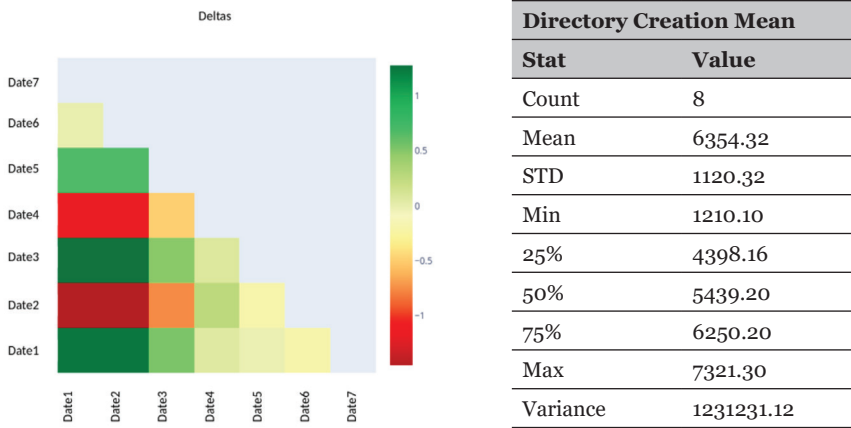


Figura 7.a – Gráfica de deltas de una versión específica, para los diferentes build dates.

Figura7.b – Ejemplo de estadísticos usados para el análisis de desempeño

Por otra parte, la plataforma DDP, al contar con un repositorio centralizado que permite almacenar, organizar y visualizar resultados de desempeño, ayuda a los equipos de desarrollo en la toma de decisiones. Esto por medio de reportes, analíticos, tendencias y alertas.

6. Conclusiones y Trabajo Futuro

Las evaluaciones de desempeño de un producto son complejas debido a la variedad en software, hardware, configuración de redes y características específicas de los datos que pueden ser integrados en los sistemas actuales. Un emergente enfoque orientado a datos se presenta en diversas organizaciones que reconocen el valor de los datos como una ventaja competitiva y diferenciador en el mercado, brindando un mayor soporte en la toma de decisiones que mejoren la calidad de sus productos.

Este artículo describe una plataforma basada en un modelo orientado a datos como una propuesta de transformación en el proceso de evaluación de desempeño. Esta plataforma centraliza, almacena, organiza y visualiza los datos de desempeño producidos durante las evaluaciones ejecutadas en el ciclo de desarrollo de un producto. El artículo describe la arquitectura y el conjunto de tecnologías usadas en DDP, así como el proceso de transformación, manipulación y visualización de los datos con la implementación de interfaces interactivas; considerando que estos provienen de diversas fuentes y formatos, por ejemplo: benchmarks, logs, configuraciones de hardware o software. Finalmente, se describe un caso de éxito de la implementación de DDP y las mejoras aportadas a un proceso real de desarrollo de Software.

Algunas de las conclusiones claves que destacamos a partir del desarrollo e implementación de DDP, se listan a continuación:

- Las herramientas orientadas a datos son clave para las mejoras en el proceso de software. Estas pueden ayudar a generar nuevas perspectivas sobre la calidad de

software, adicional a la información que brindan las pruebas funcionales. Una explotación eficiente de los datos puede arrojar tendencias, alertas, modelos predictivos y de desempeño que faciliten la evaluación de la calidad de un producto.

- Una plataforma como DDP acelera el camino a la Inteligencia Artificial. DDP permite recolectar y centralizar los datos de diversos tipos y fuentes. Transformándolos en datos simples y accesibles, organizados en modelos con sentido y fácil de manejar. Los datos se preparan para la fase de análisis, ayudando a los desarrolladores a tomar mejores decisiones que con un impacto positivo en la calidad del producto.
- DDP automatiza la carga de datos y generación de reportes. En el caso de uso se describe como el analista puede visualizar si se presentó una mejora o degradación en el desempeño con solo seleccionar la versión de interés. Hay un ahorro de tiempo implícito que puede ser re-invertido por el analista en la investigación de la causa de los resultados.
- Un modelo orientado a datos con un repositorio central permite explotar los datos históricos e identificar las tendencias entre cualquier periodo de tiempo.
- Un modelo orientado a datos con analíticos y visualizaciones reduce la dependencia de expertos para poder leer y comprender los resultados de las evaluaciones de desempeño.

Actualmente la plataforma cumple el objetivo de automatizar actividades manuales que realizaban los analistas durante las evaluaciones de regresión de desempeño, mejorando así el proceso de desarrollo de software. Se han incorporado los modelos predictivos y sus visualizaciones en dashboards para optimizar el análisis de causa-efecto gracias a los reportes, tendencias, y alertas que despliega la plataforma. Los trabajos futuros se enfocan en profundizar en la explotación de los datos y el desarrollo de modelos predictivos que nos conduzcan a configuraciones del producto que brinden el óptimo desempeño ante diferentes escenarios o modelos cognitivos que permitan anticipar una degradación de desempeño previo a que sea introducido.

Referencias

- Aldrich, J. O., & Cunningham, J. B. (2015). *Using IBM® SPSS® Statistics: An Interactive Hands-On Approach*. (S. Publications, Ed.)
- Aleti, A., Trubiani, C., van Hoorn, A., & Jamshidi, P. (2018). *An efficient method for uncertainty propagation in robust software performance estimation*. DOI: <https://doi.org/10.1016/j.jss.2018.01.010>
- Dain, J., Forestier, E., Guaitani, P., Guaitani, P., Haas, R., Maestas, C., . . . Sherman, B. (2018). *IBM Software-Defined Storage Guide*. Obtenido de [redbooks.ibm.com](http://www.redbooks.ibm.com/abstracts/redp5121.html?Open): <http://www.redbooks.ibm.com/abstracts/redp5121.html?Open>
- Denar, G., Polini, A., & Emmerich, W. (2005). *Performance Testing of Distributed Component Architectures*. DOI: https://doi.org/10.1007/3-540-27071-x_14
- Dessau, R., & Bressen Pipper, C. (2007). *"R"-project for statistical computing*. Obtenido de Europe PMC: <http://europepmc.org/article/med/18252159#abstract>

- École Polytechnique Fédérale Lausanne. (2002-2020). *The Scala Programming Language*. Obtenido de scala-lang: <https://www.scala-lang.org/>
- euskadi.eus. (2015). *Visita de Arantxa Tapia a la planta central de SIEMENS en Amberg (Baviera)*. Obtenido de euskadi: https://www.euskadi.eus/gobierno-vasco/contenidos/noticia/2015_10_08_28651/es_28651/28651.html
- Foo, K. C., Jiang, Z. M., Adams, B., Hassan, A., Zou, Y., & Flora, P. (2015). An Industrial Case Study on the Automated Detection of Performance Regressions in Heterogeneous Environments. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. DOI: <https://doi.org/10.1109/icse.2015.144>
- Foo, K. C., Jiang, Z. M., Adams, B., Hassan, E., A., Zou, Y., & Flora, P. (2015). Automated Detection of Performance Regressions Using Regression Models on Clustered Performance Counters. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, 15–26.
- Huijgens, H., Spadini, D., Stevens, D., Visser, N., & van Deursen, A. (2018). Software analytics in continuous delivery: a case study on success factors. In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software*, 1-10.
- IBM Hybrid Cloud. (2018). *Accelerate the journey to AI*. Obtenido de ibm: <https://www.ibm.com/downloads/cas/MZEA2GKW>
- IOzone. (2018). *IOzone Filesystem Benchmark*. Obtenido de iozone: <http://www.iozone.org>
- Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2016). The emerging role of data scientists on software development teams. In: *Proceedings of the 38th International Conference on Software Engineering*, 96–107. DOI: <https://doi.org/10.1145/2884781.2884783>
- Kroß, J., Willnecker, F., Zwickl, T., & Krcmar, H. (2016). PET: continuous performance evaluation tool. In: *Proceedings of the 2nd International Workshop on Quality-Aware DevOps*, 42–43. DOI: <https://doi.org/10.1145/2945408.2945418>
- Lawrence Livermore National Laboratory. (2020). *Used for testing the metadata performance of a file system*. Obtenido de <https://github.com/LLNL/mdtest>
- Lawrence Livermore National Laboratory. (2020). *Parallel filesystem I/O benchmark*. Obtenido de <https://github.com/LLNL/ior>
- Lizy Kurian, J., & Lieven, E. (2018). *Performance Evaluation and Benchmarking*. CRC Press LLC.
- Luís Berral, J., Poggi, N., Carrera, D., Call, A., Reinauer, R., & Green, D. (2017). ALOJA: A Framework for Benchmarking and Predictive Analytics in Hadoop Deployments. *IEEE Transactions on Emerging Topics in Computing*, 5, 480-493. DOI: <https://doi.org/10.1109/tetc.2015.2496504>
- Menzies, T., & Zimmermann, T. (2013). Software Analytics: So What? *Software IEEE*, 30, 31-37. Obtenido de ieeexplore.ieee.org: <https://ieeexplore.ieee.org/document/6547619>

- Mockus, A. (2014). *Engineering big data solutions*. DOI: <https://doi.org/10.1145/2593882.2593889>
- mongoDB. (2020). *The database for modern applications*. Obtenido de mongodb: <https://www.mongodb.com/>
- Mostafa, S., Wang, X., & Xie, T. (2017). PerfRanker: prioritization of performance regression tests for collection-intensive software. In: *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 23–34. DOI: <https://doi.org/10.1145/3092703.3092725>
- Okanović, D., van Hoorn, A., & Zorn, C. (2019). Concern-driven Reporting of Software Performance Analysis Results. In: *Companion of the 2019 ACM/SPEC International Conference on Performance Engineering*, 1-4.
- Price, W. (1989). A benchmark tutorial. *IEEE Micro*, 9, 28-43.
- Quintero, D., Bolinches, L., Chaudhary, P., Davis, W., Duersch, S., Fachim, C., . . . Weiser, O. (2015). *IBM Spectrum Scale (formerly GPFS)*. Obtenido de <http://www.redbooks.ibm.com/redbooks/pdfs/sg248254.pdf>
- Anbunathan, R., & Basu, A. (2015). Data driven architecture based automated test generation for Android mobile. *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. DOI: <https://doi.org/10.1109/iccic.2015.7435772>
- Siemens Deutschland. (1996 – 2020). *Unser Anliegen: eine bessere Zukunft für alle*. Obtenido de new.siemens: <https://new.siemens.com/de/de/unternehmen/jobs.html>
- The Apache Software Foundation. (2016). *Manage massive amounts of data, fast, without losing sleep*. Obtenido de cassandra.apache: <http://cassandra.apache.org/>
- The Apache Software Foundation. (2018). *Apache Spark™ is a unified analytics engine for large-scale data processing*. Obtenido de spark.apache: <https://spark.apache.org>
- Velcu-Laitinen, O., & Yigitbasioglu, O. (2012). The Use of Dashboards in Performance Management: Evidence from Sales Managers. *The International Journal of Digital Accounting Research*, 12.
- Yandrapally, R., Thummalapenta, S., Sinha, S., & Chandra, S. (2014). Robust test automation using contextual clues. In: *Proceedings of the 2014 International Symposium on Software Testing and Analysis*, 304. DOI: <https://doi.org/10.1145/2610384.2610390>
- Yigitbasioglu, O., & Velcu, O. (2012). *A review of dashboards in performance management: Implications for design and research*. DOI: <https://doi.org/10.1016/j.accinf.2011.08.002>

Algoritmo para controlar un brazo antropomórfico usando una transformación lineal

Yadira Quiñonez¹, Oscar Zatarain¹, Carmen Lizárraga¹, Juan Peraza¹,
Juan Peraza¹, Jezreel Mejía²

yadiraqui@uas.edu.mx, 2016030617@upsin.edu.mx, carmen.lizarraga@uas.edu.mx,
jfperaza@uas.edu.mx, restrada@uas.edu.mx ozatarain@uas.edu.mx, jmejia@cimat.mx

¹ Universidad Autónoma de Sinaloa, Facultad de Informática, Av. Leonismo Internacional S/N y Av. de los Deportes, Ciudad Universitaria, 82000, Mazatlán, Sinaloa, México.

² Centro de Investigación en Matemáticas, Unidad Zacatecas, Parque Quantum, Ciudad el Conocimiento Avenida Lassec, Andador Galileo Galilei, Manzana, 3 Lote 7, 98160. Zacatecas, México.

DOI: 10.17013/risti.36.65–81

Resumen: En este artículo se propone un algoritmo para manipular un brazo antropomórfico utilizando una pantalla táctil LCD y una transformación lineal para obtener la trayectoria del brazo robótico, cinemática inversa y directa. Con la finalidad de que las personas con discapacidades físicas que tienen algún problema para mover sus cuerpos o simplemente no tienen la fuerza para moverlo puedan controlar el brazo robótico en las actividades cotidianas de una forma natural y fácil. La implementación del algoritmo se ha desarrollado en MATLAB, a través de esta simulación se visualiza la trayectoria del brazo robótico y la representación con las tres articulaciones que son movidas por motores paso a paso. El artículo termina con una discusión crítica de los resultados experimentales.

Palabras-clave: Brazo Antropomórfico; Transformación Lineal; Pantalla Táctil LCD; Personas con Discapacidades Físicas; MATLAB.

Algorithm to control an anthropomorphic arm using a linear transformation

Abstract: In this paper proposes an algorithm to manipulate an anthropomorphic arm using an LCD touch screen and a linear transformation to obtain the trajectory robotic arm, inverse and direct kinematics. In order to people with physical disabilities who have an issue to move their bodies or simply do not have enough force to move it, can control the robotic arm in daily activities in a natural and easy way. The implementation of the algorithm has been developed in MATLAB, through this simulation the trajectory of the robotic arm is visualized and the representation with the three joints that are moved by stepper motors. The article ends with a critical discussion of the experimental results.

Keywords: Anthropomorphic Arm; Linear Transformation; LCD Touch Screen; People with Physical Disabilities; MATLAB.

1. Introducción

En las últimas dos décadas, la automatización y el control se han convertido en un tema de interés para investigadores en diferentes áreas de aplicación, donde lo más importante es la Experiencia de Usuario (UX) (Fernández-Ordoñez, 2019), esto con la finalidad de facilitar la interacción hombre-máquina tan natural como sea posible (Lara, 2019). Existe una gran variedad de aplicaciones de automatización y control, sin embargo, el área médica ha sido muy beneficiada, a continuación, se mencionan algunos trabajos relacionados con esta área. En la robótica industrial (Grau, 2017, Yenorkar, 2018) y en los sistemas robotizados aplicados en el área médica, tales como la cirugía teleoperada (Burgner-Kahrs, 2015), el corte de patrones quirúrgicos (Murali, 2015), según Shademan et al. (2016) la precisión de un robot industrial ofrece grandes ventajas en esta área. Actualmente, existen un sinnúmero de desarrollos tecnológicos y diversas aplicaciones como prótesis (Allen, 2016), ortesis (Niyetkaliyev, 2017), exoesqueletos (Proietti, 2016, Rehmat 2018, Young, 2017) y dispositivos para la teleoperación con el fin de mejorar las capacidades humanas (Makin, 2017, Beckerle, 2018, Jiang, 2012, Kruthika, 2016). De acuerdo con Chung et al. (2013) realizó una revisión de la literatura sobre los diferentes manipuladores de asistencia robótica desde 1970 hasta 2012, menciona que la confiabilidad, rentabilidad, apariencia, funcionalidad y facilidad de uso son factores determinantes para lograr una comercialización exitosa. Algunos trabajos relacionados para la rehabilitación de pacientes con capacidad de movimiento reducida han utilizado la realidad virtual para mejorar la manipulación de los movimientos e influir positivamente en un entorno virtual (Perez-Marcos, 2017, Levin, 2015, Kokkinara, 2015, Bovet, 2018). Recientemente, en un trabajo de Atre et al. (2018) han utilizado técnicas de visión por computadora para crear un brazo robótico controlado por gestos, mencionan la importancia de que un brazo robótico sea más eficiente en relación con la precisión y la velocidad, en otros trabajos, Badrinath et al. (2016) proponen el control de un brazo robótico utilizando una cámara Kinect 3-D para rastrear los movimientos de las personas mediante la técnica de visión por computador.

2. Descripción del algoritmo

En la actualidad, es común utilizar un joystick para controlar los movimientos de un brazo robótico, de hecho, el uso del joystick es implementado en muchos tipos de aplicaciones electrónicas y una de estas, en la que se enfoca este trabajo, es controlar un brazo robótico antropomórfico para dar asistencia a las personas con discapacidades físicas. El Joystick puede ser una gran herramienta de control, sin embargo, al momento de implementarlo en alguna problemática en particular como lo es el brazo robótico para personas con discapacidades físicas no cubre con las necesidades o expectativas para sacar el máximo aprovechamiento, debido a que se requiere de un esfuerzo constante y tiempo de espera para ejercer una fuerza, sin mencionar que se necesita coordinación al momento de ejecutarlo. Suponiendo que un paciente también sufre de diabetes (en el caso de hipoglucemia) o para las personas que sufren de lupus u otra enfermedad que impida tener una buena coordinación, usar un Joystick puede ser una alternativa complicada para las personas que sufren alguno de estos casos, ya que a veces se requieren movimientos muy largos para que el brazo robótico llegue al punto

deseado, y aunque el Joystick sea muy fácil de usar y programar, controlar un brazo robótico puede ser aún más fácil, sin tanto esfuerzo, sin aplicar una fuerza constante, de manera intuitiva y con un solo dedo poder mover el brazo robótico hasta el punto deseado. Esto es posible utilizando una pantalla LCD táctil, que ayuda a ejecutar el algoritmo usando una transformación lineal (también llamada función lineal, aplicación u operador lineales) la cual es una herramienta importantísima para la ejecución del algoritmo y diversos problemas, en el siguiente apartado se desarrolla este tema.

Al implementar este control, una persona con limitaciones físicas solo es necesario realizar un ligero movimiento con el dedo y tocar un punto en la superficie de la pantalla táctil, para hacer que el brazo robótico obtenga un punto en el espacio, realizar este proceso es similar a utilizar un teléfono celular, una tableta o cualquier otro aparato electrónico que cuente con una pantalla táctil y que se puedan realizar las actividades cotidianas que se realizan hoy en día, de esta forma, se proporciona al paciente más libertad para controlar un brazo robótico en las actividades cotidianas de una forma natural.

El algoritmo está pensado para que el brazo robótico antropomórfico llegue al punto deseado a más tardar en un segundo, es decir, todas las articulaciones se moverán para que el brazo robótico llegue al punto deseado en menos de un segundo si la posición es menor que a la posición máxima, en otro caso, las articulaciones se mueven para que el brazo robótico llegue en un segundo si la posición es la máxima. Por ejemplo, este algoritmo está pensado para que una articulación llegue como máximo en 1 segundo a 180 grados, por lo que los grados α serán menor o iguales a 180 grados ($\alpha \leq 180$) en sentido horario o antihorario dependiendo el punto que se haya tocado en la pantalla LCD. En este sentido, de la pantalla táctil LCD se puede obtener una coordenada (x,y) , sin embargo, es necesario obtener otra dimensión para poder mover el brazo robótico en el espacio, entonces, de las coordenadas (x,y) se puede obtener un tercer dato, el cual es $\alpha = \tan^{-1}(\frac{y}{x})$, por lo tanto, se puede decir que x es diferente a cero y los números x,y y α serán representados en intervalos de entre cero y 1, entonces, y se encuentra en un conjunto cerrado $[0,1]$, x estará en el conjunto $[0,1]$, y α estará en el intervalo $[0,1]$. Otro punto importante es que las articulaciones son movidas por motores paso a paso, sin embargo, este método también se puede utilizar cuando se tiene otro tipo de motor que use un pulso PWM para el control de su torque, pero tiene mejor rendimiento cuando se utilizan motores paso a paso.

3. Transformación lineal y condiciones

Para representar una coordenada que se toca en la pantalla táctil LCD, se tienen cuatro condiciones, en las cuales los valores de x e y son caracterizados por ser mayor o igual que el otro, es decir, se puede tener el caso donde $x \geq y$ o $y \geq x$ de esta manera se identifican las cuatro condiciones, se construyen cuatro triángulos rectángulos que cubren la pantalla táctil LCD la cual se cumple que $x \geq y$ o $y \geq x$.

Primera condición: En esta condición se obtiene una coordenada (x,y) la cual se obtenga un $\alpha \leq 45$, entonces, construyendo un triángulo rectángulo se obtiene lo siguiente:

$$\alpha \leq 45^\circ \Rightarrow \tan^{-1}\left(\left(\frac{y}{x}\right)\right) \leq 45 \Rightarrow \frac{y}{x} \leq \tan(45) \Rightarrow \frac{y}{x} \leq 1 \quad (1)$$

Para $|x| \geq |y|$, el brazo robótico se mueve en el área del triángulo rectángulo que se muestra en la figura 1, con un x en $(0,1]$, y un y en $[0,1]$, de tal forma que α esté en el intervalo $[0,0.25]$.

Segunda condición: Al obtener la coordenada (x,y) , se tiene que $y \geq x$ y se obtiene un $\alpha \geq 45$, con un x en $(0,1]$, y un y en $[0,1]$ tal que α se encuentre en el intervalo $[0.25,0.50]$, esto sucede debido a que el movimiento del brazo robótico es realizado en el área del segundo triángulo rectángulo como se muestra en la figura 1, por lo tanto, para $45 \leq \alpha \leq 90$, y para que $\alpha = 90$, entonces, $\left(\frac{y}{x}\right)$ tiene que alcanzar un valor muy grande, por lo que $\alpha = 90$ cuando $y > 0$ y x se aproxima tanto a 0 tomando en cuenta que x debe ser $x > 0$.

$$\lim_{(x) \rightarrow 0^+} \alpha = \lim_{(x) \rightarrow 0^+} \tan^{-1}\left(\frac{1}{x}\right) = \frac{\pi}{2} = 90 \quad (2)$$

Es decir, existe el límite para cuando x se aproxima a cero, pero no existe $\alpha(0)$ porque se trata de una función divergente que tiene discontinuidad de salto.

Tercera condición: la coordenada (x,y) se posiciona en el área del triángulo rectángulo número 3, es fácil observar porque se obtiene $|y| \geq |x|$, entonces $90 \leq \alpha \leq 135$, por lo que,

$$\alpha = \tan^{-1}\left(\frac{y}{x}\right) + 180 \quad (3)$$

Se tiene entonces que x se encuentra en $[-1,0)$, y en $[0,1]$, y α en $[0.50,0.75]$.

Cuarta condición: por último, la cuarta condición proporciona la coordenada (x,y) , se encuentra en el área del cuarto triángulo rectángulo, por lo que nuevamente se tiene que $|x| \geq |y|$ y α es definida igual como al caso de la tercera condición pero ahora $135 \leq \alpha \leq 180$ por ser $|x| \geq |y|$, por lo que se tiene que x se localiza en $[-1,0)$, y en $[0,1]$ y α en $[0.75,1]$. A continuación, en la Figura 1 se representan las cuatro condiciones.

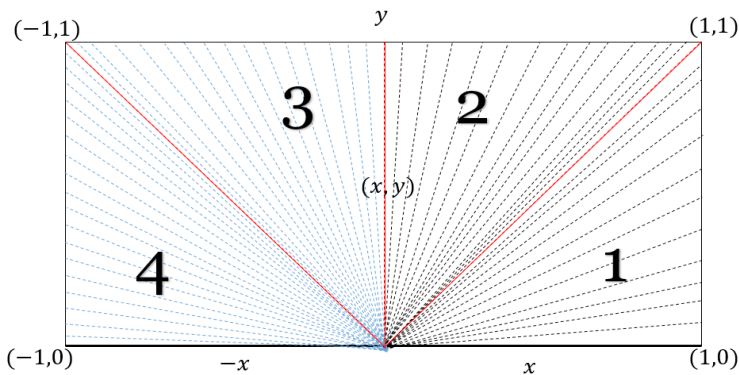


Figura 1 – Representación de las cuatro condiciones, donde el brazo robótico se coloca

dependiendo la ubicación de la coordenada.

Transformación Lineal: En realidad, la figura 1 es la representación de la transformación lineal que se comentó en el apartado de desarrollo del algoritmo. Se sabe que una pantalla táctil LCD es un espacio con dos dimensiones y se necesita mover el brazo robótico en un espacio con 3 dimensiones. Con los datos que hemos definido, tenemos entonces un vector (x, y, α) , se puede definir la siguiente transformación, sea $T: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ y k es cualquier número real. Se define la siguiente transformación lineal para cuando α gira en un sentido antihorario.

$$T \begin{pmatrix} x \\ y \\ \alpha \end{pmatrix} = \begin{pmatrix} kx \\ ky \\ k\alpha \end{pmatrix} \quad (4)$$

Si α tiene que girar en el sentido horario, entonces se utiliza la siguiente transformación lineal.

$$T \begin{pmatrix} x \\ y \\ \alpha \end{pmatrix} = \begin{pmatrix} -kx \\ ky \\ k\alpha \end{pmatrix} \quad (5)$$

La transformación lineal (5) es utilizada cuando la garra del brazo robótico se encuentre en la tercera o cuarta condición y se necesite llegar a la primera o segunda condición (o si la garra está en la cuarta condición, y se necesite llegar a la primera, segunda o tercera condición de manera que α tenga que girar en un sentido horario), entonces α necesitaría rotar de forma en sentido inverso, es decir, las condiciones se intercambian de lugar, la primera condición por la cuarta y la segunda condición por la tercera, tal y como se muestra a continuación:

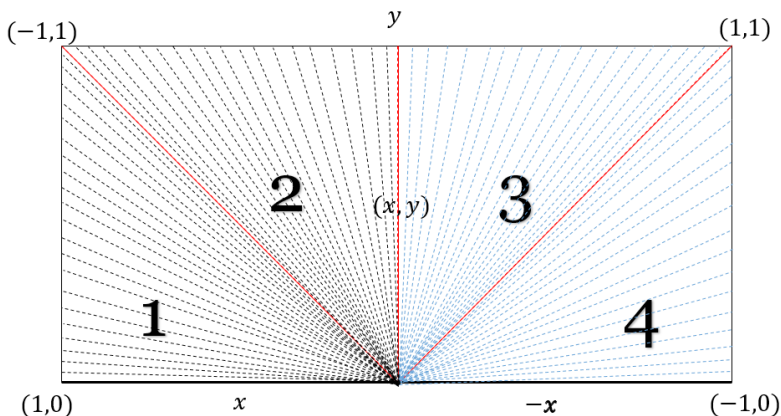


Figura 2 – Representación de las cuatro condiciones y uso de la segunda transformación lineal.

Por ejemplo, se tiene una coordenada inicial $(-x_0, y_0)$ y se encuentra en la cuarta condición de la figura 1, quiere decir que $-x_0 \geq y$, y si se quiere llegar a la tercera condición, entonces

$$T \begin{pmatrix} -x_0 \\ y_0 \\ \alpha \end{pmatrix} = \begin{pmatrix} -k(-x_0) \\ ky_0 \\ k\alpha \end{pmatrix} = \begin{pmatrix} k(x_0) \\ ky_0 \\ k\alpha \end{pmatrix} \quad (6)$$

Entonces $(kx_0, ky_0, k\alpha)$ y también $x_0 \leq -y$, luego $|x_0| \leq |-y| = x_0 \leq y_0$, por lo que efectivamente se encuentra el vector en la segunda condición de la segunda transformación lineal representada en la figura 2.

Después, si se tiene una coordenada inicial (x_0, y_0) y se encuentra en la tercera condición vista desde el plano de la figura 1 y se desea llegar a la segunda condición, entonces $y_0 \geq x_0$ y viendo el plano como en la figura 2, entonces se toma la transformación lineal (5)

$$T \begin{pmatrix} x_0 \\ y_0 \\ \alpha \end{pmatrix} = \begin{pmatrix} -kx_0 \\ ky_0 \\ k\alpha \end{pmatrix} = \begin{pmatrix} -kx_0 \\ ky_0 \\ k\alpha \end{pmatrix} \quad (7)$$

Se tiene que $(-kx_0, y_0, k\alpha)$ y se sabe que $y \geq x_0$, por lo que viendo al nuevo vector se tiene que $y \geq x_0 \geq -x_0$, luego $|y| \geq |x_0|$, teniendo entonces que $y_0 \geq x_0$, por lo que el vector se encuentra en la tercera condición de la figura 2 y en la segunda condición visto desde el plano de la Figura 1.

Es muy fácil demostrar que las transformaciones son lineales ya que cumple con las condiciones para que una transformación sea lineal, primero se seleccionan dos vectores:

$$v_1 \begin{pmatrix} x_1 \\ y_1 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ \alpha_2 \end{pmatrix} \quad (8)$$

Entonces, se prueba la primera condición:

$$T(v_1 + v_2) = T \left(\begin{pmatrix} x_1 \\ y_1 \\ \alpha_1 \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \\ \alpha_2 \end{pmatrix} \right) = \begin{pmatrix} k(x_1 + x_2) \\ k(y_1 + y_2) \\ k(\alpha_1 + \alpha_2) \end{pmatrix} = \begin{pmatrix} kx_1 \\ ky_1 \\ k\alpha_1 \end{pmatrix} + T \begin{pmatrix} kx_2 \\ ky_2 \\ k\alpha_2 \end{pmatrix} = T(v_1) + (v_2) \quad (9)$$

Probando la segunda condición, tomando a λ en el conjunto de los reales, entonces:

$$\lambda T(v_1) = \lambda T \begin{pmatrix} x_1 \\ y_1 \\ \alpha_1 \end{pmatrix} = \lambda \begin{pmatrix} kx_1 \\ ky_1 \\ k\alpha_1 \end{pmatrix} = \begin{pmatrix} \lambda kx_1 \\ \lambda ky_1 \\ \lambda k\alpha_1 \end{pmatrix} = T \left(\begin{pmatrix} \lambda x_1 \\ \lambda y_1 \\ \lambda \alpha_1 \end{pmatrix} \right) = T \left(\lambda \begin{pmatrix} x_1 \\ y_1 \\ \alpha_1 \end{pmatrix} \right) = T(\lambda v_1) \quad (10)$$

Por lo que T es una transformación lineal, de la misma manera se demuestra para la transformación lineal (5).

4. Desarrollo del Algoritmo

Utilizando la pantalla táctil LCD, se intenta mover tres motores paso a paso para llegar a las posiciones correspondientes dada por la transformación lineal, cada motor tiene una resolución de 1.8 grados por paso la cual se alcanza cuando se le aplica a la bobina del motor un pulso PWM de 10 milisegundos. Se nombra al motor α como el motor de la primera articulación, este motor hace girar al brazo robótico en el sentido horario y anti horario en el eje vertical paralelo al brazo robótico, después se define el motor x el cual se mueve a través del eje x de la figura 1, y por último el motor y se moverá a través del eje y . Para ver mejor la forma en la que se mueve el brazo robótico, se utilizó como ejemplo el brazo, el motor α representa a los movimientos que hace el hombro de manera vertical, el motor y representa los movimientos de reflexión y contracción del codo, por último, el motor x representa los movimientos del hombro de manera horizontal, posicionando nuestra mano en un punto en el espacio. Se sabe que la resolución del motor es de 0° grados y que se necesita un pulso PWM para mover un paso de 1.8 grados, entonces, los grados obtenidos son por el número de pasos de 10 milisegundos, por lo que se construye la siguiente función:

$$f(n) = 1.8n \quad (11)$$

donde n es el número de pulsos PWM con grosor de 10 milisegundos, quiere decir que, para alcanzar 180 grados, se necesitarán 100 pulsos, por lo tanto, se alcanzan en 1 segundo. La función discreta (11) funciona de una forma correcta, de hecho, es una función precisa, sin embargo, al momento de utilizar la pantalla LCD y mandar los pulsos PWM se debe hacer por lo menos una pausa de 5 milisegundos para poder mandar otro pulso de 10 milisegundos, de no ser así, puede hacer que las uniones del brazo robótico no lleguen a la posición deseada, pues se genera un tráfico en cada unión del brazo robótico. Para llegar 180 grados se necesita más de un segundo para llegar, entonces, se propone la siguiente función, la cual puede alcanzar una cantidad muy grande con un solo pulso.

$$f(s) = \sum_{n=0}^s 1.8n \quad (12)$$

Esta función incrementa la anchura del pulso dependiendo el valor de s , esta función es muy fácil de programar y aunque no es muy precisa, puede ser combinada con la función (11) para obtener un valor preciso. El límite superior de la sumatoria de la función (12) puede tomar los siguientes valores $s = \{1,2,3,4,5,6,7,8,9,10,11,12,13\}$, así que mientras n incrementa, el grosor del pulso incrementa, tal y como se muestra en la siguiente tabla.

$f(s)$	s	Tiempo
1.8	1	10 ms
5.4	2	20 ms
10.8	3	30 ms
18	4	40 ms
27	5	50 ms

$f(s)$	s	Tiempo
37.8	6	60 ms
50.4	7	70 ms
64.8	8	80 ms
81	9	90 ms
99	10	100 ms
118.8	11	110 ms
140.4	12	120 ms
163.8	13	130 ms

Tabla 1 – grosor de los pulsos usando la función $f(s)$.

4.1. Representación del Algoritmo

Hasta el momento, se han presentado las dos funciones discretas con las que el algoritmo trabaja, en resumen, el algoritmo combina las funciones discretas para alcanzar los grados deseados utilizando la menor cantidad de pulsos o grosor de los pulsos, luego, el número de combinaciones con la que se puede usar las funciones es 2^n formas, cada manera puede tener diferente tiempo y algunas veces algunas combinaciones pueden tener la misma cantidad de tiempo para alcanzar los grados deseados. Además, se debe saber que algunas combinaciones pueden ser más exactas que otras, pero más lentas en alcanzar los grados deseados. Sin embargo, la mejor combinación es encontrada por el algoritmo y puede anticipar la localización del tiempo de la mejor combinación.

$$\alpha \geq \lim_{t_s \rightarrow b} \frac{2^n}{t_s} \geq \alpha - b \quad (13)$$

El número α representa la cantidad de los grados deseados, el número b ayuda a tener el intervalo para encontrar los límites de tiempo (límite superior o inferior) y ayuda a que la inecuación se cumpla, el número 2^n es la cantidad de combinaciones en las que se encuentra nuestra combinación a buscar. Para desarrollar el algoritmo, es importante saber qué $\alpha < 2^n$ y b puede tener valores menores o iguales a $(2^n/\alpha)$, también, puede ser redondeado mientras la inecuación se cumpla. Entonces, cuando se encuentra el valor de b , se iguala a la división del tiempo límite superior entre el tiempo límite inferior, tal y como se muestra a continuación:

$$b = \left(\frac{t_p}{t_d} \right) \quad (14)$$

Donde t_p es el tiempo límite superior y t_d es el tiempo límite inferior. Cuando se sabe los límites de tiempo, se puede estimar el tiempo que le corresponde a la combinación que alcanza los grados deseados.

$$t_p \geq t \geq t_d \quad (15)$$

Encontrar el número b puede ser difícil cuando el número deseado es muy grande, sin embargo, se tiene que poner atención a los límites de tiempo del comportamiento de la inecuación para cuando el número n es repetitivo, esto quiere decir que el límite superior esta cerca al tiempo medio o máximo del proceso (500 milisegundos o 1000 milisegundos). Por ejemplo, un ejemplo fácil y claro es para cuando se toma un $\alpha = 16.2$, entonces:

$$16.2 \geq \lim_{t_s \rightarrow b} \frac{2^n}{t_s} \geq 16.2 - b \quad (16)$$

Entonces n debe ser igual a 5, debido a que $16.2 < 2^5$ y $b \leq \left(\frac{2^5}{16.2}\right) = 1.9753 \dots$, pero b puede tener el valor 2, ya que la inecuación se cumple, entonces:

$$16.2 \geq \lim_{t_s \rightarrow 2} \frac{2^5}{t_s} \geq 14.2 \quad (17)$$

Como $b = \left(\frac{t_p}{t_d}\right)$ y $n = 5$, se puede decir que el tiempo limite inferior es 50 milisegundos, entonces:

$$2 = \left(\frac{t_p}{50ms}\right) \Rightarrow t_p = 100ms \quad (18)$$

Por lo que la combinación se encuentra para cuando $n = 5$ con 32 formas de combinar las dos funciones discretas en un tiempo entre 50 milisegundos y 100 milisegundos. En la Figura 3, se muestra como se construye la combinación:

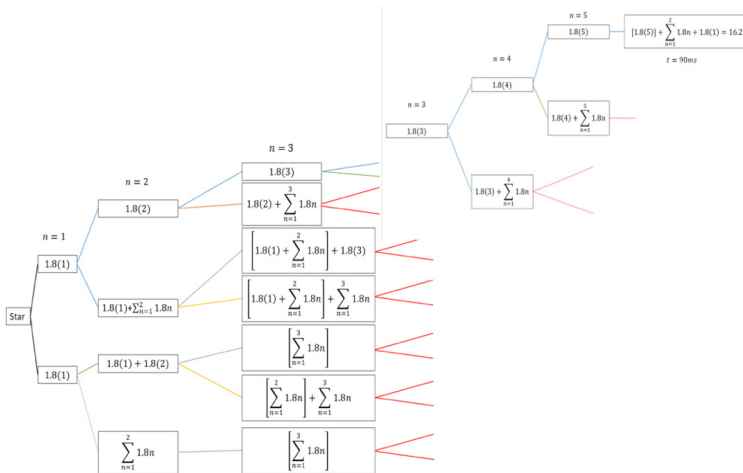


Figura 3 – Desarrollo del algoritmo para cuando $\alpha = 16.2$, los caminos de color rojo son los que fueron rechazados al momento de hacer las combinaciones.

El resultado es $[1.8(5)] + \sum_{n=1}^5 1.8n = 16.2$ en un tiempo de 90 milisegundos.

4.2.Simulación del Algoritmo en Matlab

Se ha seleccionado Matlab para hacer la simulación del algoritmo, se realizó la trayectoria del brazo robótico haciendo las combinaciones de las funciones discretas presentadas. Se ha utilizado una interfaz gráfica de usuario tal y como se muestra en la Figura 4. Colocando los grados en los que se quiere mover cada junta y oprimiendo el push button llamado “Get position” el algoritmo realiza la trayectoria y trabaja simultáneamente con la transformación lineal presentada para obtener la cinemática inversa del brazo robótico y se muestra una representación del brazo robótico con las tres articulaciones haciendo la trayectoria. Además, se realiza el cálculo del número de combinaciones, tiempo límite superior, tiempo límite superior sin los tiempos de espera agregados por cada pulso de 10 milisegundos de la primera función discreta, tiempo límite inferior y los grados que se obtienen tomando el tiempo límite superior y el tiempo límite inferior.

La obtención de la trayectoria está inspirada en el método de polinomios cúbicos, en este caso, los grados en el tiempo límite inferior son calculados usando los polinomios cúbicos y tomando el tiempo límite inferior para aproximar los grados que se pueden obtener en caso de usar dicho límite. Se presenta en este apartado un ejemplo para cuando el motor α (primera articulación) se mueva a 45 grados. En resumen, el programa calcula los grados que corresponden al tiempo límite superior y el tiempo límite inferior con el método de polinomios cúbicos para comparar los resultados con los resultados del algoritmo propuesto.

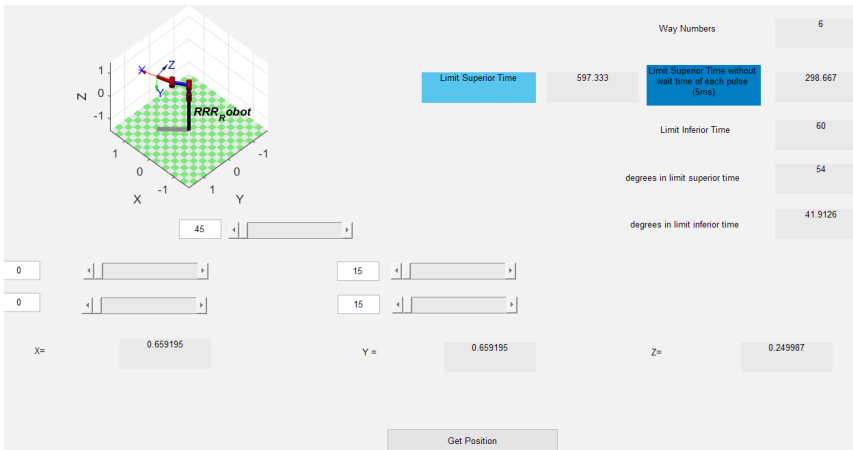


Figura 4 – Simulación del Algoritmo y aplicación de la transformación lineal para representar la trayectoria y la cinemática directa e inversa del brazo robótico.

El eje Z representa al eje α que se mencionó en la transformación lineal. En la interface se puede observar que la coordenada (x, y) igual a $(0.659195, 0.659195)$ debido a que $\alpha=45$ grados, y aunque en realidad se pide que en (x, y) se muevan las articulaciones en 15 grados, la transformación lineal ha convertido los 15 grados en $x = 1/12 = y$, después, al momento de transformar el vector (x, y, α) , se obtiene el nuevo vector $(kx, ky, k\alpha)$ y para este caso, $k = 8$ ya que para obtener k se debe igualar a la distancia sobre el eje X

o eje Y (en este caso las distancias de los dos eslabones se encuentran en el eje X) de los eslabones que se encuentran fuera del origen (0,0,0) y cada uno tiene de distancia igual a 4 unidades, entonces $k = 8$, también en este ejemplo, la k que multiplica a α se encuentra en el origen, entonces se dice que $k = 1$.

5. Resultados

Los resultados están enfocados en la simulación en MATLAB y en el código de Arduino generado mediante la simulación en MATLAB, se utilizó una pantalla táctil LCD modelo 2.4 TFT y motores paso a paso Nema 17, los resultados se presentan como los valores deseados de α obtenida al tocar la pantalla LCD comparando los con los resultados de la simulación en MATLAB y la ejecución del código de Arduino. Las Tablas 2, 3, 4 y 5 representan a los resultados obtenidos tocando la pantalla LCD para las 4 condiciones.

α	$\left(\frac{y}{x}\right)$	n	$f(n) = 1.8n$	s	$f(s) = \sum_{n=0}^s 1.8n$
5.71	0.1	3	5.4	2	5.4
11.3	0.2	6	10.8	3	10.8
16.69	0.3	9	16.2	3	10.8
18	0.35	12	21.6	4	18
21.8	0.4	12	21.6	4	18
26.56	0.5	14	25.2	4	18
27	0.5095	14	25.2	5	27
30.96	0.6	14	25.2	5	27
34.99	0.7	17	30.6	5	27
37.8	0.7756	19	34.2	6	38.7
38.65	0.8	21	37.8	6	38.7
41.98	0.9	23	41.4	6	38.7
45	1	25	45	6	38.7

Tabla 2 – número de pulsos que $f(s)$ y $f(n)$ necesitan para aproximarse a α en la primera condición.

Entonces, se tiene la siguiente función la cual representa la mejor opción para usarse en esta primera condición.

$$f(s,n) := \begin{cases} \sum_{n=0}^s 1.8n & \text{si } 0 \leq \left(\frac{y}{x}\right) \leq 0.35, \text{ con } s = \{1,2,3,4\}, \text{ si } 0.51 \leq \left(\frac{y}{x}\right) \leq 0.79, s = \{5,6\} \\ 1.8n & \text{si } 0.4 \leq \left(\frac{y}{x}\right) \leq 0.5, \text{ si } 0.6 \leq \left(\frac{y}{x}\right) \leq 0.7 \\ 1.8n & \text{si } 0.8 \leq \left(\frac{x}{y}\right) \leq 1 \end{cases} \quad (19)$$

α	$\left(\frac{y}{x}\right)$	n	$f(n) = 1.8n$	s	$f(s) = \sum_{n=0}^s 1.8n$
45	1	25	45	8	64.8
63.434	2	35	63	8	64.8
75.963	4	42	75.6	8	64.8
80.53	6	44	79.2	8	64.8
81	6.313	45	81	9	81
82.87	8	46	82.8	9	81
84.298	10	47	84.6	9	81
87.137	20	48	86.4	9	81
88.56	40	49	88.2	9	81
88.85	50	49	88.2	9	81
89.427	100	49	88.2	9	81
89.93	900	49	88.2	9	81

Tabla 3 – número de pulsos que $f(s)$ y $f(n)$ que necesitan para aproximarse a α en la segunda condición.

Entonces, la combinación de las funciones es representada como:

$$f(n,s) := \begin{cases} 1.8n & \text{si } 1 \leq \left(\frac{y}{x}\right) \leq 1.99 \\ \sum_{n=0}^s 1.8n & \text{si } 2 \leq \left(\frac{y}{x}\right) \leq 2.2 \text{ or si } 6 \leq \left(\frac{y}{x}\right) \leq 8 \\ 1.8n & \text{si } 2.3 \leq \left(\frac{y}{x}\right) \leq 5.9 \text{ o si } 8.1 \leq \left(\frac{y}{x}\right) \leq 1000 \end{cases} \quad (20)$$

α	$\left(\frac{y}{x}\right)$	n	$f(n) = 1.8n$	(s,n)	$f(s,n) = f(s) + f(n)$
91.145	-50	50	90	(9,0)	81
91.6	-30	51	91.8	(9,0)	81
92.29	-25	51	91.8	(9,0)	81
92.86	-20	52	93.6	(9,0)	81
93.814	-15	52	93.6	(9,0)	81
95.71	-10	53	95.4	(9,0)	81
96.34	-9	54	97.2	(10,0)	99
97.125	-8	54	97.2	(10,0)	99
98.13	-7	55	99	(10,0)	99
99	-6.313	55	99	(10,0)	99

α	$\left(\frac{y}{x}\right)$	n	$f(n) = 1.8n$	(s, n)	$f(s, n) = f(s) + f(n)$
99.462	-6	56	100.8	(10,1)	100.8
100.8	-5.242	56	100.8	(10,1)	100.8
101.309	-5	56	100.8	(10,1)	100.8
102.6	-4.0473	57	102.6	(10,2)	102.6
104.036	-4	58	104.4	(10,3)	104.4
108	-3.0077	60	108	(10,5)	108
108.434	-3	60	108	(10,5)	108
115.2	-2.125	64	115.2	(10,9)	115.2
116.565	-2	65	117	(10,10)	117
133.2	-1.064	74	133.2	(10,19)	133.2
135	-1	75	135	(10,20)	135

Tabla 4 – número de pulsos que $f(s)$ y $f(n)$ que necesitan para aproximarse a α en la tercera condición.

Se obtiene la siguiente función:

$$f(s, n) := \begin{cases} 1.8n & \text{si } -50 \leq \left(\frac{y}{x}\right) < -7 \\ \sum_{n=0}^s 1.8n + 1.8n & \text{si } -7 \leq \left(\frac{y}{x}\right) \leq -1 \text{ con } s = 10 \text{ y } 0 < n \leq 20 \end{cases} \quad (21)$$

α	$\left(\frac{y}{x}\right)$	n	$f(n) = 1.8n$	(s, n)	$f(s, n) = f(s) + f(n)$
-0.9999	135.01	75	135	(10,20)	135
-0.99	135.28	75	135	(10,20)	135
-0.9	138.012	77	138.6	(10,20)	135
-0.85	139.012	78	140.4	(10,20)	135
-0.8	141.34	79	142.2	(10,20)	135
-0.75	143.036	80	144	(10,20)	135
-0.7	145.007	81	145.8	(10,20)	135
-0.6	149.036	83	149.4	(10,20)	135
-0.5	153.43	85	153	(10,20)	135
-0.4	158.198	87	156.6	(10,20)	135
-0.3	163.3	90	162	(10,20)	135
-0.2905	163.8	91	163.8	(13,0)	168.8
-0.2	168.69	94	169.2	(13,3)	169.2
-0.158	171	95	171	(13,4)	171

α	$\left(\frac{y}{x}\right)$	n	$f(n) = 1.8n$	(s, n)	$f(s, n) = f(s) + f(n)$
-0.1	174.28	97	174.6	(13,6)	174.6
-0.031	178.2	99	178.2	(13,8)	178.2
-1	180	100	180	(13,9)	180

Tabla 5 – número de pulsos que $f(s)$ y $f(n)$ que necesitan para aproximarse a α en la cuarta condición.

Los resultados de $f(s)$ la función son obtenidos mezclando $f(s)$ y $f(n)$ es muy preciso y mas rapido que solo usando $f(s)$ o $f(n)$, en este caso $f(s, n)$ se mantiene en 135 mientras (y/x) es menor que -0.2905. La Figura 5, se muestra el resultado de cada función y la aproximación de la función $f(n)$ y $f(s)$, se puede observar como la combinaciones pueden ser construidas y cuales pueden ser usadas. Por último se obtiene la función para la cuarta condición:

$$f(s, n) := \begin{cases} 1.8n & \text{si } 0.999 \leq \left(\frac{y}{x}\right) < -0.3 \\ \sum_{n=0}^s 1.8n & \text{si } -0.29 \leq \left(\frac{y}{x}\right) \leq -0.25 \\ \sum_{n=0}^s 1.8n + 1.8n' & \text{si } -0.25 < \left(\frac{y}{x}\right) \leq -1 \end{cases} \quad \text{con } s = 13 \text{ y } 0 < n' \leq 9 \quad (22)$$

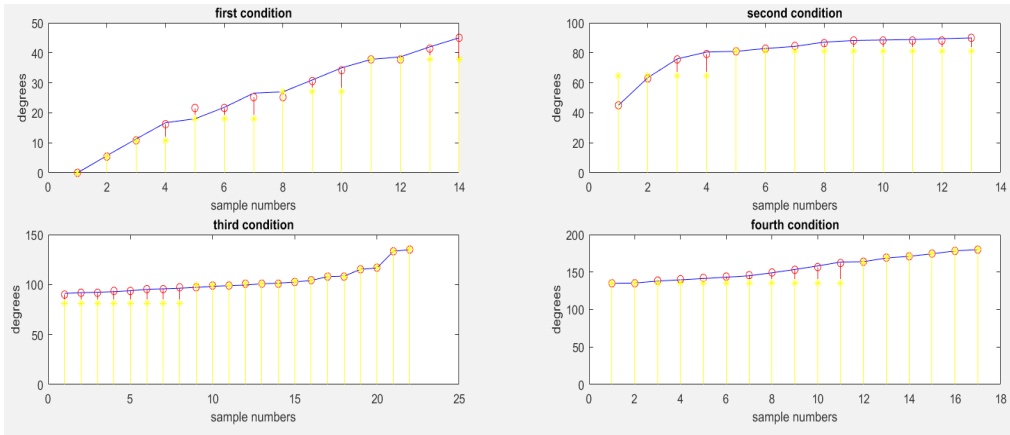


Figura 5 – Resultados de la trayectoria en cada condición en Arduino y en Matlab.

6. Conclusiones

Se ha analizado un método para mejorar el control de un brazo robótico antropomórfico donde se puede calcular fácilmente su cinemática inversa o cinemática directa, además, se puede generar una trayectoria combinando las dos funciones discretas, esto es lo más interesante, debido a que de esta forma se puede utilizar una pantalla táctil para que un paciente con discapacidades/dificultades físicas pueda controlar un brazo robótico tan solo tocar con un solo dedo la coordenada de la pantalla táctil LCD. Entonces, usando un espacio con dos dimensiones, es posible generar una trayectoria en un espacio con tres dimensiones, como se mostró anteriormente, esto se lleva a cabo con una transformación lineal, la cual obtiene una coordenada en el espacio con dos dimensiones, después, se selecciona la condición donde se posiciona coordenada, y de esta forma, saber cuántos grados se debe mover cada articulación y la posición final en el eje X , eje Y y eje α , la cual se representa como el vector de llegada de la transformación lineal. Después, se usa el algoritmo propuesto para traducir los datos que la transformación lineal ha proporcionado, entonces, se genera la trayectoria del brazo robótico moviendo motores paso a paso de cada articulación, sin embargo, el brazo robótico puede tardar como máximo 1.5 segundos, esto hace que los movimientos del brazo robótico sean muy rápidos.

Sería interesante controlar el tiempo de la trayectoria del brazo robótico, sin embargo, con este método no es posible, entonces, para futuros trabajos, se busca encontrar una función que genere la trayectoria del brazo robótico controlando el tiempo en el que el brazo robótico llega a su partida y que además pueda ser fácil de implementar usando diferentes motores y reduciendo el número de combinaciones. Se espera que con este algoritmo y la transformación lineal sea más fácil diseñar una función en la que se pueda utilizar cualquier pantalla táctil LCD y usando cualquier tipo de motor en las articulaciones del brazo robótico antropomórfico para generar la trayectoria. Además, al encontrar una nueva función, será posible que sea mucho más fácil de programarse, más rápido de ejecutar el programa, mejorar el consumo de memoria, y por lo tanto, la ejecución será más fácil y rápida.

Referencias

- Allen, S. (2016). New Prostheses and Orthoses Step Up their Game: Motorized Knees, Robotic Hands, and Exosuits Mark Advances in Rehabilitation Technology. *IEEE Pulse*, 7(3), 6-11.
- Atre, P., Bhagat, S., Pooniwalla, N., & Shah, P. (2018). Efficient and Feasible Gesture Controlled Robotic Arm. In: *Proceeding of Second International Conference on Intelligent Computing and Control Systems*, (pp. 1–6), Madurai, India: IEEE Publishing.
- Badrinath, A.S., Vinay, P.B., & Hegde, P. (2016). Computer Vision based semi-intuitive Robotic arm. In: *Proceeding of 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics*, pp. 563-567. Chennai, India: IEEE Publishing.

- Beckerle, P., Kõiva, R., Kirchner, E. A., Bekrater-Bodmann, R., Dosen, S., Christ, O., Abbink, D. A., Castellini, C., & Lenggenger, B. (2018). Feel-Good Robotics: Requirements on Touch for Embodiment in Assistive Robotics. *Frontiers in Neurorobotics*, 12, 1-84.
- Bovet, S., Debarba, H. G., Herbelin, B., Molla, E., & Boulic, R. (2018). The Critical Role of Self-Contact for Embodiment in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 24(4), 1428-1436.
- Burgner-Kahrs, J., Rucker, D.C., & Choset, H. (2015). Continuum Robots for Medical Applications: A Survey. *IEEE Transactions on Robotics and Automation*, 31(6), 1261-1280.
- Chung, C.S., Wang, H., & Cooper, R.A. (2013). Functional assessment and performance evaluation for assistive robotic manipulators: Literature review. *The Journal of Spinal Cord Medicine*, 36(4), 273-289.
- Fernández-Ordoñez, J., Maza, L., Torres-Carrión, P., Barba-Guzmán, L., & Rodríguez-Morales, G. (2019). Experiencia Afectiva Usuario en ambientes con Inteligencia Artificial, Sensores Biométricos y/o Recursos Digitales Accesibles: Una Revisión Sistemática de Literatura. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (35), 35-53. DOI: 10.17013/risti.35.35-53.
- Grau, A., Indri, M., Bello, L.L., & Sauter, T. (2017). Industrial robotics in factory automation: From the early stage to the Internet of Things. In: *Proceeding of 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 6159-6164. Beijing, China: IEEE publishing.
- Jiang, H., Wachs, J.P., & Duerstock, B.S. (2012). Facilitated Gesture Recognition Based Interfaces for People with Upper Extremity Physical Impairments. In: L. Alvarez, M. Mejail, L. Gomez, J. Jacobo (eds.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 228-235. Berlin, Heidelberg: Springer.
- Kokkinara, E., Slater, M., & López-Moliner, J. (2015). The effects of visuomotor calibration to the perceived space and body through embodiment in immersive virtual reality. *ACM Transactions on Applied Perception*, 13(1), 1-22.
- Levin, M.F., Weiss, P.L., & Keshner, E.A. (2015). Emergence of virtual reality as a tool for upper limb rehabilitation: incorporation of motor control and motor learning principles. *Physical Therapy*, 95(3), 415-425.
- Kruthika, K., Kumar, B.M.K., & Lakshminarayanan, S. (2016). Design and development of a robotic arm. In: *Proceeding of International Conference on Circuits, Controls, Communications and Computing*, pp. 1-4. Bangalore, India: IEEE publishing.
- Lara, G., Santana, A., Lira, A., & Peña, A. (2019). El Desarrollo del Hardware para la Realidad Virtual. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (31), 106-117. DOI: 10.17013/risti.31.106-117.

- Makin, T., de Vignemont, F., & Faisal, A. (2017). Neurocognitive barriers to the embodiment of technology. *Nature Biomedical Engineering*, 1(0014), 1-3.
- Murali, A., Sen, S., Kehoe, B., Garg, A., Mcfarland, S., Patil, S., Goldberg, K. (2015). Learning by observation for surgical subtasks: Multilateral cutting of 3D viscoelastic and 2D Orthotropic Tissue Phantoms. In: *Proceeding of International Conference on Robotics and Automation*, pp. 1202-1209. Seattle, WA, USA: IEEE Publishing.
- Niyetkaliyev, A.S., Hussain, S., Ghayesh, M.H., & Alici, G. (2017). Review on Design and Control Aspects of Robotic Shoulder Rehabilitation Orthoses. *IEEE Transactions on Human-Machine Systems*, 47(6), 1134-1145
- Perez-Marcos, D., Chevalley, O., Schmidlin, T., Garipelli, G., Serino, A., Vuadens, P., Tadi, T., Blanke, O., & Millán, J.D. (2017). Increasing upper limb training intensity in chronic stroke using embodied virtual reality: a pilot study. *Journal of NeuroEngineering and Rehabilitation*, 14(1), pp. 119.
- Proietti, T., Crocher, V., Roby-Brami, A., & Jarrassé, N. (2016). Upper-Limb Robotic Exoskeletons for Neurorehabilitation: A Review on Control Strategies. *IEEE Reviews in Biomedical Engineering*, 9, 4-14,
- Rehmat, N., Zuo, J., Meng, W., Liu, Q., Xie, S.Q., & Liang, H. (2018). Upper limb rehabilitation using robotic exoskeleton systems: a systematic review. *International Journal of Intelligent Robotics and Applications*, 2(3), 283-295.
- Shademan, A., Decker, R.S., Opfermann, J.D., Leonard, S., Krieger, A., & Kim, P.C. (2016). Supervised autonomous robotic soft tissue surgery. *Science Translational Medicine*, 8(337), 337ra64.
- Yenorkar, R., & Chaskar, U. M. (2018). GUI Based Pick and Place Robotic Arm for Multipurpose Industrial Applications. In: *Proceeding of the Second International Conference on Intelligent Computing and Control Systems*, (pp. 200-203). Madurai, India: IEEE publishing.
- Young, A.J., & Ferris, D.P. (2017). State of the Art and Future Directions for Lower Limb Robotic Exoskeletons. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(2), 171-182.

Una revisión sistemática de la literatura enfocada al uso de vehículos aéreos no tripulados durante el proceso de detección de vegetación

Enrique Ponce-Corona¹, María Guadalupe Sánchez¹, Daniel Fajardo-Delgado¹,
Brenda Acevedo-Juárez², Miguel De-la-Torre³, Himer Avila-George³, Wilson Castro⁴

**eponceo8@gmail.com, msanchez@itcg.edu.mx, dfajardo@itcg.edu.mx,
brenda.acevedo@academicos.udg.mx, miguel.dgomora@academicos.udg.mx,
himer.avila@academicos.udg.mx, wcastro@unf.edu.pe**

¹ Departamento de Sistemas y Computación, TecNM - Instituto Tecnológico de Ciudad Guzmán, Ciudad Guzmán 49100, Jalisco, México.

² Departamento de Ciencias Naturales y Exactas, Universidad de Guadalajara, Ameca 46600, Jalisco, México.

³ Departamento de Ciencias Computacionales e Ingenierías, Universidad de Guadalajara, Ameca 46600, Jalisco, México.

⁴ Facultad de Ingeniería de Industrias Alimentarias, Universidad Nacional de Frontera, Sullana 20100, Perú.

DOI: 10.17013/risti.36.82-101

Resumen: En los últimos años, se ha popularizado el uso de vehículos aéreos no tripulados en el desarrollo de aplicaciones agrícolas y forestales. Debido a la importancia de los cultivos tanto en entornos rurales como urbanos, la extracción de información a partir de datos obtenidos por sensores remotos es un área de investigación relevante. Dicha información es fundamental para realizar la planeación urbana y rural, establecer de nichos ecológicos sustentables, estimar el rendimiento de cultivos, realizar inventarios forestales, supervisar los sistemas de gestión agrícola y algunas otras aplicaciones agrícolas. En este artículo, se presenta una revisión sistemática de la literatura referente al uso de los vehículos aéreos no tripulados en aplicaciones relacionadas con la detección de vegetación y el inventario de plantas. Asimismo, se realiza un análisis de las principales técnicas computacionales y estadísticas utilizadas para el procesamiento de las imágenes tomadas desde vehículos aéreos no tripulados.

Palabras-clave: Vehículos aéreos no tripulados; vegetación; conteo; inventario; procesamiento de imágenes.

A systematic review of the literature focused on the use of unmanned aerial vehicles during the vegetation detection process

Abstract: In recent years, the use of unmanned aerial vehicles has become popular in the development of agricultural and forestry applications. Due to the importance of plants in both rural and urban environments, the extraction of information from

remote sensing data is a relevant research area; such information is fundamental for urban and rural planning, establishing sustainable ecological niches, estimating crop yields, conducting forest inventories, monitoring agricultural management systems and some other agricultural applications. In this article, a systematic review of the literature concerning the use of unmanned aerial vehicles in applications related to vegetation detection and plant inventory is presented. Likewise, an analysis of the main computational and statistical techniques used for processing unmanned aerial vehicles images is also made.

Keywords: Unmanned aerial vehicles; vegetation; counting; inventory; image processing; algorithms.

1. Introducción

Actualmente, la agricultura en todo el mundo tiene como objetivo garantizar la seguridad alimentaria presente y futura. Esto, motiva a investigadores, tecnólogos y agricultores a buscar y desarrollar nueva tecnología que ayude a optimizar los recursos en la producción de alimentos.

De acuerdo con González *et al.* (2016), la agricultura de precisión se basa en el manejo específico de una zona de cultivo. Para ello, se utilizan herramientas tecnológicas como el posicionamiento global (GPS), dispositivos de distribución de riego, fertilizantes y plaguicidas variables, redes de sensores y, recientemente, vehículos aéreos no tripulados (VANT). Estas herramientas en conjunto realizan el proceso de recolección de la información, la cual es finalmente plasmada en mapas digitales sobre los cuales se toman decisiones sobre el manejo agronómico de los cultivos. El objetivo principal de la agricultura de precisión es la obtención de mayores rendimientos económicos, sociales y medioambientales. Dicho enfoque se ha extendido en los últimos años como un método eficaz para obtener mayores rendimientos en la producción agrícola y optimizar el uso de recursos en dicho proceso.

Los VANT son aeronaves que vuelan autónomamente hasta una altura de 120 m usando un sistema de piloto automático. Los vuelos son monitoreados desde un centro de control en tierra. Según Escalante *et al.* (2016), fue el ejército estadounidense, en la década de los cincuenta, el primero en utilizar los VANT en aplicaciones militares relacionadas con tareas de reconocimiento, vigilancia, y mapeo.

En los últimos años, se ha popularizado el uso de VANT para el desarrollo de aplicaciones civiles en el área de percepción remota. Lo anterior se ha debido a los recientes avances tecnológicos en el desarrollo de sistemas de control de vuelo y navegación, así como en las mejoras en las propiedades de las cámaras. En este contexto, las imágenes tomadas desde un VANT se han incluido en varios estudios de percepción remota, principalmente en aplicaciones agrícolas y forestales (Koc-San *et al.*, 2018).

Entre las principales aplicaciones en la agricultura se encuentran la detección, identificación, conteo de cultivos, e inventarios de árboles, bosques, hierbas y coníferas. El conteo de árboles, plantas o cultivos es una actividad importante para los agricultores en la toma de decisiones. De acuerdo con She *et al.* (2014), dicha actividad por lo regular se realiza de manera manual consumiendo tiempo y recursos, lo que obliga a los productores a sobreestimar los recuentos de plantas, árboles o cultivos, a realizarlos con

poca frecuencia y tener censos inexactos; además, de no tener acceso a un inventario en tiempo real. Esta problemática puede ser atendida a través del análisis de imágenes de alta resolución. Sin embargo, las imágenes de alta resolución temporal son difíciles y costosas de obtener, ya sea por imágenes satelitales o usando aeronaves (Grenzdörffer *et al.*, 2008).

La aparición de sistemas VANT cambió en gran medida el modo de trabajo de los sistemas convencionales y proporcionó a los productores de cultivos una plataforma área estable y de bajo costo para la percepción remota (She *et al.*, 2014). Los sensores con los que cuenta los VANT pueden proporcionar imágenes de áreas que resuelven las principales deficiencias que actualmente tienen los sistemas de adquisición de imágenes (Hunt *et al.* 2010). Los VANT pueden usarse para observar las estructuras pequeñas, específicas y detalladas de los cultivos, a diferencia de otros sistemas; al respecto, Hadas *et al.*, (2019), mencionaron que la teledetección realizada con VANT permite realizar inventarios frecuentes de zonas agrícolas y obtener productos de gran calidad y alta resolución.

Este artículo tiene como objetivo establecer el estado actual sobre la detección de vegetación usando imágenes tomadas desde un VANT. Después de esta sección introductoria, el artículo se estructura de la siguiente manera: la sección 2 presenta una breve contextualización sobre los VANT y las aplicaciones agrícolas y forestales, la sección 3 describe el método de revisión sistemática realizado, la sección 4 presenta el análisis de los resultados, y finalmente, en la sección 5 se presentan las conclusiones y trabajos futuros.

2. Contextualización

Debido a la gran cantidad de plataformas aéreas consideradas como VANT, existen diversas maneras de clasificarlas, una de las más comunes es según el tipo de plataforma, es decir, ala fija o multirrotor. Para saber las ventajas que ofrece cada plataforma, recomendamos al lector revisar (Tahar & Ahmad, 2013).

De acuerdo con Wong (2001), el uso de VANT en actividades agrícolas data desde principio de la década de los 90, sin embargo, su popularización se ha dado en años recientes en aplicaciones relacionadas con el monitoreo de cultivos o áreas forestales.

Las imágenes adquiridas mediante fotogrametría VANT son una solución atractiva debido a su bajo costo de monitoreo ambiental y alta resolución espaciotemporal. Dichas imágenes generalmente son evaluadas y procesadas usando algún software específico (ej. Pix4DMapper, PhotoScan/Metashape, EnsoMOSAIC) el cual permite la generación de productos cartográficos tales como los ortomosaicos (ortofotos).

Un mosaico corresponde a un conjunto de imágenes que presentan áreas de traslape entre sí, las cuales son combinadas en una sola imagen para ampliar el rango de visión de la escena. Cuando el mosaico es corregido de las distorsiones causadas por el relieve del terreno y los objetos en el, se denomina ortomosaico.

Para extraer información desde un ortomosaico se utilizan desde técnicas relacionadas con la segmentación de las imágenes (ej. binarización, filtros, operadores morfológicos)

hasta técnicas de aprendizaje máquina (ej. redes neuronales artificiales, máquinas de soporte vectorial, árboles de decisión, entre otras).

3. Revisión sistemática de la literatura

En esta sección se ilustra el método que se siguió para realizar la presente revisión sistemática de la literatura. El método utilizado se basa en las recomendaciones hechas en Kitchenham & Charters (2007) y aplicado recientemente por Miramontes et al. (2016), Hernández et al. (2017), Muñoz et al. (2019), entre otros; dicho método consta de tres fases principales: planificación de la revisión, desarrollo de la revisión y reporte de resultados.

3.1. Planificación de la revisión

La planificación es la primera etapa de la revisión sistemática, en esta etapa se desarrolla el protocolo que guiará la revisión. En este trabajo se consideraron las siguientes actividades: 1) identificación de la necesidad de la revisión, 2) formulación las preguntas de investigación, 3) definición de la cadena de búsqueda y 4) selección de las fuentes de datos. Las cuales se describen brevemente a continuación.

Identificación de la necesidad para realizar la revisión sistemática

Con el abaratamiento de los VANT y los avances tecnológicos en sistemas de control de vuelo y navegación así como en el diseño de sensores más potentes, se ha popularizado el desarrollo de aplicaciones civiles basadas en el procesamiento de imágenes adquiridas desde VANT, principalmente en áreas relacionadas con el monitoreo de cultivos o áreas forestales. En este tipo de aplicaciones, la extracción de información a partir de las imágenes juega un rol fundamental, por lo tanto, en este trabajo se busca revisar qué técnicas de procesamiento de imágenes y extracción de objetos se han utilizado en las áreas de detección de vegetación, conteo de plantas e inventario de árboles.

Establecimiento de las preguntas de investigación

Se establecieron las siguientes preguntas de investigación.

- PIO1: ¿Cómo ha evolucionado el interés por usar VANT en la detección, identificación y conteo de plantas?
- PIO2: ¿Cuáles son las principales fuentes de información en el tema de conteo de plantas usando VANT?
- PIO3: ¿Cuáles son los principales problemas abordados con la detección de la vegetación usando imágenes tomadas desde VANT?
- PIO4: ¿Qué plantas han sido las más estudiadas usando imágenes adquiridas desde VANT?
- PIO5: ¿Qué técnicas son las más utilizadas para el conteo de plantas usando imágenes tomadas desde VANT?
- PIO6: ¿Qué tipos de imágenes se usan en estudios relacionados con la detección de plantas usando imágenes VANT?

Creación de la cadena de búsqueda

Para la creación de la cadena de búsqueda, se identificaron las palabras clave; las cuales fueron extraídas de las preguntas de investigación. Se decidió dividir el concepto plantas en: bosques, árboles, cultivos, y plantas. Los principales problemas identificados fueron: detección, identificación, conteo e inventario de plantas/árboles. Como resultado: Las palabras clave seleccionados fueron las siguientes: *Forest, Tree, Plant, Crop, Inventory, Detection, Counting, Identification, and "Unmanned Aerial Vehicle"*.

La cadena de búsqueda se obtuvo a partir del agrupamiento de las palabras clave y el uso de los operadores lógicos AND y OR; como resultado, se obtuvo la siguiente cadena:

(Forest OR Crop OR Plant OR Tree) AND (Detection OR Identification OR Counting OR Inventory) AND (UAV OR "Unmanned Aerial Vehicle")

Para el presente estudio, se entiende por *detección* la acción de descubrir un tipo de cultivo, árbol o planta que no era patente en una imagen; *identificación* es el establecimiento de la clasificación taxonómica de una planta; *conteo* es la acción de contar objetos en una imagen, dichos objetos están asociados a arboles, plantas, surcos, e incluso frutas; finalmente, *inventario* consiste en la elaboración de un catálogo que recoge la diversidad de especies de plantas existentes en un área determinada.

Identificación de las fuentes de datos

Atendiendo las recomendaciones propuestas por Kitchenham & Chartes (2007), sobre sus lecciones aprendidas para los procedimientos de búsqueda de datos; para este estudio, se decidió utilizar las siguientes fuentes de datos: (1) ACM Digital Library, (2) IEEE Xplore Digital Library, (3) ScienceDirect, Wiley y Scopus. Además, se decidió consultar MDPI dado su amplio contenido de acceso abierto en áreas relacionadas directamente con el presente estudio (percepción remota, sensores, y medioambiente).

3.2. Desarrollo de la revisión

El objetivo de esta segunda fase del método de revisión sistemática es identificar tantos estudios primarios como sea posible usando una estrategia de búsqueda imparcial; dichos estudios primarios permitirán responder a las preguntas de investigación. Para llevar a cabo el desarrollo de la presente revisión sistemática se deben realizar las siguientes dos actividades: (1) seleccionar los estudios primarios y (2) extraer y sintetizar la información.

Seleccionar los estudios primarios

Para garantizar que la estrategia de búsqueda de los estudios primarios sea imparcial, se definieron algunos criterios de inclusión y exclusión, y se siguió un procedimiento bien definido.

Criterios de inclusión. (CI01) estudios en idioma inglés; (CI02) estudios publicados durante el periodo comprendido entre enero del 2010 y diciembre de 2019; (CI03) estudios que contengan al menos dos palabras clave en el título; (CI04) estudios relacionados con la detección de plantas (inventario, conteo, identificación y detección).

Criterios de exclusión. (CE01) estudios duplicados; (CE02) estudios que no se basen en el uso de imágenes tomadas desde vehículos aéreos no tripulados.

Procedimiento. Para seleccionar los estudios primarios se realizaron los siguientes pasos:

1. Adaptar la cadena de búsqueda al motor de la fuente de datos seleccionada.
2. Aplicar los criterios de inclusión CIO1 y CIO2.
3. Aplicar el criterio de inclusión CIO3.
4. Leer resumen, introducción y conclusiones, y aplicar el criterio de inclusión CIO4 y los criterios de exclusión.

Según Wohlin (2014), el proceso de revisión sistemática puede ser ampliado usando un enfoque de búsqueda conocido como *snowballing*. El *snowballing* se refiere al uso de la lista de referencias de un documento para identificar documentos adicionales.

La Tabla 1, resume el número de trabajos de investigación publicados durante el periodo definido para el presente estudio (véase la columna 2), así como la selección de los estudios primarios (véase la columna 3). Durante el proceso de búsqueda, indexación, y clasificación varios estudios aparecieron duplicados. Para gestionar las referencias y eliminar los duplicados se utilizó la herramienta Mendeley. Tras aplicar el procedimiento definido para la selección de los estudios primarios se seleccionaron 88 estudios.

Fuentes de datos	Resultados de la cadena de búsqueda	Estudios primarios
ScienceDirect	91	25
IEEEExplore Digital Library	47	21
ACM Digital Library	2	1
Wiley	16	1
Scopus	79	20
MDPI	85	20
Total	320	88

Tabla 1 – Número de investigaciones publicadas y estudios primarios

Finalmente, se llevó a cabo una etapa de *snowballing* y se verificaron las referencias de los estudios seleccionados para no perder ningún documento relevante; como resultado, se agregaron 2 artículos más. Por lo tanto, se seleccionaron en total 90 artículos para responder a las preguntas de investigación. El Anexo A muestra el listado de los estudios primarios.

Extracción y síntesis de la información

Antes de realizar la extracción de la información, los estudios primarios fueron organizados por medio de la herramienta de gestión de archivos PDF Mendeley. Para llevar a cabo la presente actividad se diseñó una hoja de cálculo donde se recopilaban

los siguientes datos de cada estudio primario: título, autor, año, objetivo, tipo de vegetación, técnicas utilizadas, tipo de imagen, tipo de manuscrito, título de la fuente. Para extraer y analizar la información de todos los artículos, se siguieron los siguientes pasos: (1) La información fue extraída por el primer autor, y (2) por separado se verificó la información por los otros autores.

4. Análisis de resultados

En esta sección se muestran los resultados obtenidos de la revisión sistemática de la literatura, los cuales permiten conocer el panorama general de la detección de plantas usando imágenes adquiridas desde VANT. Los resultados se organizan por pregunta de investigación.

PI01: ¿Cómo ha evolucionado el interés por usar VANT en la detección, identificación y conteo de plantas?

Los estudios identificados se analizaron durante la última década para conocer la frecuencia y la evolución del número de publicaciones. La Fig. 1 muestra los resultados de este proceso de análisis, allí se puede observar que recientemente se han incrementado notablemente la cantidad de estudios relacionados con la detección de vegetación utilizando imágenes VANT.

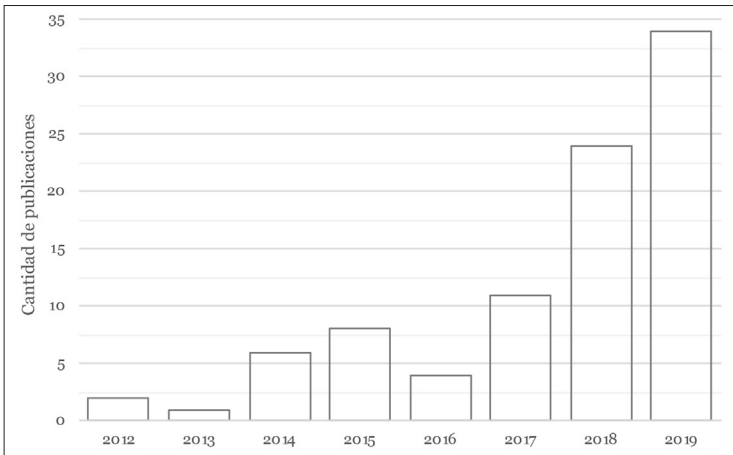


Figura 1 – Publicaciones por año

En lo que respecta a que países se encuentran desarrollando proyectos de investigación relacionadas con el uso de VANT como soporte a las actividades agrícolas y forestales destacan Brasil, China y España, ver la Fig. 2. En Brasil destacan sus estudios de cítricos y árboles frutales, en China sus estudios sobre coníferas, y en España el estudio de plantas de olivo.

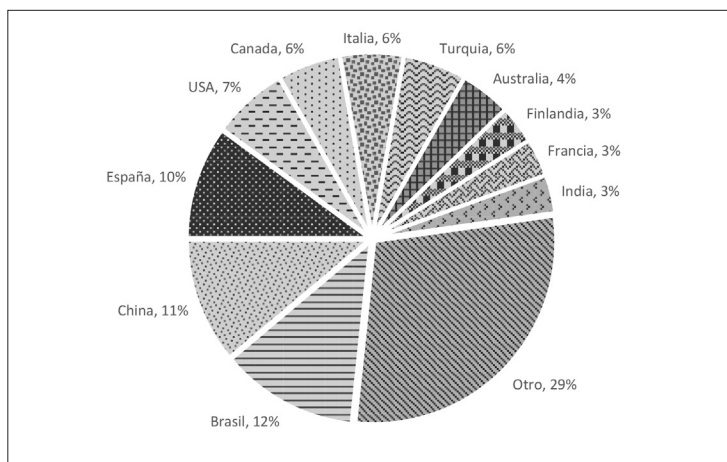


Figura 2 – Distribución de estudios primarios por país

PI02: ¿Cuáles son las principales fuentes de información en el tema de conteo de plantas usando VANT?

De los 90 estudios que componen el conjunto de estudios primarios de la presente revisión sistemática, 70 fueron publicados en revistas y 20 en conferencias. En la Fig. 3, se muestran las revistas más activas en el área de detección de plantas usando imágenes VANT. Destaca las revistas *Remote Sensing* de la editorial MDPI y *Computer and Electronics in Agriculture* de la editorial Elsevier, en dichas revistas se han publicado 17 y 11 de los estudios seleccionados en esta revisión respectivamente.

PI03: ¿Cuáles son los principales problemas abordados con la detección de la vegetación usando imágenes tomadas desde VANT?

La Fig. 4, muestra los tipos de problemas abordados mediante el procesamiento de imágenes adquiridas desde un VANT, los estudios se encuentran agrupados por el tipo de problema que solucionan (detección, identificación y conteo). Son 8 los estudios relacionados con problemas de conteo; 62 estudios se enfocan en la detección de plantas a nivel de bosques, plantaciones, surcos, árboles, arbustos, e incluso frutos; finalmente, 20 estudios se centran en la identificación de plantaciones, hierba y árboles.

PI04: ¿Qué plantas han sido las más estudiadas usando imágenes adquiridas desde VANT?

En la Fig. 5, se muestra un gráfico de pastel el cual proporciona la información referente al tipo de planta que se han estudiado, se observa que la detección de maleza es lo que se ha estudiado más usando imágenes VANT; los árboles de cítricos, palmeras y la vid también han sido ampliamente estudiados.

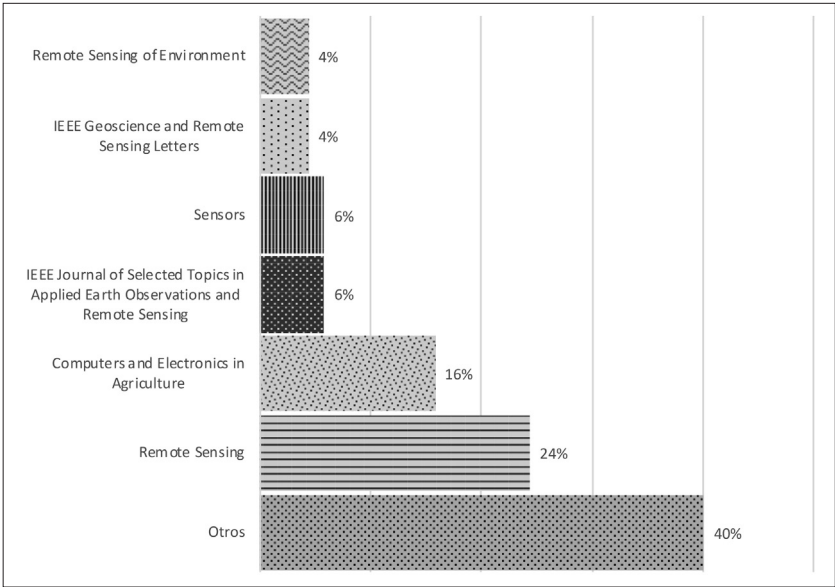


Figura 3 – Revistas en las cuales se ha publicado alguno de los estudios primarios

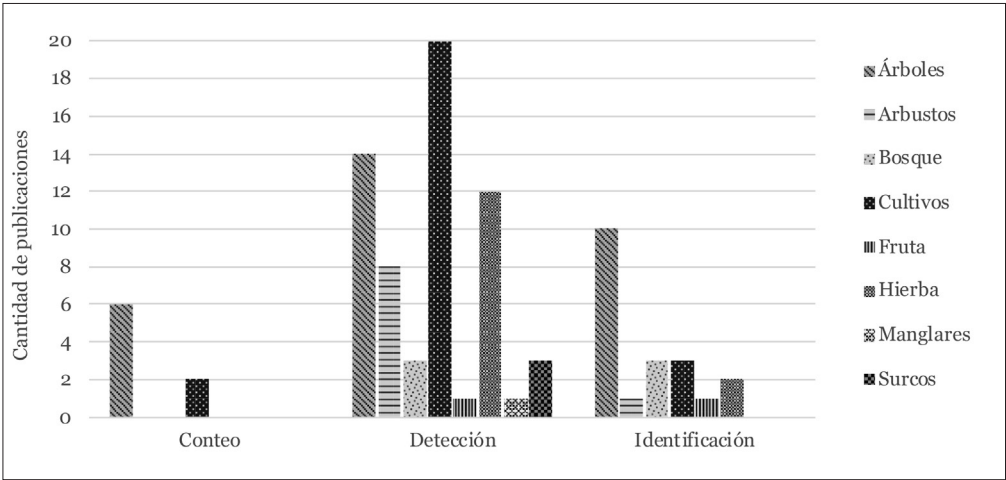


Figura 4 – Tipos de problemas abordados

PI05: ¿Qué técnicas son las más utilizadas para el conteo de plantas usando imágenes tomadas desde VANT?

Para atender el problema de conteo de plantas, se observó que la técnica más usada son las redes neuronales convolucionales (CNN, por sus siglas en inglés), en total 18

estudios utilizaron dicha técnica. Es de resaltar que el primer estudio que se encontró fue publicado en 2017, ver EP08. Otra técnica muy utilizada, son las máquinas de soporte vectorial (SVM, por sus siglas en inglés), dicha técnica fue utilizada en 7 estudios primarios. Las SVM, son una técnica madura de aprendizaje máquina y ampliamente utilizada en problemas de clasificación y regresión, por lo cual no es extraño que el primer estudio primario date del año 2012, véase el EP43; en dicho estudio se propone una SVM para detectar coronas de árboles.

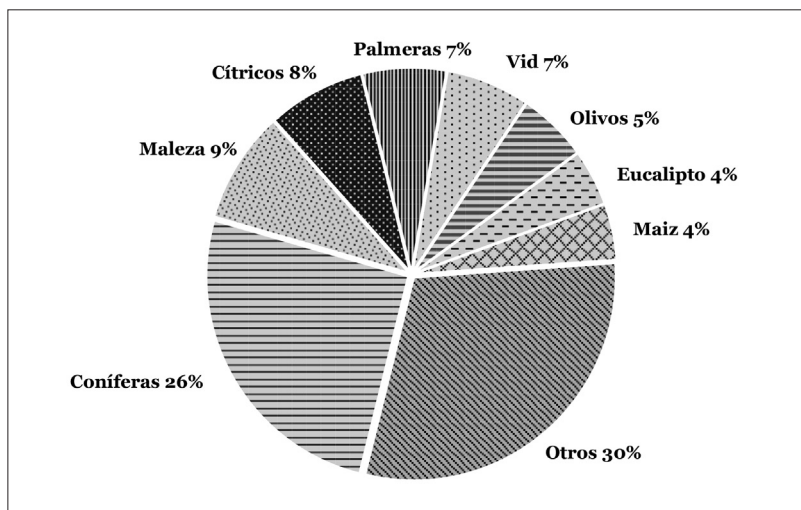


Figura 5 – Plantas estudiadas

Cabe mencionar que el software utilizado para procesar las imágenes es muy variado, desde lenguajes de programación de propósito general Matlab, Python y C++, hasta software especializado como TerraScan, Agisoft PhotoScan y Pix4D.

PI06: ¿Qué tipos de imágenes se usan en estudios relacionados con la detección de plantas usando imágenes VANT?

Los espacios de color juegan un papel fundamental en los estudios relacionados con el procesamiento de datos obtenidos desde sensores montados en VANT (De-la-Torre, 2019); entre los principales espacios de color se encuentran: RGB (por sus siglas en inglés *Red, Green, Blue*), HSV (por sus siglas en inglés *Hue, Saturation, Value*), y CIE Lab (también conocido como CIE $L^*a^*b^*$) el cual fue definido por la Comisión Internacional de Iluminación (CIE). En la Fig. 6, se muestra que el espacio de color más utilizado en los estudios primarios es el RGB, de hecho, más del 50% de las investigaciones lo han utilizado; sin embargo, se observan algunos trabajos que utilizan los espacios de color tribanda CIE Lab y HSV. Por otra parte, se encontró que un 14% de los estudios primarios utilizó imágenes multiespectrales y el 10% datos LiDAR. Cabe mencionar que los estudios que basados en imágenes multiespectrales utilizaron principalmente las cámaras Canon S110 NIR, Tetracam mini-MCA-6 y Parrot Sequoia.

Las imágenes hiperespectrales, a pesar de tener la ventaja de la gran cantidad de datos que contienen se han utilizado muy poco (2%), tal vez debido al alto coste de los sensores de este tipo.

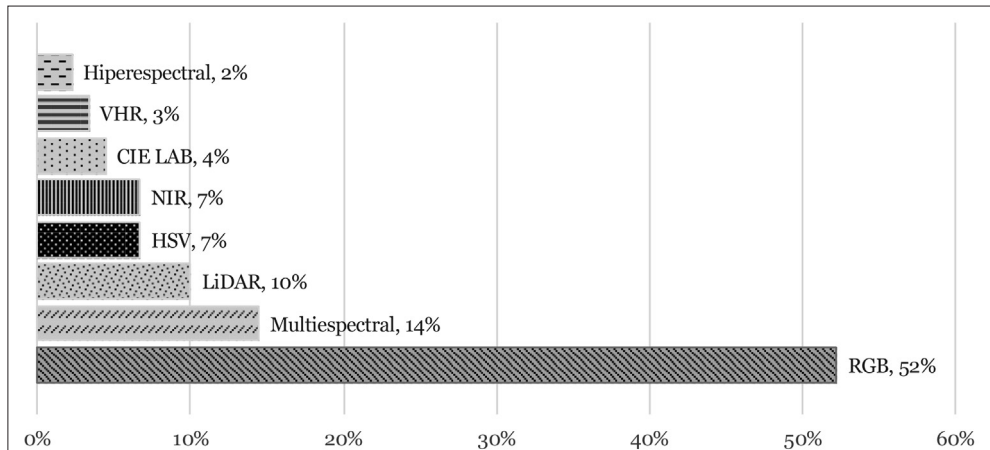


Figura 6 – Tipos de imágenes/datos utilizados

5. Conclusiones

La agricultura de precisión ha sido sustancialmente favorecida por las aplicaciones de la percepción remota en las últimas tres décadas. Sin embargo, la mayoría de los estudios se centran en la vigilancia de los cultivos. Se han hecho muchos esfuerzos para calcular parámetros como los índices de vegetación, rendimiento de los cultivos, la biomasa de las plantas, el estrés hídrico, entre otras.

En este trabajo de investigación, se realizó una revisión sistemática de la literatura referente al uso de VANT en aplicaciones agrícolas y forestales, en específico se abordó el tema de la detección/identificación/conteo de plantas usando imágenes adquiridas desde VANT.

Se observó, que desde el año 2012 dada la popularidad que han alcanzado los VANT y los avances tecnológicos en el desarrollo de sensores, se han venido desarrollando cada vez más aplicaciones relacionadas con la detección de plantas. Los estudios abarcan diferentes niveles de granularidad, desde la detección de bosques, parcelas, surcos, hasta la identificación de árboles, arbustos, hierba, (incluso frutas y espigas) pasando por el conteo de plantas y la realización de inventarios de árboles.

Brasil, China, España y Estados Unidos de América son los países donde se han desarrollado la mayoría de las investigaciones reportadas en la presente revisión sistemática. La mayoría de los estudios primarios (78%) fueron publicados en revistas científicas, siendo las revistas *Remote Sensing* de la editorial MDPI y la revista *Computers and Electronics in Agriculture* de la editorial Elsevier, las más activas en estos temas.

También se mostró que la mayoría de los estudios primarios estudiaron la maleza, los cítricos y los árboles de palmera; las técnicas de aprendizaje más populares son las CNN y SVM. Con respecto al software utilizado para procesar las imágenes se ha utilizado desde lenguajes de programación de propósito general Matlab, Python y C++, hasta software especializado como TerraScan, Agisoft PhotoScan y Pix4D. Con respecto al tipo de aeronave, mas del 74% de los VANT utilizados en los estudios primarios fueron vehículos multirrotor, destacando los de la marca DJI en particular el modelo Phantom 4. Con respecto a los sensores más utilizados la mayoría son del tipo RGB, multiespectral y LiDAR; con respecto a los estudios basados en cámaras multiespectrales se utilizaron principalmente los siguientes equipos: MicaSense, Canon S110 NIR, Tetracam y Parrot Sequoia.

Finalmente, se concluye que los VANT equipados con sensores basados en tecnología RGB, multiespectral y LiDAR son una excelente opción tecnológica para el desarrollo de aplicaciones agrícolas y forestales que ayuden a la obtención de mayores rendimientos económicos, sociales y medioambientales.

Referencias

- Escalante Torrado, J. O., Aceres Jiménez, J. J., & Porras Díaz, H. (2016). Ortomosaicos y modelos digitales de elevación generados a partir de imágenes tomadas con sistemas UAV. *Tecnura*, 20(50), 119-140.
- De-la-Torre, M., Zatarain, O., Avila-George, H., Muñoz, M., Oblitas, J., Lozada, R., Mejía, J., & Castro, W. (2019). Multivariate Analysis and Machine Learning for Ripeness Classification of Cape Gooseberry Fruits. *Processes*, 7(12), 928.
- González, A., Amarillo, G., Amarillo, M., & Sarmiento, F. (2016). Drones aplicados a la agricultura de precisión. *Revista especializada en ingeniería*, 10, 23-37.
- Grenzdörffer, G. J., Engel, A., & Teichert, B. (2008). The photogrammetric potential of low-cost UAVs in forestry and agriculture. *The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, 31(B3), 1207-1214.
- Hadas, E., Jozkow, G., Walicka, A., & Borkowski, A. (2019). Apple orchard inventory with a LiDAR equipped unmanned aerial system. *International Journal of Applied Earth Observation and Geoinformation*, 82, 101911.
- Hernández, L., Muñoz, M., Mejía, J., Peña, A., Rangel, N., & Torres, C. (2017). Una revisión sistemática de la literatura enfocada en el uso de gamificación en equipos de trabajo en la ingeniería de software. *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação*, (21), 33-50.
- Hunt, E. R., Hively, W. D., Fujikawa, S. J., Linden, D. S., Daughtry, C. S., & McCarty, G. W. (2010). Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sensing*, 2(1), 290-305.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. *Keele University and Durham University*, Technical report, ver. 2.3 EBSE.

- Koc-San, D., Selim, S., Aslan, N., & San, B. T. (2018). Automatic citrus tree extraction from UAV images and digital surface models using circular Hough transform. *Computers and electronics in agriculture*, 150, 289-301.
- Miramontes, J., Muñoz, M., Calvo-Manzano, J. A., & Corona, B. (2016). Establecimiento del estado del arte sobre el aligeramiento de procesos de software. *RISTI-Revista Ibérica de Sistemas e Tecnologías de Informação*, (17), 16-25.
- Muñoz, M., Peralta, M., & Laporte, C. Y. (2019). Análisis de las debilidades que presentan las Entidades Muy Pequeñas al implementar el estándar ISO/IEC 29110: Una comparativa entre estado del arte y el estado de la práctica. *RISTI-Revista Ibérica de Sistemas e Tecnologías de Informação*, (34), 85-96.
- She, T., Ehsani, R., Robbins, J., Leiva, J. N., & Owen, J. (2014, July). Applications of small UAV systems for tree and nursery inventory management. In: *Proceedings of the 12th International Conference on Precision Agriculture*, Sacramento, CA, USA (pp. 20-23).
- Tahar, K. N., & Ahmad, A. (2013). An evaluation on fixed wing and multi-rotor UAV images using photogrammetric image processing. *International Journal of Computer and Information Engineering*, 7(1), 48-52.
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, London, England (pp. 1-10).
- Wong, K. C. (2001). Survey of regional developments: Civil applications. In: *UAV Australia Conference*, Melbourne, Australia (pp. 8-9).

Anexo A: Estudios primarios

- (EP01) T. Adão, L. Pádua, J. Hruška, P. Marques, E. Peres, and J. J. Sousa, "A pilot digital image processing approach for detecting vineyard parcels in Douro region through high-resolution aerial imagery," *Proceedings of the International Conference on Geoinformatics and Data Analysis - ICGDA '18*. 2018.
- (EP02) Y. Bazi, S. Malek, N. Alajlan, and H. AlHichri, "An automatic approach for palm tree counting in UAV images," *2014 IEEE Geoscience and Remote Sensing Symposium*. 2014.
- (EP03) S. Malek, Y. Bazi, N. Alajlan, H. AlHichri, and F. Melgani, "Efficient Framework for Palm Tree Detection in UAV Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 12. pp. 4692–4703, 2014.
- (EP04) L. Wallace, A. Lucieer, and C. S. Watson, "Evaluating Tree Detection and Segmentation Routines on Very High Resolution UAV LiDAR Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12. pp. 7619–7628, 2014.
- (EP05) J. R. Souza et al., "Automatic detection of Ceratocystis wilt in Eucalyptus crops from aerial images," *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015.

- (EPo6) O. Guldogan et al., “Automated tree detection and density calculation using unmanned aerial vehicles,” 2016 Visual Communications and Image Processing (VCIP). 2016.
- (EPo7) G. L. A. Carrijo, D. E. Oliveira, G. A. de Assis, M. G. Carneiro, V. C. Guizilini, and J. R. Souza, “Automatic detection of fruits in coffee crops from aerial images,” 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR). 2017.
- (EPo8) J. Ribera, Y. Chen, C. Boomsma, and E. J. Delp, “Counting plants using deep learning,” 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP). 2017.
- (EPo9) G. V. Nardari et al., “Crop Anomaly Identification with Color Filters and Convolutional Neural Networks,” 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). 2018.
- (EP10) H. C. Oliveira, V. C. Guizilini, I. P. Nunes, and J. R. Souza, “Failure Detection in Row Crops From UAV Images Using Morphological Operators,” IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 7. pp. 991–995, 2018.
- (EP11) H. Huang, X. Li, and C. Chen, “Individual Tree Crown Detection and Delineation From Very-High-Resolution UAV Images Based on Bias Field and Marker-Controlled Watershed Segmentation Algorithms,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 7. pp. 2253–2262, 2018.
- (EP12) Z. Fan, J. Lu, M. Gong, H. Xie, and E. D. Goodman, “Automatic Tobacco Plant Detection in UAV Images via Deep Neural Networks,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 3. pp. 876–887, 2018.
- (EP13) A. P et al., “CNN Based Technique for Automatic Tree Counting Using Very High Resolution Data,” 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C). 2018.
- (EP14) A. O. Ok and A. Ozdarici-Ok, “Combining Orientation Symmetry and LM Cues for the Detection of Citrus Trees in Orchards From a Digital Surface Model,” IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 12. pp. 1817–1821, 2018.
- (EP15) I. Ahmed et al., “Automatic Detection and Segmentation of Lentil Crop Breeding Plots From Multi-Spectral Images Captured by UAV-Mounted Camera,” 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). 2019.
- (EP16) Bouachir, W., Ihou, K. E., Gueziri, H. E., Bouguila, N., & Bélanger, N. (2019). “Computer Vision System for Automatic Counting of Planting Microsites Using UAV Imagery”. IEEE Access, 7, 82491-82500.
- (EP17) Kitano, B. T., Mendes, C. C., Geus, A. R., Oliveira, H. C., & Souza, J. R. (2019). “Corn Plant Counting Using Deep Learning and UAV Images”. IEEE Geoscience and Remote Sensing Letters.

- (EP18) da Silva Vieira, G., Rocha, B. M., Soares, F., Lima, J. C., Pedrini, H., Costa, R., & Ferreira, J. (2019, November). "Extending the Aerial Image Analysis from the Detection of Tree Crowns". In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 1681-1685). IEEE.
- (EP19) Yancho, J. M. M., Coops, N. C., Tompalski, P., Goodbody, T. R., & Plowright, A. (2019). "Fine-Scale Spatial and Spectral Clustering of UAV-Acquired Digital Aerial Photogrammetric (DAP) Point Clouds for Individual Tree Crown Detection and Segmentation". IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12(10), 4131-4148.
- (EP20) Windrim, L., & Bryson, M. (2018). "Forest Tree Detection and Segmentation using High Resolution Airborne LiDAR". IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3898-3905).
- (EP21) Pignatti, S., Casa, R., Harfouche, A., Huang, W., Palombo, A., & Pascucci, S. (2019, July). "Maize Crop and Weeds Species Detection by Using UAV VNIR Hyperpectral Data". In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium (pp. 7235-7238). IEEE.
- (EP22) Morales, V. A., Galvis, J. R., Garcia, E. G., & Salcedo, I. L. (2019, July). "Estimation of Individual Potato Plants Area and Volume From UAV-Based Multispectral Images". In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium (pp. 6259-6262). IEEE.
- (EP23) L. Wallace, A. Lucieer, C. Watson, and D. Turner, "Development of a UAV-LiDAR System with Application to Forest Inventory," Remote Sensing, vol. 4, no. 6. pp. 1519–1543, 2012.
- (EP24) J. M. Peña, J. Torres-Sánchez, A. Serrano-Pérez, A. I. de Castro, and F. López-Granados, "Quantifying efficacy and limits of unmanned aerial vehicle (UAV) technology for weed seedling detection as affected by sensor resolution," Sensors , vol. 15, no. 3, pp. 5609–5626, Mar. 2015.
- (EP25) S. Puliti, H. Ørka, T. Gobakken, and E. Næsset, "Inventory of Small Forest Areas Using an Unmanned Aerial System," Remote Sensing, vol. 7, no. 8. pp. 9632–9654, 2015.
- (EP26) C. Poblete-Echeverría, G. Olmedo, B. Ingram, and M. Bardeen, "Detection and Segmentation of Vine Canopy in Ultra-High Spatial Resolution RGB Imagery Obtained from Unmanned Aerial Vehicle (UAV): A Case Study in a Commercial Vineyard," Remote Sensing, vol. 9, no. 3. p. 268, 2017.
- (EP27) H. Jiang, S. Chen, D. Li, C. Wang, and J. Yang, "Papaya Tree Detection with UAV Images Using a GPU-Accelerated Scale-Space Filtering Method," Remote Sensing, vol. 9, no. 7. p. 721, 2017.
- (EP28) O. Nevalainen et al., "Individual Tree Detection and Classification with UAV-Based Photogrammetric Point Clouds and Hyperspectral Imaging," Remote Sensing, vol. 9, no. 3. p. 185, 2017.

- (EP29) J. Albetis et al., "Detection of Flavescence dorée Grapevine Disease Using Unmanned Aerial Vehicle (UAV) Multispectral Imagery," *Remote Sensing*, vol. 9, no. 4. p. 308, 2017.
- (EP30) T. K. Alexandridis et al., "Novelty Detection Classifiers in Weed Mapping: *Silybum marianum* Detection on UAV Multispectral Images," *Sensors*, vol. 17, no. 9, Sep. 2017.
- (EP31) M. Mohan et al., "Individual Tree Detection from Unmanned Aerial Vehicle (UAV) Derived Canopy Height Model in an Open Canopy Mixed Conifer Forest," *Forests*, vol. 8, no. 9. p. 340, 2017.
- (EP32) O. Csillik, J. Cherbini, R. Johnson, A. Lyons, and M. Kelly, "Identification of Citrus Trees from Unmanned Aerial Vehicle Imagery Using Convolutional Neural Networks," *Drones*, vol. 2, no. 4. p. 39, 2018.
- (EP33) M. Hassanein, Z. Lari, and N. El-Sheimy, "A New Vegetation Segmentation Approach for Cropped Fields Based on Threshold Detection from Hue Histograms," *Sensors*, vol. 18, no. 4, Apr. 2018.
- (EP34) L. Han et al., "Quantitative Identification of Maize Lodging-Causing Feature Factors Using Unmanned Aerial Vehicle Images and a Nomogram Computation," *Remote Sensing*, vol. 10, no. 10. p. 1528, 2018.
- (EP35) J. Yeom, J. Jung, A. Chang, M. Maeda, and J. Landivar, "Automated Open Cotton Boll Detection for Yield Estimation Using Unmanned Aircraft Vehicle (UAV) Data," *Remote Sensing*, vol. 10, no. 12. p. 1895, 2018.
- (EP36) X. Wu, X. Shen, L. Cao, G. Wang, and F. Cao, "Assessment of Individual Tree Detection and Canopy Cover Estimation using Unmanned Aerial Vehicle based Light Detection and Ranging (UAV-LiDAR) Data in Planted Forests," *Remote Sensing*, vol. 11, no. 8. p. 908, 2019.
- (EP37) J. Wu, G. Yang, X. Yang, B. Xu, L. Han, and Y. Zhu, "Automatic Counting of in situ Rice Seedlings from UAV Images Based on a Deep Fully Convolutional Neural Network," *Remote Sensing*, vol. 11, no. 6. p. 691, 2019.
- (EP38) E. Salamí, A. Gallardo, G. Skorobogatov, and C. Barrado, "On-the-Fly Olive Trees Counting Using a UAS and Cloud Services," *Remote Sensing*, vol. 11, no. 3. p. 316, 2019.
- (EP39) A. Safonova, S. Tabik, D. Alcaraz-Segura, A. Rubtsov, Y. Maglinets, and F. Herrera, "Detection of Fir Trees (*Abies sibirica*) Damaged by the Bark Beetle in Unmanned Aerial Vehicle Images with Deep Learning," *Remote Sensing*, vol. 11, no. 6. p. 643, 2019.
- (EP40) Lin, Q., Huang, H., Wang, J., Huang, K., & Liu, Y. (2019). "Detection of Pine Shoot Beetle (PSB) Stress on Pine Forests at Individual Tree Level using UAV-Based Hyperspectral Imagery and Lidar". *Remote Sensing*, 11(21), 2540.
- (EP41) Zhou, J., Tian, Y., Yuan, C., Yin, K., Yang, G., & Wen, M. (2019). "Improved UAV Opium Poppy Detection Using an Updated YOLOv3 Model". *Sensors*, 19(22), 4851.

- (EP42) Salamí, E., Gallardo, A., Skorobogatov, G., & Barrado, C. (2019). "On-the-fly olive tree counting using a UAS and cloud services". *Remote Sensing*, 11(3), 316.
- (EP43) C. Hung, M. Bryson, and S. Sukkarieh, "Multi-class predictive template for tree crown detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 68. pp. 170–183, 2012.
- (EP44) R. Calderón, J. A. Navas-Cortés, C. Lucena, and P. J. Zarco-Tejada, "High-resolution airborne hyperspectral and thermal imagery for early detection of Verticillium wilt of olive using fluorescence, temperature and narrow-band spectral indices," *Remote Sensing of Environment*, vol. 139. pp. 231–245, 2013.
- (EP45) R. A. Diaz-Varela, P. J. Zarco-Tejada, V. Angileri, and P. Loudjani, "Automatic identification of agricultural terraces through object-oriented analysis of very high resolution DSMs and multispectral imagery obtained from an unmanned aerial vehicle," *J. Environ. Manage.*, vol. 134, pp. 117–126, Feb. 2014.
- (EP46) P. J. Zarco-Tejada, R. Diaz-Varela, V. Angileri, and P. Loudjani, "Tree height quantification using very high resolution imagery acquired from an unmanned aerial vehicle (UAV) and automatic 3D photo-reconstruction methods," *European Journal of Agronomy*, vol. 55. pp. 89–99, 2014.
- (EP47) Y. Lin, M. Jiang, Y. Yao, L. Zhang, and J. Lin, "Use of UAV oblique imaging for the detection of individual trees in residential environments," *Urban Forestry & Urban Greening*, vol. 14, no. 2. pp. 404–412, 2015.
- (EP48) M. Pérez-Ortiz, J. M. Peña, P. A. Gutiérrez, J. Torres-Sánchez, C. Hervás-Martínez, and F. López-Granados, "A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method," *Applied Soft Computing*, vol. 37. pp. 533–544, 2015.
- (EP49) J. Torres-Sánchez, F. López-Granados, and J. M. Peña, "An automatic object-based method for optimal thresholding in UAV images: Application for vegetation detection in herbaceous crops," *Computers and Electronics in Agriculture*, vol. 114. pp. 43–52, 2015.
- (EP50) L. Comba, P. Gay, J. Primicerio, and D. R. Aimonino, "Vineyard detection from unmanned aerial systems images," *Computers and Electronics in Agriculture*, vol. 114. pp. 78–87, 2015.
- (EP51) O. Hassaan, A. K. Nasir, H. Roth, and M. Fakhir Khan, "Precision Forestry: Trees Counting in Urban Areas Using Visible Imagery based on an Unmanned Aerial Vehicle," *IFAC-PapersOnLine*, vol. 49, no. 16. pp. 16–21, 2016.
- (EP52) A. dos S. Ferreira, A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, and M. T. Folhes, "Weed detection in soybean crops using ConvNets," *Computers and Electronics in Agriculture*, vol. 143. pp. 314–324, 2017.
- (EP53) S. Puliti, L. T. Ene, T. Gobakken, and E. Næsset, "Use of partial-coverage UAV data in sampling for large scale forest inventories," *Remote Sensing of Environment*, vol. 194. pp. 115–126, 2017.

- (EP54) J. Gao et al., “Fusion of pixel and object-based features for weed mapping using unmanned aerial vehicle imagery,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 67. pp. 43–53, 2018.
- (EP55) V. Otero et al., “Managing mangrove forests from the sky: Forest inventory using field data and Unmanned Aerial Vehicle (UAV) imagery in the Matang Mangrove Forest Reserve, peninsular Malaysia,” *Forest Ecology and Management*, vol. 411. pp. 35–45, 2018.
- (EP56) R. Näsi et al., “Remote sensing of bark beetle damage in urban forests at individual tree level using a novel hyperspectral camera from UAV and aircraft,” *Urban Forestry & Urban Greening*, vol. 30. pp. 72–83, 2018.
- (EP57) L. Comba, A. Biglia, D. R. Aimonino, and P. Gay, “Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture,” *Computers and Electronics in Agriculture*, vol. 155. pp. 84–95, 2018.
- (EP58) D. Koc-San, S. Selim, N. Aslan, and B. T. San, “Automatic citrus tree extraction from UAV images and digital surface models using circular Hough transform,” *Computers and Electronics in Agriculture*, vol. 150. pp. 289–301, 2018.
- (EP59) D. Yin and L. Wang, “Individual mangrove tree measurement using UAV-based LiDAR data: Possibilities and challenges,” *Remote Sensing of Environment*, vol. 223. pp. 34–49, 2019.
- (EP60) K. S. Ochoa and Z. Guo, “A framework for the management of agricultural resources with automated aerial imagery detection,” *Computers and Electronics in Agriculture*, vol. 162. pp. 53–69, 2019.
- (EP61) Ochoa, K. S., & Guo, Z. (2019). “A framework for the management of agricultural resources with automated aerial imagery detection”. *Computers and Electronics in Agriculture*, 162, 53-69.
- (EP62) Andújar, D., Moreno, H., Bengochea-Guevara, J. M., de Castro, A., & Ribeiro, A. (2019). “Aerial imagery or on-ground detection? An economic analysis for vineyard crops”. *Computers and electronics in agriculture*, 157, 351-358.
- (EP63) Hadas, E., Jozkow, G., Walicka, A., & Borkowski, A. (2019). “Apple orchard inventory with a LiDAR equipped unmanned aerial system”. *International Journal of Applied Earth Observation and Geoinformation*, 82, 101911.
- (EP64) Ampatzidis, Y., Partel, V., Meyering, B., & Albrecht, U. (2019). “Citrus rootstock evaluation utilizing UAV-based remote sensing and artificial intelligence”. *Computers and Electronics in Agriculture*, 164, 104900.
- (EP65) Madec, S., Jin, X., Lu, H., De Solan, B., Liu, S., Duyme, F., & Baret, F. (2019). “Ear density estimation from high resolution RGB imagery using deep learning technique”. *Agricultural and forest meteorology*, 264, 225-234.
- (EP66) Torres, I., Sánchez, M. T., Cho, B. K., Garrido-Varo, A., & Pérez-Marín, D. (2019). “Setting up a methodology to distinguish between green oranges and leaves using hyperspectral imaging”. *Computers and Electronics in Agriculture*, 167, 105070.

- (EP67) Stefas, N., Bayram, H., & Isler, V. (2019). "Vision-based monitoring of orchards with UAVs". *Computers and Electronics in Agriculture*, 163, 104814.
- (EP68) M. Pérez-Ortiz, P. A. Gutiérrez, J. M. Peña, J. Torres-Sánchez, C. Hervás-Martínez, and F. López-Granados, "An Experimental Comparison for the Identification of Weeds in Sunflower Crops via Unmanned Aerial Vehicles and Object-Based Analysis," *Advances in Computational Intelligence*. pp. 252–262, 2015.
- (EP69) I. H. Yano, J. R. Alves, W. E. Santiago, and B. J. T. Mederos, "Identification of weeds in sugarcane fields through images taken by UAV and Random Forest classifier," *IFAC-PapersOnLine*, vol. 49, no. 16. pp. 415–420, 2016.
- (EP70) R. A. Diaz-Varela, P. J. Zarco-Tejada, V. Angileri, and P. Loudjani, "Automatic identification of agricultural terraces through object-oriented analysis of very high resolution DSMs and multispectral imagery obtained from an unmanned aerial vehicle," *J. Environ. Manage.*, vol. 134, pp. 117–126, Feb. 2014.
- (EP71) M. K. Hentz, A. P. D. Corte, S. P. Netto, M. P. Strager, and E. R. Schoeninger, "Treedetection: automatic tree detection using UAV-based data," *Floresta*, vol. 48, no. 3. p. 393, 2018.
- (EP72) M. Bah, A. Hafiane, and R. Canals, "Deep Learning with Unsupervised Data Labeling for Weed Detection in Line Crops in UAV Images," *Remote Sensing*, vol. 10, no. 11. p. 1690, 2018.
- (EP73) Y. Wang, X. Zhu, and B. Wu, "Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier," *International Journal of Remote Sensing*. pp. 1–15, 2018.
- (EP74) R. Kestur, A. Angural, B. Bashir, S. N. Omkar, G. Anand, and M. B. Meenavathi, "Tree Crown Detection, Delineation and Counting in UAV Remote Sensed Images: A Neural Network Based Spectral–Spatial Method," *Journal of the Indian Society of Remote Sensing*, vol. 46, no. 6. pp. 991–1004, 2018.
- (EP75) K. Kuželka and P. Surový, "Automatic detection and quantification of wild game crop damage using an unmanned aerial vehicle (UAV) equipped with an optical sensor payload: a case study in wheat," *European Journal of Remote Sensing*, vol. 51, no. 1. pp. 241–250, 2018.
- (EP76) N. Demir, "Using UAVs for detection of trees from digital surface models," *Journal of Forestry Research*, vol. 29, no. 3. pp. 813–821, 2018.
- (EP77) J. A. Thomasson et al., "Disease detection and mitigation in a cotton crop with UAV remote sensing," *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping III*. 2018.
- (EP78) M. Basso and E. P. de Freitas, "A UAV Guidance System Using Crop Row Detection and Line Follower Algorithms," *Journal of Intelligent & Robotic Systems*. 2019.
- (EP79) S. Selim, N. K. Sonmez, M. Coslu, and I. Onur, "Semi-automatic Tree Detection from Images of Unmanned Aerial Vehicle Using Object-Based Image Analysis Method," *Journal of the Indian Society of Remote Sensing*, vol. 47, no. 2. pp. 193–200, 2019.

- (EP80) Hassanein, M., Khedr, M., & El-Sheimy, N. (2019). "Crop row detection procedure using low-cost uav imagery system". *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
- (EP81) Neupane, B., Horanont, T., & Hung, N. D. (2019). "Deep learning based banana plant detection and counting using high-resolution red-green-blue (RGB) images collected from unmanned aerial vehicle (UAV)". *PloS one*, 14(10).
- (EP82) Dash, J. P., Watt, M. S., Paul, T. S., Morgenroth, J., & Pearse, G. D. (2019). "Early detection of invasive exotic trees using UAV and manned aircraft multispectral and LiDAR Data". *Remote Sensing*, 11(15), 1812.
- (EP83) Chakraborty, K., Saikom, V., Borah, S. B., Kalita, M., Gupta, C., Meitei, L. R., & Raju, P. L. N. (2019). "Forest biometric parameter extraction using unmanned aerial vehicle to aid in forest inventory data collection". *Current Science* (00113891), 117(7).
- (EP84) Harto, A. B., Prastiwi, P. A. D., Ariadji, F. N., Suwardhi, D., Dwivany, F. M., Nuarsa, I. W., & Wikantika, K. (2019). "Identification of Banana Plants from Unmanned Aerial Vehicles (UAV) Photos Using Object Based Image Analysis (OBIA) Method (A Case Study in Sayang Village, Jatinangor District, West Java)". *HAYATI Journal of Biosciences*, 26(1), 7.
- (EP85) Zaforemska, A., Xiao, W., & Gaulton, R. (2019). "Individual tree detection from UAV Lidar data in a mixed species woodland". *ISPRS Geospatial Week 2019*.
- (EP86) Alekseev, A., Danilov, Y., Nikiforov, A., Guzik, M., & Kireev, D. (2019, August). "The use of unmanned aerial vehicle (UAV) for inventory and assessment of the experimental plantation in Lisino training and experimental forest of Saint-Petersburg State Forest Technical University". In *IOP Conference Series: Earth and Environmental Science* (Vol. 316, No. 1, p. 012004). IOP Publishing.
- (EP87) Marques, P., Pádua, L., Adão, T., Hruška, J., Peres, E., Sousa, A., & Sousa, J. J. (2019). "UAV-based automatic detection and monitoring of chestnut trees". *Remote Sensing*, 11(7), 855.
- (EP88) T. Kattenborn, M. Sperlich, K. Bataua, and B. Koch, "Automatic Single Tree Detection in Plantations using UAV-based Photogrammetric Point clouds," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3. pp. 139-144, 2014.
- (EP89) M. Zortea, M. M. G. Macedo, A. B. Mattos, B. C. Ruga, and B. H. Gemignani, "Automatic Citrus Tree Detection from UAV Images based on Convolutional Neural Networks," in *Conference on Graphics, Patterns and Images*, Foz do Iguaçu.
- (EP90) M. B. Cruzan et al., "Small unmanned aerial vehicles (micro-UAVs, drones) in plant ecology," *Appl. Plant Sci.*, vol. 4, no. 9, Sep. 2016.

Uso del asistente virtual Alexa como herramienta de interacción para el monitoreo de clima en hogares inteligentes por medio de Raspberry Pi y DarkSky API

Adrián Calvopiña¹, Freddy Tapia¹, Luis Tello-Oquendo²

ascalvopina@espe.edu.ec, fmtapia@espe.edu.ec, luis.tello@unach.edu.ec

¹ Computer Sciences Department, Universidad de las Fuerzas Armadas ESPE, Quito, Ecuador

² College of Engineering, Universidad Nacional de Chimborazo, Riobamba, Ecuador

DOI: 10.17013/risti.36.102–115

Resumen: En los últimos años, el desarrollo de la domótica ha aumentado gracias al uso de las capacidades de la Internet de las Cosas (IoT, por sus siglas en inglés) que proveen tecnologías emergentes como Raspberry Pi. Esto permite automatizar tareas en casa o monitorear parámetros específicos. Este artículo propone una mejora en la interacción entre el sistema domótico inteligente y el usuario final utilizando un asistente de voz inteligente en lugar de aplicaciones móviles o una plataforma web local. Además, se exploran otras fuentes de información para pronósticos meteorológicos, como las interfaces de programación de aplicaciones (API) que proporcionan este servicio.

Palabras-clave: Raspberry Pi; Asistente de voz inteligente; Clima; Alexa.

Using the Alexa virtual assistant as an interaction tool for climate monitoring in smart homes by Raspberry Pi and DarkSky API

Abstract: During the last few years, the development of home automation has increased thanks to the use of Internet of Things (IoT) capabilities provided by emerging technologies such as Raspberry Pi. This allows us to automate tasks at home or monitor specific parameters. This article proposes an improvement in the interaction between the smart home automation system and the end-user using an intelligent voice assistant instead of mobile applications or a local web platform. Furthermore, other sources of information for weather forecasts are explored, such as application programming interfaces (APIs) that provide this service.

Keywords: Raspberry Pi; Intelligent Voice Assistant; Weather; Alexa.

1. Introducción

La Internet de las cosas (IoT, por sus siglas en inglés) es un paradigma que se está volviendo cada vez más importante en nuestras actividades diarias, pero que aún no

ha sido explotado plenamente en los hogares. Es un concepto que escuchamos todo el tiempo. La idea que IoT intenta representar está bastante bien ilustrada por su nombre: cosas cotidianas que se conectan a Internet para realizar varias tareas; sin embargo, se trata de mucho más que eso. Para comprender de qué se trata IoT, también debemos entender que sus fundamentos no son nada nuevos.

Desde hace 30 años, se han realizado varias investigaciones con el objetivo de hacer que todos los objetos cotidianos sean un poco más interactivos. Conceptos como el hogar inteligente (Ricquebourg, et al., 2006) (Bugeja, Jacobsson, & Davidsson, 2018) también conocido como el hogar del mañana, han evolucionado antes de que nos demos cuenta de un hogar conectado para entrar en la era de IoT. Si se definiera IoT, sería mejor decir que es una red que interconecta objetos físicos usando Internet (Ma & Liu, 2017). Los objetos mencionados anteriormente utilizan sistemas embebidos o, en otras palabras, hardware especializado que no solo permite la conectividad a Internet, sino que también permite establecer eventos específicos de acuerdo con las tareas que se le dictan de forma remota (González García, Meana Llorián, Pelayo García-Bustelo, & Cueva Lovelle, 2017).

Las cosas se pueden clasificar como objetos que funcionan como sensores y objetos que realizan acciones activas. Cada uno de los objetos conectados a Internet tiene una dirección específica de protocolo de Internet (IP), y a través de esa dirección IP se puede acceder para recibir instrucciones. También puede contactar a un servidor externo y enviar los datos que recopila (Hipertextual, 2018). En un hogar inteligente, aparatos y dispositivos como luces, cerraduras, alarmas, refrigeradores, aspiradoras, televisores e incluso lavadoras se pueden manejar desde un smartphone, tablet o por voz y ser configurados para funcionar de forma automatizada.

En este estudio se propone una solución de hogar inteligente en la que la voz realiza la interacción entre el sistema y el usuario. Usamos la voz ya que es un medio de comunicación más natural y económico; específicamente, se propone el uso de Amazon Alexa (Hayes, 2017) como asistente inteligente. Siempre que se trata el tema de domótica o hogares inteligentes, existe la idea de que la interfaz final debe o puede ser una aplicación para un teléfono inteligente o una página web a la que se accede e interactúa con el sistema. En este estudio, proponemos cambiar este paradigma mediante el uso de Alexa Voice Service (AVS) (Whatis, 2018) y Raspberry Pi (Upton & Halfacree, 2012) utilizando comandos de voz para la interacción hombre-máquina.

El resto del documento está estructurado de la siguiente manera: La Sección II presenta el trabajo relacionado y analiza la evolución del uso de Amazon Alexa y Raspberry Pi. La Sección III presenta las áreas de conocimiento que abordamos y utilizamos en esta investigación. La Sección IV presenta la arquitectura propuesta, detalla sus componentes y cómo interactúan dentro de la arquitectura. La Sección V detalla la implementación de la solución y la configuración de los componentes. Finalmente, la Sección VI muestra las conclusiones y presenta el trabajo futuro planificado.

2. Trabajos relacionados

En (Jindarat & Wuttidittachotti, 2015), los autores proponen una “granja inteligente” con el uso de IoT implementado con Raspberry Pi, Arduino y varios sensores, para monitorear entornos, automatización de dispositivos electrónicos y recopilación de

datos. Por el contrario, el sistema que se propone en esta investigación elimina el soporte de Arduino para que Raspberry Pi por sí mismo realice la recopilación y el análisis de datos. Además, nuestra propuesta simplifica el sistema al no usar una base de datos; esto se debe a que no habrá informes o análisis posteriores con la información recopilada.

En otros estudios, el monitoreo inteligente de entornos se desarrolla mediante el uso de varios sensores, para vigilar la calidad del aire, la temperatura, la humedad, la intensidad de la luz y los movimientos sísmicos (Ibrahim, Elgamri, Babiker, & Mohamed, 2015). Aunque más datos aumentan el alcance del sistema, no se considera necesario centrarse tanto en los datos debido a que se presentan al usuario final de este sistema. En (Ibrahim, Elgamri, Babiker, & Mohamed, 2015), los autores utilizan el protocolo SSH para conectarse remotamente, lo que resulta en una interfaz para interactuar con el sistema bastante versátil pero rústico y primitivo. En (Rao & Uma, 2015) se plantean entornos más tradicionales, monitoreando el consumo de energía y la temperatura con el uso de IoT y el control de luces y puertas a través de domótica. Esto se realiza a través de una aplicación móvil desarrollada para Android, que accede al sistema a través de Internet a todos los servicios montados en una Raspberry Pi que sirve como unidad central de procesamiento. En el caso de un entorno inteligente, siempre es preferible una interacción más natural y fácil de, por lo tanto, en este estudio se considera el uso de Alexa como la interfaz final del sistema y se enfoca en la temperatura y el clima como campo de monitoreo.

En otros trabajos, diferentes asistentes de voz como Siri y Bixby también se plantean (Hoy, 2018), pero dentro de todas las opciones Alexa sigue siendo la mejor decisión porque Amazon permite dar más acciones a su lista de habilidades que permiten plantear proyectos interesantes como se menciona en (Kim, Row, & Nam, 2018) y otros más tradicionales como en (Aravindan & James, 2017). En estas investigaciones, Alexa es el software intermedio para el sistema de domótica/IoT; aunque también hay propuestas como en (Putthapipat, Woralert, & Sirinimnuankul, 2018) donde Amazon y Alexa se dejan de lado para comenzar a usar la API de Google para el discurso a texto que será procesado por Raspberry Pi. Google Translate también propone el uso de Wit. AI (Mitrevski, 2018) como la interfaz final del sistema que se conecta con la API de voz de Google para poder comunicarse con el usuario.

3. Marco conceptual

En esta sección se presentan diferentes áreas de conocimiento las cuales han sido utilizadas para el desarrollo de esta propuesta. Primero, asistente de voz inteligente, la cual se presenta como el enlace entre el usuario final y el sistema; principalmente a través del Amazon Voice Service (AVS, por sus siglas en inglés), el cual contribuye significativamente al estudio y tiene rol principal dentro de la arquitectura del sistema. Por otro lado, Raspberry Pi, se presenta como la estructura física que actúa como punto final para servicios AVS y permitir el monitoreo de los servicios.

3.1. Asistentes de voz inteligentes

Cada compañía que construye un asistente aplica métodos y acercamientos al desarrollo lo cual afecta al resultado final. Un asistente puede sintetizar palabras

más cualitativamente, otro puede hacerlo con más precisión y sin explicaciones ni correcciones adicionales, otros pueden realizar un rango más estrecho de tareas, pero con mayor precisión y como el usuario quiera. El conjunto de características que tiene un asistente depende completamente de las áreas a las que un desarrollador ha prestado mayor atención, dado que todos los sistemas se basan en el aprendizaje automatizado (Polyakov, et al., 2018).

Un asistente de voz inteligente se relaciona por medio de software, asociado con el sistema operativo de dispositivos específicos, el cual pueden interactuar con el ser humano a través de comandos de voz. Para esto, los asistentes de voz solo pueden responder preguntas simples, búsquedas de información, así como el pronóstico del tiempo o realizar acciones telefónicas básicas como “llamar a Jordi”.

Es por esto por lo que las versiones móviles de las páginas web comenzaron a priorizar y adaptar sus interfaces a los comandos de voz. Por ejemplo, estos asistentes están disponibles en muchos de los servicios de Google, desde la búsqueda del navegador hasta Google Maps y, por supuesto, en teléfonos Android, donde el usuario tiene solo que indicar lo que necesita saber y el dispositivo responderá de inmediato. Amazon promueve “echo”, una serie de altavoces inteligentes, para tener un asistente digital en casa. Esto es un cambio tan trascendental que significa olvidar la pantalla tradicional y empezar a interactuar de una manera radicalmente diferente por medio de la voz (Oleoshop, 2018).

Estos dispositivos introdujeron una interfaz conversacional dentro del hogar que permite a los usuarios el pedir y guardar información, controlar casas inteligentes y una serie de acciones dentro del internet. Por ejemplo, una persona con escasa movilidad podría controlar las luces dentro de la casa o las cerraduras usando la voz, también una persona con discapacidad visual puede preguntar la hora o información sobre el clima. Varios estudios encontraron que este nuevo paradigma de las interfaces visuales avanzadas (AVI, por sus siglas en inglés) tiene un tremendo potencial para inclusividad y accesibilidad (Pradhan, Mehta, & Findlater, 2018).

3.2. Servicio de voz ALEXA

Alexa Voice Service (AVS) es el conjunto de servicios de Amazon basado en su asistente de inteligencia artificial controlado por voz para el hogar y otros ambientes. AVS y Alexa se presentaron por primera vez con “echo”, el altavoz inteligente permite la interacción por medio de la voz con varios sistemas del medio y en línea. Alexa está disponible para un número creciente de otros dispositivos, incluidos teléfonos inteligentes, tabletas y controles remotos.

Amazon Alexa Voice Service (AAVS) proporciona servicios de control de voz para dispositivos de hardware integrados como Alexa, entre otros, la línea de productos de la familia Echo de Amazon y dispositivos domésticos inteligentes como termostatos y cámaras de seguridad.

En 2017, Amazon anunció un refrigerador LG compatible con Alexa y Volkswagen anunció que incrustaría Alexa directamente en sus sistemas de automóviles. Alexa Voice servicios no se limitan a productos de consumo. En noviembre 2017, Amazon anunció Alexa for Business, que es una extensión de Alexa Voice Services en el calendario

corporativo, reuniones de mercado, llamadas de conferencia y gestión de tareas. Se predice que 60 millones de dispositivos integrados de Alexa serán vendidos al 2020 y, con la integración actual de Alexa con más de 50 productos individuales de la empresa, parece que esta predicción se alinea con la visión estratégica de Amazon de Alexa para estar en todas partes.

Alexa es un programa de software basado en la nube que realiza los deberes de un asistente personal inteligente mientras la Línea de productos Echo de Amazon (Echo Dot, Echo Tap o Echo Show) es el hardware conectado a Internet (altavoz, micrófono y cámara) siempre escuchando la frase de activación del usuario conocida como la palabra de activación, cuando un usuario dice “Alexa, ¿qué hora es?”, la palabra de activación Alexa le indica al Echo que grabe y transmita el siguiente comando de voz a Alexa Voice Services. Usando una interfaz de programa de aplicación (API), AVS luego pasa los comandos de voz a Alexa, que utiliza reconocimiento de voz y del lenguaje natural para el procesamiento y formulación de las respuestas de Alexa.

Alexa, entonces, responde a los comandos del usuario con los altavoces integrados de Echo,” La hora es las 3:15 pm.” y envía una copia del comando de voz y la respuesta de Alexa a la aplicación Alexa ubicada en la nube de Amazon. Las copias de los comandos de audio del usuario y las respuestas de Alexa se conocen como tarjetas de respuesta. Se puede acceder a las tarjetas de respuesta a través de la aplicación Alexa del teléfono inteligente (Ford & Palmer, 2019).

AVS está integrado con el entorno de comercio electrónico de Amazon, lo que significa que hace que las compras sean rápidas y simples. El sistema puede funcionar como un centro de automatización del hogar, lo que permite al usuario poder controlar los sistemas de calefacción e iluminación. Por ejemplo, Alexa también se conecta con servicios de transmisión en línea y admite If This Then That (IFTTT), un servicio en línea que automatizar las tareas basadas en la web para que cuando los eventos ocurridos, las tareas de seguimiento se activan y gestionan.

En el entorno AVS, los servicios se conocen como habilidades. El Centro de Huracanes, por ejemplo, es una habilidad de Alexa que proporciona información constantemente actualizada sobre sistemas de tormentas, basado en datos de agencias gubernamentales. Una habilidad llamada virtual el bibliotecario es un motor de recomendaciones que sugiere libros, basado en nominaciones a premios, listas de bestsellers y reseñas de usuarios. Alexa Skills Kit, un kit de desarrollo de software (SDK, por sus siglas en inglés), está disponible gratuitamente para desarrolladores y las habilidades están disponibles para descarga instantánea desde Amazon.

3.3. Raspberry Pi

Raspberry Pi es una placa base de computadora de bajo costo; es una computadora pequeña, del tamaño de una tarjeta de crédito, desarrollada en el Reino Unido por la Fundación Raspberry Pi en 2011. Fue ideado para estimular la enseñanza de la informática en escuelas, aunque no comenzó su comercialización hasta 2012.

Raspberry Pi es un pequeño, potente, barato, hackeable y computadora de tablero orientado a la educación introducido en 2012. Funciona de la misma manera que una PC estándar, lo que requiere un teclado para entrada de comandos, una unidad de

visualización y una fuente de alimentación. Esta computadora del tamaño de una tarjeta de crédito con muchas actuaciones y asequible por \$25-\$35 es la plataforma perfecta para interfaz de muchos dispositivos. La gran mayoría de los componentes del sistema: su procesamiento central y unidad gráfica, hardware de audio y comunicaciones junto con 256 MB (Modelo A): se construyen chips de memoria de 512 MB (Modelo B) en un solo componente. La placa Raspberry Pi contiene lo esencial (procesador, chip de gráficos, memoria de programa - RAM) y otros dispositivos opcionales (varios interfaces y conectores para periféricos). El procesador de Raspberry Pi es un sistema en un chip de 32 bits y 700 MHz, que es construido en la arquitectura ARM11 y se puede overclockear para más poder. La memoria flash SD sirve como disco duro para Raspberry Pi.

La Raspberry Pi, como cualquier otra computadora, usa un sistema operativo. La opción de Linux llamada Raspbian es excelente compatible con Raspberry Pi porque es gratis y de código abierto, manteniendo el precio de la plataforma bajo y haciéndolo más hackeable. También hay algunas opciones de sistema operativo que no son de Linux disponible. Una de las mejores cosas de Raspberry Pi es que tiene una amplia gama de usos (Jain, Vaibhav, & Goyal, 2014).

Consiste en una placa que soporta varios componentes que necesita una computadora convencional y es capaz de comportarse como tal. En mayo de 2009, la Fundación Raspberry Pi fue fundada en Caldecote, South Cambridgeshire en Inglaterra como asociación caritativa regulada por la Comisión de caridad de Inglaterra y Gales. La idea es conseguir computadoras portátiles baratas que permiten que los niños las usen sin miedo, abriendo sus mentes y ayudándoles en la ética de “abrir y ver cómo funciona”.

El cofundador de la fundación es Eben Upton, un ex empleado de Broadcom, responsable del software y arquitectura de hardware de la Raspberry Pi. Eben Upton, contactó a un grupo de profesores, académicos e informáticos entusiastas de crear una computadora para alentar a los niños a aprender informática como lo hizo el microordenador Acorn BBC en 1981 (Historia de la Informática, 2018).

4. Arquitectura propuesta

A continuación, se presenta el esquema de arquitectura propuesto. La Figura 1 ilustra los elementos de software y hardware utilizados en este estudio. Amazon “echo” (Colon & Greenwald, 2015) es la plataforma física que admite el uso de Alexa y, por ende, AVS y Alexa Skill, los cuales registran las habilidades personalizadas del usuario. Esto permite la interacción con Raspberry Pi; la información de Alexa Skill se transporta de forma directa y segura a través de un ngrok o túnel virtual (VPN, por sus siglas en inglés), tan pronto como la información llega al Raspberry PI.

Ngrok es un software de proxy inverso de túnel multiplataforma que establece túneles seguros desde un punto final público como Internet a un servicio de red que se ejecuta localmente mientras captura todo el tráfico para una inspección y reproducción detalladas como se muestra en la Figura 2.

Antes de usar ngrok, cuando necesitábamos exponer una aplicación localhost a la web (internet), todo lo que estábamos haciendo era implementar la aplicación en un servidor que ejecutaba una DMZ o solíamos reubicar el host en DMZ y configurar NATing en el

formulado por cada usuario. Para esto se utiliza comandos de voz y la respuesta se procesa leyendo en ese momento los datos de los sensores y haciendo un requerimiento a la API (DarkSky), uno de estos puede ser los sensores de clima y temperatura. Con estos datos, una respuesta se procesa y se envía a través de un servidor JSON, el cual interpreta la respuesta con los datos necesarios para ser aceptado por el AVS según su estándar. Esta respuesta viaja de la misma manera que llegó, pasando a través de ngrok a AVS y se muestra a través de Alexa en Amazon Echo.

Vale la pena enfatizar que el uso del lenguaje natural permite interpretar los servicios a realizar y responder las solicitudes que el usuario formula disminuyendo la barrera entre los sistemas domóticos y entornos del usuario.

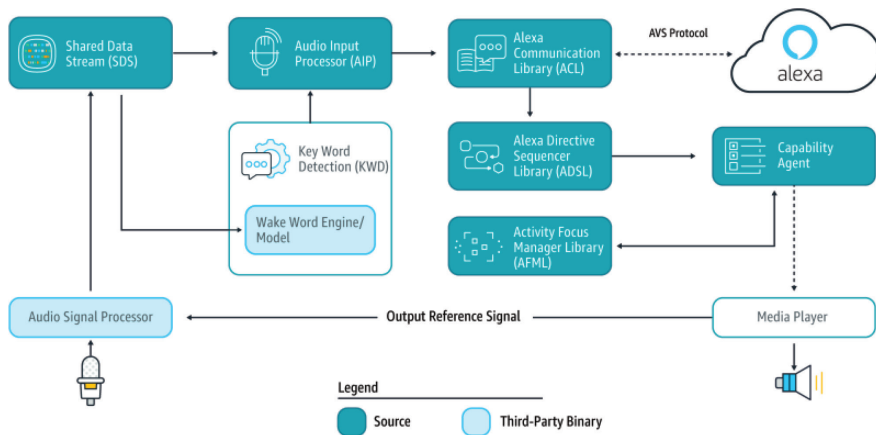


Figura 3 – Arquitetura AVS.

Alexa reside en la nube. El acceso es a través de la API de AVS. Alexa recibe solicitudes como secuencias de voz. Alexa analiza el discurso e identifica la habilidad solicitada. Se crea una representación estructurada de la habilidad y la intención.

Existen dos maneras, si el código de habilidad se implementa en AWS Lambda, ese servicio se activará dentro de AWS. Si se trata de un servicio web al que se accede a través de HTTPS, se realiza la llamada a la API REST (ver Figura 3). En cualquier caso, Alexa recibirá la respuesta como texto. Opcionalmente, la respuesta puede tener imágenes. Alexa convertirá el texto en voz, que se transmite al usuario a través de su dispositivo actual. Las imágenes se mostrarán donde el dispositivo tenga una pantalla.

Una habilidad se identifica mediante un nombre de invocación y se asigna a un servicio de fondo. Una habilidad puede ser diseñada para hacer múltiples cosas, cada una de las cuales se llama una intención. Cada intento puede identificarse mediante una o más palabras o frases, cada una de las cuales se denomina enunciado. Por ejemplo, puede crear una habilidad relacionada con el viaje que tenga tres propósitos: alquilar un automóvil, reservar un vuelo o reservar un hotel (Devopedia, 2018).

5. Desarrollo

El desarrollo de software comienza con la configuración de los componentes dentro de la Raspberry Pi; para esto, el uso de Alexa simplifica el montaje de parte de la infraestructura, por lo que tiene prioridad preparar el punto de llegada en base de habilidades personalizadas que vendrían hacer órdenes que luego serán respondidas por el Raspberry Pi. Para comenzar con el proceso de desarrollo se requiere una cuenta de Amazon para acceder al sitio de desarrolladores de Amazon donde lo primero que debes hacer es crear son las habilidades para Alexa.

5.1. Creando el modelo básico de Alexa Skill

Dentro de la sección de creación de habilidades (skills), se le da un nombre a dicha habilidad, esto debe crearse con una serie de palabras con las que Alexa podría invocar las habilidades creadas. Es importante recordar el nombre de la intención, ya que se utilizará para reconocer las diferentes tareas que Raspberry Pi puede reconocer.

El modelo de interacción es una parte muy importante de nuestra habilidad. Contiene el nombre de invocación de la habilidad, sus intents y expresiones, entre otros aspectos que son cruciales para la interacción con los servicios de habilidades. Los Intents incorporados tienen enunciados incorporados. Es posible extender el número de enunciados existentes agregando nuevos enunciados. Cuando se crea una habilidad, se agregan cuatro intents incorporados al modelo de interacción de forma predeterminada. Estos son: AMAZON.HelpIntent, AMAZON.CancelIntent, AMAZON.FallbackIntent y AMAZON.StopIntent.

El “AMAZON.NoIntent”, “AMAZON.CancelIntent” y “AMAZON.StopIntent” manejarán la solicitud de voz del usuario cuando el usuario decida interactuar con la habilidad, o quiera detener la interacción. El “Amazon.FallbackIntent” se encargará de la solicitud de voz del usuario que no coincide con ninguna de las declaraciones. El “AMAZON.HelpIntent” se encargará de la solicitud de ayuda de voz del usuario. El “AMAZON.NoIntent” no se incluye de manera predeterminada, como los otros intentos integrados, por lo que tendremos que agregarlo (Ralevic, 2018).

```
{
  "name": "raspTemperatureIntent",
  "slots": [],
  "samples": [
    "how hot it is",
    "how cold it is",
    "how is the weather",
    "what is the temperature",
    "look out for the temperature",
    "find me the temperature",
    "get me the temperature",
    "tell me the temperature",
    "give me the temperature"
  ]
}
```

Figura 4 – Intent en format JSON.

La Figura 4 muestra las diferentes formas de invocar las habilidades, esto con el objetivo de que sea más cómodo y flexible trabajar por parte del usuario. Amazon recomienda el uso de frases cortas para esto, porque el usuario debe invocar la habilidad de la manera más natural posible.

Las tragamonedas son un accesorio muy poderoso para construir una habilidad personalizada de Alexa. Por ejemplo, la declaración “how hot it is”. Además, las ranuras múltiples brindan la posibilidad de usar cuadros de diálogo en los que la habilidad Alexa solicita al usuario que complete todos los valores de las ranuras para cumplir con la intención.

5.2. Configuración y mapeo del servidor Ngrok

Ngrok es un túnel VPN que conecta dos puntos, uno de esos actúa como un servidor. Los datos van de punto a punto de forma segura, manteniendo la integridad de los datos a través del sistema propuesto. Esto es crucial porque los estándares y reglas que Amazon aplica para enviar y recibir datos de los AVS. La Figura 4 muestra el servidor ngrok configurado y corriendo; usa el puerto 5000, y debe ser el mismo puerto que el servidor JSON está escuchando.

Puede usar cualquier framework del lado del servidor para trabajar con ngrok. El único requisito aquí es asegurarse de que el servidor local se esté ejecutando correctamente. Para ello se debe colocar “http://6fd1bdc1.ngrok.io” o “https://6fd1bdc1.ngrok.io” (SSL) en la barra de direcciones del navegador y debería mostrar el sitio local. Estas son URL únicas que se crean cada vez que reinicia ngrok, lo que facilita compartirlas para sesiones de prueba únicas en un equipo (SitePoint, 2015).

5.3. Crear el script de Python para JSON

Un servidor que interpreta los datos en formato JSON desde Amazon debería estar configurado. Para este propósito, se ha creado un script en lenguaje Python que escuchó el mismo puerto que se asignó al ngrok para responder a las solicitudes hechas por Alexa. La Figura 5 muestra el código del servidor JSON. Como puede verse, está escuchando el puerto 5000, que es lo mismo que el servidor *ngrok*. Además, podemos resaltar el uso de la biblioteca de Amazon para el desarrollo de este tipo de sistemas.

5.4. Configurando la API de Dark Sky

El código vinculado a la cuenta y la ubicación desde la cual se quiere recopilar los datos deben especificarse para usar la API. La API podría devolver los datos sobre la velocidad del viento y la temperatura, así como los niveles de humedad, pero para el sistema propuesto, solo se usará la temperatura ambiente y un pronóstico por día. La Figura 6 ilustra cómo los pedidos de clima y temperatura podrían ser realizados por la API Dark Sky para ser usado en la respuesta enviada para el usuario.

5.5. Implementación de Infraestructura Física

Hasta ahora se han detallado las configuraciones relacionadas con el software del sistema, pero el hardware específico necesita ser habilitado para poder completar las

solicitudes hechas por el usuario y definidas por las habilidades personalizadas. Para esto se necesitan sensores de temperatura y humedad DHT22, que fueron adquiridos y probados con modelos de la misma serie para asegurar su correcto funcionamiento. Los sensores de humedad y temperatura deben estar conectados para leer los datos y enviarlos al AVS.

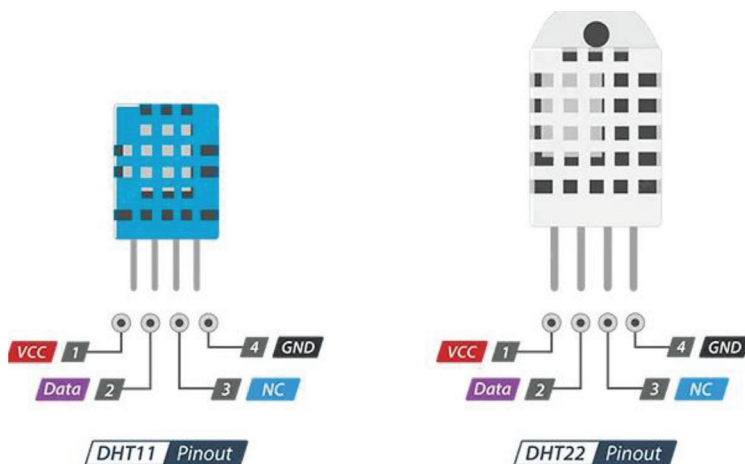


Figura 5 – Sensores DHT11 y DHT22.

En el mercado hay varios sensores de este tipo como se observa en la Figura 5. El sensor DHT11 trabaja con un rango de medición de temperatura de 0 a 50 °C con precisión de ± 2.0 °C y un rango de humedad de 20% a 90% RH con precisión de 4% RH. Los ciclos de lectura deben ser como mínimo 1 o 2 segundos. Por otro lado, en el sensor DHT22 el rango de medición de temperatura es de -40°C a 80 °C con precisión de ± 0.5 °C y rango de humedad de 0 a 100% RH con precisión de 2% RH, el tiempo entre lecturas debe ser de 2 segundos. Ambas unidades usan un sensor de humedad capacitivo y un termistor para medir el aire circundante, y emiten una señal digital en el pin de datos (no se necesitan pines de entrada analógicos), para que un microcontrolador pueda leerlos fácilmente. Son básicos y lentos, pero excelentes para proyectos a pequeña escala que requieren un registro de datos básico (Wia Community, 2018).

Para garantizar la precisión de los datos recibidos, se probaron los sensores DHT11 y DHT22 por cinco días comparando la lectura de datos de cada sensor. Teniendo en cuenta los resultados obtenidos, el sensor DHT22 fue elegido para realizar el trabajo ya que ofrece un 99% de precisión. El sensor está conectado a la Raspberry Pi a través de sus pines de entrada y salida (GPIO), más específicamente a los pines 22 y 2 también especificados en el código que se observa en la Figura 6.

```

logging.getLogger("flask_ask").setLevel(logging.DEBUG)
@ask.intent(' raspTemperatureIntent')
def gpio_control():

    try:
        humidity, temperatureSensor = Adafruit_DHT.read(22, 2)
        print(temperatureSensor)
        location = forecast(key, "house,units="si")
        temperatureOutside = location.temperature
        forecaste = location['currently']['summary']
    except Exception as e:
        return statement('something went wrong with temperature')
    return statement('the temperature of the room is {:.2f} degrees, outside is {} degrees and {}'.format(temperatureSensor, temperatureOutside, forecaste))

@ask.intent(' forecastIntent', mapping={'day': 'day'})
def gpio_control(day):
    print(day)
    days = ["monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday"]
    format_date='%Y-%m-%d'
    try:
        location = forecast(key, "house,units="si")
        datetime_obj = datetime.strptime(day, format_date)
        dayWeek=datetime_obj.weekday()
        dayName = days[dayWeek]
        sum = location.daily[dayWeek].summary
        tempMin = location.daily[dayWeek].temperatureMin
        tempMax = location.daily[dayWeek].temperatureMax
    except Exception as e:
        return statement('something went wrong with forecast')
    return statement('{}: {} Temperature range: {} - {}'.format(dayName, sum, tempMin, tempMax))

app.run(debug=False)

```

Figura 6 – Lectura e interpretación del código del servidor.

6. Conclusiones

El presente trabajo de investigación pone de manifiesto las bondades de las nuevas tecnologías aplicadas a temas de interés colectivos, tal es el caso del medio ambiente. Para esto se hace necesario la orquestación de varios servicios a través de una placa base como Raspberry Pi con otros elementos externos como DarkSky API y las habilidades (skills) que Alexa incorpora, siendo estos últimos los que mayor problema presentan ya que dependen del versionamiento, código fuente, entre otras características que impiden o permiten su integración con otro hardware.

La configuración de eventos (intentos, peticiones), son los que permiten una fácil interacción hombre – máquina, para esto se hace necesario combinar ciertas instrucciones (Python y DarkSkyAPI), he aquí donde se llegaron a evidenciar ciertas necesidades que garanticen la confiabilidad del prototipo, una de estas es la implementación de túneles (Ngrok).

El uso de un servicio de intérprete de lenguaje natural para realizar y responder a las solicitudes del usuario naturaliza y disminuye la barrera entre el sistema de domótica y el entorno del usuario. Como trabajo futuro, se plantea reemplazar el AVS por una herramienta más flexible que admita una variedad de idiomas.

Referencias

- Aravindan, V., & James, D. (2017). Smart Homes using Internet of Things. *International Research Journal of Engineering and Technology*, (pp. 1725-1729).
- Bugeja, J., Jacobsson, A., & Davidsson, P. (2018). Smart Connected Homes. In J. Bugeja, A. Jacobsson, & P. Davidsson, *Internet of Things A to Z: Technologies and Applications* (pp. 359-384). Wiley.
- Colon, A., & Greenwald, M. (2015). Amazon echo. *PC Magazine*.
- Devopedia. (2018). *Amazon Alexa*. Retrieved from <https://devopedia.org/amazon-alexa>
- Ford, M., & Palmer, W. (2019). Alexa, are you listening to me? An analysis of Alexa voice service network traffic. *Personal and Ubiquitous Computing*, 23(1), 67-79.
- González García, C., Meana Llorián, D., Pelayo García-Bustelo, B., & Cueva Lovelle, J. (2017). A review about smart objects, sensors, and actuators. *International Journal of Interactive Multimedia and Artificial Intelligence*.
- González García, C., Meana Llorián, D., Pelayo García-Bustelo, B., & Cueva Lovelle, J. (2017). A review about smart objects, sensors, and actuators. *International Journal of Interactive Multimedia and Artificial Intelligence*.
- Hayes, A. (2017). *Amazon Alexa: A Quick-Start Beginner's Guide*. Andy Hayes.
- Hipertextual. (2018). *Qué es y cómo funciona el Internet de las cosas?* Retrieved from <https://hipertextual.com/archivo/2014/10/internet-cosas>
- Historia de la Informática. (2018). *RASPBERRY PI*. Retrieved from <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi>
- Hoy, M. (2018). Alexa, siri, cortana, and more: An introduction to voice assistants. *Medical reference services quarterly*, 37(1), 81-88.
- Ibrahim, M., Elgamri, A., Babiker, S., & Mohamed, A. (2015). Internet of things based smart environmental monitoring using the Raspberry-Pi computer. *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, (pp. 159-164).
- Jain, S., Vaibhav, A., & Goyal, L. (2014). Raspberry Pi based interactive home automation system through E-mail. *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, (pp. 277-280).
- Jindarat, S., & Wuttidittachotti, P. (2015). Smart farm monitoring using Raspberry Pi and Arduino. *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, (pp. 284-288).
- Kim, S., Row, Y.-k., & Nam, T.-J. (2018). Thermal Interaction with a Voice-based Intelligent Agent. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, (p. LBW631).
- Ma, H., & Liu, W. (2017). A progressive search paradigm for the internet of things. *IEEE MultiMedia*, 25(1), 76-86.

- Mitrevski, M. (2018). Getting started with wit. ai. In M. Mitrevski, *Developing Conversational Interfaces for iOS* (pp. 143-164). Springer.
- Oleoshop. (2018). *Principales asistentes de voz y en qué se diferencian*. Retrieved from <https://www.oleoshop.com/blog/asistentes-de-voz>
- Polyakov, E., Mazhanov, M., Rolich, A., Voskov, L., Kachalova, M., & Polyakov, S. (2018). Investigation and development of the intelligent voice assistant for the Internet of Things using machine learning. *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, (pp. 1-5).
- Pradhan, A., Mehta, K., & Findlater, L. (2018). “Accessibility Came by Accident” Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, (pp. 1-13).
- Putthapipat, P., Woralert, C., & Sirinimnuankul, P. (2018). Speech recognition gateway for home automation on open platform. *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, (pp. 1-4).
- Ralevic, U. (2018). *How To Build A Custom Amazon Alexa Skill, Step-By-Step: My Favorite Chess Player*. Retrieved from <https://medium.com/crowdbotics/how-to-build-a-custom-amazon-alexa-skill-step-by-step-my-favorite-chess-player-dccoedae53fb>
- Rao, P., & Uma, S. (2015). Raspberry Pi home automation with wireless sensors using smart phone. *International Journal of Computer Science and Mobile Computing*, 4(5), 797-803.
- Ricquebourg, V., Menga, D., Durand, D., Marhic, B., Delahoche, L., & Loge, C. (2006). The smart home concept: our immediate future. *2006 1st IEEE international conference on e-learning in industrial electronics*, (pp. 23-28).
- SitePoint. (2015). *How to Use ngrok to Test a Local Site*. Retrieved from <https://www.sitepoint.com/use-ngrok-test-local-site/>
- Upton, E., & Halfacree, G. (2012). *Meet the Raspberry Pi*. John Wiley & Sons.
- Vmoksha. (2020). *Open source technology - What is Ngrok?* Retrieved from <https://vmokshagroup.com/blog/expose-your-localhost-to-web-in-50-seconds-using-ngrok>
- Whatis. (2018). *Alexa Voice Services (AVS)*. Retrieved from <https://whatis.techtarget.com/definition/Alexa-Voice-Services-AVS>
- Wia Community. (2018). *DHT11 vs DHT22: What is the difference?* Retrieved from <https://community.wia.io/d/54-dht11-vs-dht22-what-s-the-difference>

Ameyali: Evaluación de la usabilidad de la plataforma Integral para el control y monitoreo del servicio de agua potable en comunidades rurales bajo la escala de Sus

A. Montiel-De Jesús¹, C. Morales-Constantino¹, S. Ixmattlahua-Díaz¹,
N. Hernández-Chaparro¹, H. Marín-Vega¹

146w0385@zongolica.tecnm.mx, 146w0386@zongolica.tecnm.mx,
sergio.ixmatlahua.pd169@zongolica.tecnm.mx, norma.hernandez.pd183@zongolica.tecnm.mx,
humberto_marin_pd177@zongolica.tecnm.mx.

¹ Tecnológico Nacional de México campus Zongolica, km 4 carretera a la compañía S/N, 95005, Zongolica, México.

DOI: 10.17013/risti.36.116–131

Resumen: El internet de las cosas (IoT), es una de las tendencias tecnológicas que marcan un parteaguas en la forma en que controlamos y monitoreamos las actividades de nuestra vida diaria, desde el trabajo hasta el hogar. En ese sentido, en este trabajo, se presentan los resultados obtenidos de la implementación de tecnologías de IoT aplicadas a la mejora de la operación de un sistema de bombeo hidráulico en una comunidad rural. La metodología utilizada es la orientada a Prototipos para obtener un análisis de requerimientos, el diseño de la aplicación mediante diagramas UML y posteriormente el desarrollo de la plataforma integral que incluye una aplicación Web, una aplicación móvil y un módulo electrónico. Los resultados obtenidos muestran que la plataforma desarrollada así como los sensores utilizados son factibles y viables para controlar la bomba del suministro de agua de la comunidad de Huixtitla, Veracruz, ya que favorece la reducción del costo de la automatización del sistema, para el caso de los usuarios facilita el control y monitoreo de manera remota gracias a la aplicación móvil y a nivel nacional es un producto original en su categoría y aplicabilidad.

Palabras-clave: Internet de las Cosas; automatización; suministro de agua; ingeniería de software; domótica.

Ameyali: Evaluation of the usability of the Integral platform for the control and monitoring of the drinking water service in rural communities under the scale of Sus

Abstract: The internet of things IoT, is one of the technological trends that mark a watershed in the way we control and monitor the activities of our daily lives, from work to home. In this sense, this paper presents the results obtained from the implementation of IoT technologies applied to the improvement of the operation of a hydraulic pumping system in a rural community. The methodology

used is Prototype oriented to obtain an analysis of requirements, the design of the application through UML diagrams and later the development of the integral platform that includes a Web application, a mobile application and an electronic module. The results obtained show that the platform developed as well as the sensors used are feasible and viable to control the pump of the water supply of the community of Huixtitla, Veracruz, since it favors the reduction of the cost of the automation of the system, in the case of Users facilitate remote control and monitoring thanks to the mobile application and at the national level it is an original product in its category and applicability.

Keywords: Internet of Things; automation; water supply; software engineering; domotics.

1. Introducción

El agua es vital para diversas actividades humanas, en México, el 77% del agua se utiliza en la agricultura; el 14% en el abastecimiento público, el 5% es empleada en las termoeléctricas y un 4% en la industria por lo que su disponibilidad es un derecho humano y se debe garantizar para toda la población (CONAGUA, 2010).

En el estado de Veracruz, sólo el 44.37% de las zonas rurales cuentan con servicio de abastecimiento de agua potable (CONAGUA, 2010), una de ellas es el municipio de Soledad Atzompa donde se ubica la comunidad de Huixtitla y a una hora de camino se encuentra el sistema de bombeo que abastece la red de agua a través de tomas en sus domicilios.

Actualmente, la operación del sistema de bombeo del agua en la comunidad de Huixtitla está a cargo de una persona, cuya responsabilidad es encender y apagar diariamente el sistema, además de monitorear su funcionamiento y realizar el mantenimiento. No obstante, la distancia y la orografía del camino son altamente accidentados, poniendo en riesgo la integridad física de la persona a cargo del sistema de bombeo, además de no contar con una administración adecuada de los mantenimientos y servicios que se deben hacer al sistema.

De acuerdo con lo anterior, es conveniente hacer uso de tecnología IoT para mejorar el control y monitoreo del sistema de bombeo de servicio de agua en tiempo real, y de esta manera, hacer prácticas y seguras las tareas de mantenimiento. En ese sentido, el objetivo de este trabajo es presentar los resultados del desarrollo de la plataforma integral Ameyali, la cual se conforma de tres módulos principales para el control y monitoreo del sistema de abastecimiento de agua: a) una aplicación móvil para control a distancia del sistema de bombeo mediante conexión vía Bluetooth o SMS. b) un dispositivo electrónico ubicado en la instalación de la bomba, la cual se conecta mediante GPRS o Bluetooth a la aplicación móvil y los sensores para el monitoreo de temperatura, c) una aplicación Web para visualizar los datos de temperatura obtenidos a través de los sensores.

2. Antecedentes

Existen trabajos realizados mediante tecnologías e implementación de arquitecturas en distintas partes del continente para combatir problemáticas similares.

En (Quezada, Bautista, & Flores, 2014) implementaron un sistema de control y monitoreo de descarga de agua en un pozo de agua potable. En la etapa de diseño crearon las interfaces gráficas de usuario (Graphical User Interface, GUI) para interactuar con el operador y la interfaz hombre-máquina (Human Machine Interface, HMI); ésta última se implementó en software propietario contemplando reglas para el control y monitoreo de las condiciones del sistema. Programaron la HMI interconectado con un controlador lógico programable (Programmable Logic Controller, PLC), obteniendo el control de la maquinaria del pozo.

En (Rodríguez & García, 2014) automatizaron la planta de tratamiento de agua para uso industrial en Colcerámica S.A. planta Girardota, obtuvieron un diagnóstico para establecer las variables críticas: pH, turbidez, flujo y nivel de agua en los vertederos; realizaron los diagramas de Tubería e Instrumentación (P & ID) y el plano 3D de la planta automatizada y formularon los cálculos costo-beneficio generados al implementar la propuesta. Como resultados, la medición exacta de las variables involucradas, registro continuo de las mismas, exceptuando la variable pH; un sistema de dosificación automático para el sulfato de aluminio e hipoclorito de sodio, para que el agua tratada en la planta cumpla con las condiciones requeridas por la compañía.

(Pérez, Koo, García, & Carmona, 2014) proponen un sistema de riego agrícola automatizado, monitoreado y controlado remotamente desde cualquier parte del mundo donde exista servicio de telefonía celular. Buscaron productos de bajo costo para utilizarlos y brindar al usuario un servicio accesible. Las principales herramientas que utilizaron son: Arduino para la automatización del sistema y Android para la interacción con el usuario. Como resultado ofrecen un monitoreo de datos y control del sistema de riego a través de internet, logrando seguridad y flexibilidad al agricultor.

(Robayo, Silva, & Mosquera, 2015) proporcionan un sistema de control computarizado para automatizar la planta “La Esmeralda” que realiza el cargue de agua potable para camiones cisterna. El sistema tiene instrumentos electrónicos que controlan las bombas y válvulas del tanque de almacenamiento, realiza el arreglo necesario para ubicar y montar el sensor ultrasónico para monitorizar la apertura y cierre de las válvulas, el encendido y apagado de las bombas sumergibles y el nivel del tanque en tiempo real. La planta ofrece suministro automático de los volúmenes de agua requeridos por el cliente, todo controlado por medio de una aplicación en LabView que monitorea y visualiza en tiempo real los acontecimientos que genere el proceso de cargue de agua potable a los camiones cisterna.

(Chueca, 2017) reportaron un sistema automatizado para la gestión de una planta de tratamiento de aguas residuales. El control se realizó a través de un PLC y el manejo de la planta automatizada así como la visualización de todas las condiciones que la atañen, se realizó a través de una pantalla HMI conectada al PLC. Los registros de los procesos internos se van creando en un servidor FTP para su análisis posterior. El usuario puede visualizarlos y contrastarlos.

R. López implementó una interfaz SCADA para la visualización en tiempo real del sistema de distribución de agua potable de la ESSAP en la ciudad de Coronel Oviedo. El proyecto implementa sensores de nivel de agua, sensores de presión, servomecanismos para controlar válvulas principales y las diferentes señalizaciones asociadas a un control

central, para monitoreo constante de todo el proceso de manera automática. El lenguaje utilizado fue JAVA y el microcontrolador fue Arduino Uno (López-Benítez, 2017).

F. A. Calle y J. X. Gaibor, propusieron un sistema de riego automático para dispositivo móvil, el usuario envía un mensaje de texto hacia la Shield GPRS/GSM SIM900, el texto es leído por la tarjeta Arduino y acciona los pulsadores para activar la PLC, de esa manera se enciende el riego por goteo en las plantaciones de ASOFRUT ubicada en la ciudad de Ambato (Calle-Zambrano, 2017).

Calderón et al. desarrollaron un prototipo con componentes electrónicos compatibles con el paradigma de IoT (Internet of Things, Internet de las Cosas) para monitorear el nivel de agua y el llenado de contenedores. Se hace la lectura del sensor de nivel por parte del Arduino, cada 5 segundos; se identifica la diferencia clasificándose de nivel alto, normal o bajo, se envía el dato al módulo transmisor y lo dirige hacia la estación central. La tecnología se aplicó en una planta de tratamiento de agua, y mantuvieron en constante monitoreo por 24 horas cada 6 segundos. La aportación principal es el uso de sensores de nivel de agua (Calderón, Junio 2018).

O. J. López aportó un sistema automático para monitorear y accionar la bomba de agua, mediante comunicación inalámbrica implementando una infraestructura de red basada en una tecnología Wifi con un estándar 802.11, para controlar el encendido y apagado del bombeo, y detectar el nivel de agua existente con sensores en una interfaz gráfica desarrollado en LabView, el beneficio es para la comunidad de La Parroquia, Pimocha (López-Correa, 2018).

M. A. Marueta diseñó e implementó el prototipo de un sistema SCADA basado en un microcontrolador para control del tren de filtración (tratamiento primario) de una planta de depuración de aguas grises. la finalidad fue proveer un control flexible del sistema a través de Internet por los volúmenes de agua que se manejan. El sistema alojado en una PC, permite que el microcontrolador ejecute el algoritmo de control del proceso, con las funciones de SCADA (Murueta, 2019).

3. Aplicación de la metodología

Para el desarrollo de la plataforma integral, se implementó el Modelo Orientado a Prototipos representado en la Figura 1.

Las etapas que conforman el Modelo Orientado a Prototipos son: investigación preliminar; especificación de requerimientos y prototipo; diseño técnico; programación y pruebas; operación y mantenimiento. Durante la etapa de investigación preliminar se definió la problemática y se establecieron los aspectos y la factibilidad del desarrollo del producto software.

La etapa de especificación de requerimientos y prototipo se compone de 4 fases: “Análisis”, “Diseño y construcción”, “Evaluación” y “Modificación”. En la primera fase “Análisis” para el análisis de requerimientos se utilizó la plantilla estándar IEEE830 (IEEE, std 830-1998), se generó la carta del proyecto (Project Charter) y un plan de gestión del proyecto basado en el PMBOK Sexta Edición. En la segunda fase de “Diseño y construcción”, con base en los requerimientos funcionales, se realizó el diseño de los

diagramas de casos de uso, secuencia, actividades y componentes. También se diseñó un prototipo de las interfaces gráficas de usuario para visualizar la navegabilidad de la aplicación Web y móvil.

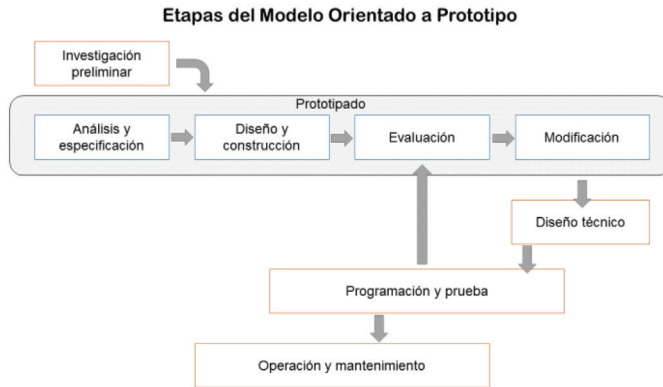


Figura 1 – Las etapas del Modelo Orientado a Prototipos

Durante la fase de “Evaluación” se realizaron reuniones con el cliente para validar el diseño, la funcionalidad y los futuros entregables del proyecto. Después de evaluar el prototipo de interfaces, el cliente realizó observaciones y comentarios como parte de la mejora del prototipo inicial. Con base en los comentarios y observaciones realizadas por el cliente, en la fase de “Modificación” se realizaron cambios en el diseño del prototipo inicial, así como ajustes en los requerimientos funcionales, por lo cual se modificó el contenido de la plantilla IEEE830. En la siguiente etapa denominada programación y pruebas, se construyó cada uno de los módulos que componen las diversas partes de la plataforma integral.

En la etapa de pruebas de funcionalidad, se realizaron pruebas de integración, es decir, durante la codificación, cada componente (electrónico y software) fue probado de manera individual validando los datos de entrada y salida. Por último, la etapa de operación y mantenimiento permitió la implementación de la plataforma integral en un ambiente de producción. Este ambiente de producción se divide en tres partes: el primero, un servidor Web, donde se hospeda la aplicación Web de manera local, para esto, en la oficina de la agencia municipal de la comunidad de Huixtla; el segundo ambiente, es un dispositivo móvil con sistema operativo Android, en este dispositivo se instaló la aplicación móvil para el control remoto de la bomba de agua; por último, el tercer ambiente, es el panel de control eléctrico, ubicado en las instalaciones del sistema de bombeo. Aquí se realizó la adaptación del cableado eléctrico del encendido de la bomba, agregando el módulo de control para operar la bomba de manera remota.

4. Resultados

Los resultados obtenidos en la primera etapa de la metodología fue información preliminar de la instalación del sistema de bombeo: a) el tiempo de traslado existente entre la instalación de la bomba y la comunidad de Huixtitla es de 40 minutos a pie, b) una persona es encargada de encender y apagar la bomba por ello recorre el camino dos veces al día (mañana y tarde), c) el encendido de la bomba usa un circuito de corte, por ello brinda la posibilidad de realizar configuraciones en el control, d) la instalación eléctrica permite la conexión de otras herramientas y aparatos.

La segunda etapa “Especificación de requerimientos y prototipos” se conforma por 4 fases: “Análisis y especificación”, “Diseño y construcción”, “Evaluación” y “Modificación”. En la etapa “Análisis y especificación” se realizó la recopilación de datos para identificar los tipos de usuarios y obtener los requerimientos específicos mediante el estándar IEEE830. A continuación, se mencionan algunos requerimientos funcionales.

- Tener acceso al funcionamiento de la bomba a distancia
- Encender la bomba mediante el uso del teléfono celular y a larga distancia
- Apagar la bomba mediante el uso del teléfono celular y a larga distancia
- Encender la bomba mediante el teléfono celular a una corta distancia
- Apagar la bomba mediante el teléfono celular a una corta distancia
- Encender y apagar el sistema de bombeo de manera manual
- Monitorear el funcionamiento del sistema de bombeo

La etapa “Diseño y construcción” implementa diagramas UML para establecer el diseño y funcionamiento de la arquitectura de la plataforma integral. En la figura 2 se presenta el diagrama de casos de uso de la aplicación móvil. En este caso de uso se definió el rol Operador, el cual representa a la persona que utilizará la aplicación móvil. De la misma manera, se definieron los casos de uso que modelan las principales funciones de la aplicación: encender vía SMS, este caso de uso permite al usuario enviar un SMS desde la App móvil para indicar el encendido de la bomba; apagar vía SMS, mediante este caso de uso, el usuario puede indicar el apagado de la bomba; encender vía Bluetooth, como alternativa a los dos casos anteriores, la aplicación permite encender el sistema de bombeo mediante Bluetooth; y por último, el caso de uso apagar vía Bluetooth, realizando la función contraria al caso anterior vía Bluetooth.

Para representar la modularidad de la plataforma integral, las dependencias entre ellas, la comunicación y el medio de enlace se presenta el diagrama de componentes donde se observan 5 paquetes (ver Figura 3), el paquete “Dispositivo móvil” se comunica con el paquete “Red móvil” con los módulos que controlan el funcionamiento de la bomba mediante señal GPRS, a su vez los comandos AT son recibidos por el módulo SIM900 que transmite la orden de encendido o apagado a la placa ARDUINO para activar los sensores de temperatura.

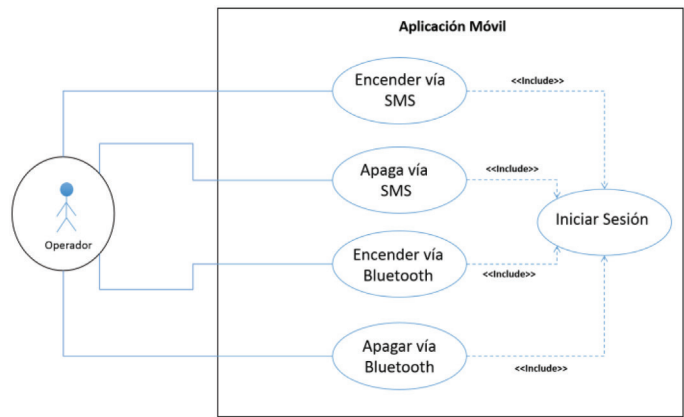


Figura 2 – Diagrama de casos de uso de la aplicación móvil

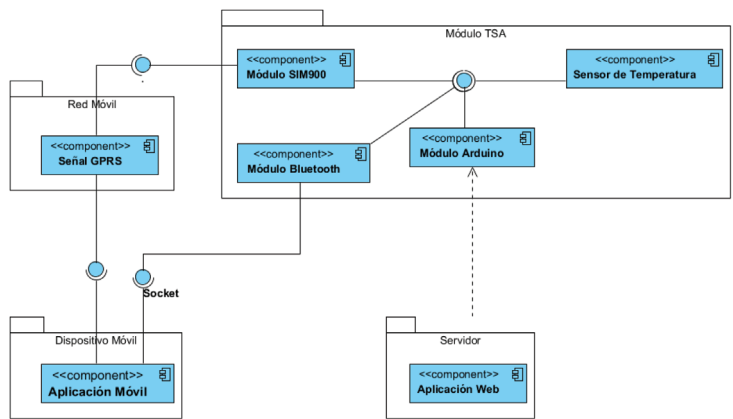


Figura 3 – Diagrama de componentes principales de la plataforma

Con base en el modelo 4 + 1 vistas, propuesto por Philippe Kruchten (Kruchten, 1995), se diseñó la arquitectura de software de la plataforma AMEYALI, en donde se establecen cada uno de los componentes y la comunicación para el funcionamiento de toda la plataforma integral. En la Figura 4, se visualiza cada uno de los elementos que forman parte de la arquitectura, así como la interacción entre ellos.

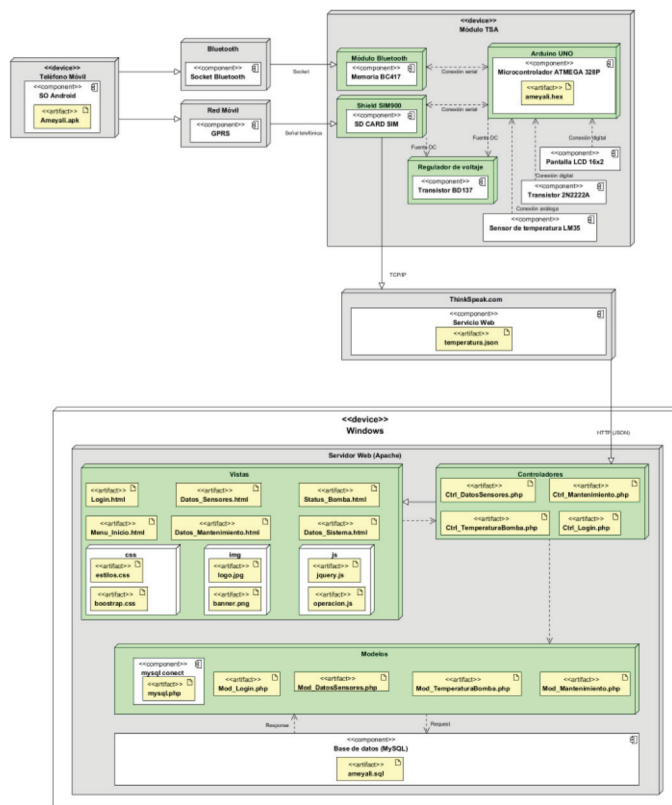


Figura 4 – Arquitectura de la plataforma (vista física)

Cada uno de los nodos tiene una funcionalidad esencial para la comunicación, en la tabla 1, se describen los más importantes.

Nombre	Documentación
<i>Módulo TSA</i>	El módulo TSA es una caja con dimensiones de 21x21x8 cm. elaborado con material de plástico para evitar descargas eléctricas hacia el operador. Dentro de ella contiene dispositivos electrónicos, cables de conexión, módulos de relevadores, módulos de comunicación y una conexión de corriente hacia la bomba, para hacerlo funcionar a distancia y de manera directa. Contiene un sensor de temperatura que va a monitorear la bomba en tiempo real, para enviar los datos hacia la aplicación Web.
<i>Arduino UNO</i>	Este microcontrolador funciona con un voltaje de 5 V a 2 A, por ello es necesario equilibrar la energía que se le conecta. Debe tener cargado el código de programación ameyali.hex, dentro del código ya se especifican las entradas de los demás dispositivos, y el orden en que los hará funcionar. Controla la entrada de datos (Bluetooth y SIM900) y el monitoreo del sensor.

Nombre	Documentación
<i>Módulo Bluetooth</i>	Contiene las configuraciones dentro de la memoria BC417 para ejercer una conexión segura y privada hacia la aplicación móvil, los comandos para su operación son las siguientes: 1. Nombre: Ameyali_CCA_01 2. SSID: o8o8 3. Velocidad en Baudios: 19200
<i>Teléfono Móvil</i>	Cualquier marca de teléfono móvil que cumpla con las siguientes características: • Contar con servicios GPRS • Tener acceso a conectividad vía Bluetooth • Tener una versión de Android 4.0 en adelante • Instalar la aplicación ameyali.apk • Abrir aplicación para comprobar la función que realiza
<i>Shield SIM900</i>	Se debe considerar que este módulo debe contener un Chip de la compañía Telcel, ya que las configuraciones se realizaron para ese fin. El voltaje que necesita es de 5 V a 3 A.
<i>Sensor de temperatura LM35</i>	Se conecta cerca de la bomba de agua, para monitorear constantemente la temperatura y enviar los datos hacia la tarjeta de Arduino.
<i>Vistas</i>	La carpeta vistas contiene los siguientes tipos de archivos los cuales se comunican entre sí para mostrar las visualizaciones al usuario: • Clases, éstas tienen la tarea de darle forma al contenido, para que el usuario pueda realizar las tareas que necesita. • JS, estos archivos realizan acceso a datos y manejan operaciones de consulta hacia el servicio en Thingspeak™. • CSS, brinda diseño a las ventanas, ejemplo de ello son los colores, figuras y formas. • Img, archivos que complementan las ventanas, imágenes, videos y sonidos.
<i>Controladores</i>	La programación de clases se hizo en la arquitectura MVC (Modelo Vista y Controlador). Los controladores tienen la función de administrar las operaciones hacia los modelos, esto mediante la programación Orientada a Objetos.
<i>Modelos</i>	Los modelos son archivos con código funcional que ejecuta operaciones hacia el gestor de base de datos, consultar, modificar, insertar, y en su caso eliminar.

Tabla 1 – Descripción de los componentes en la arquitectura general

Los diagramas que dan soporte a la arquitectura de la plataforma integral se desarrollaron en la etapa de “Diseño técnico”. Los diagramas obtenidos son de clases, esquemáticos (donde se modela el módulo electrónico de comunicación), diagrama de objetos y diagramas PCB. Estos diagramas se diseñaron con las herramientas CASE Visual Paradigm Enterprise v10.0 y Autodesk Eagle v8.2.1.

Para el desarrollo del módulo de control, se diseñó un diagrama esquemático para modelar la comunicación entre los componentes electrónicos. Estos componentes electrónicos son: placa Arduino R3, Shield SIM900, pantalla LCD 2x16, módulo Bluetooth, LEDs, resistencias, transistores, sensor de temperatura LM35, relay. En la Figura 5A se visualiza el diagrama esquemático del módulo de control, en la figura 5B se visualiza el módulo TSA que se desarrolló siguiendo el diagrama esquemático.

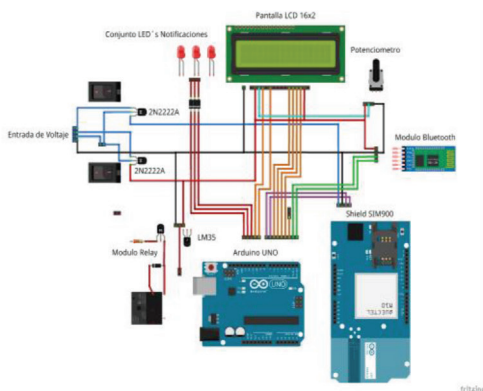


Figura 5A – Diagrama esquemático del módulo de control

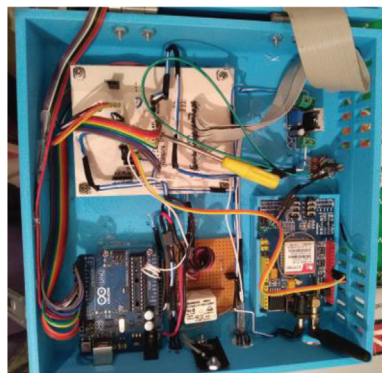


Figura 5B – Módulo TSA

5. Evaluación de usabilidad de la plataforma integral

Kitchenham (Kitchenham, 1997) propuso una metodología para la evaluación de métodos y características cuantitativas y cualitativas en ingeniería del software. Kitchenham afirma que una evaluación cualitativa es ampliamente recomendada para una evaluación de usabilidad. Por su parte, Brooke y Sauro proponen el System Usability Scale que es una herramienta rápida y confiable para medir la usabilidad (J., 1996) (Jeff, 2011).

En la evaluación de la usabilidad de plataforma integral para el control y monitoreo del sistema de abastecimiento de agua en la comunidad de Huixtitla se implementó el System Usability Scale que es una herramienta metodológica muy similar a la Escala de Likert y que se usa para medir la usabilidad de un objeto, dispositivo, aplicación o plataforma; es una manera rápida y confiable de medir la usabilidad.

Diseño de estudio

Para la evaluación cualitativa de usabilidad de la plataforma integral para el control y monitoreo del sistema de abastecimiento de agua en la comunidad de Huixtitla se implementó un cuestionario con escala System Usability Scale (SUS) para evaluar aspectos como la facilidad de uso, experiencia de seguridad y experiencia de uso; para ello se elaboraron las siguientes preguntas, en las cuales el usuario tuvo que responder si estaba fuertemente en desacuerdo, fuertemente de acuerdo o alguna respuesta intermedia.

1. ¿Me gustó usar los componentes de la plataforma integral?
2. ¿Me sentí muy seguro usando la plataforma?
3. ¿Considero que la plataforma es fácil de usar?

4. ¿Considero que no necesito apoyo de una persona técnica para usar los componentes de la plataforma?
5. ¿No necesito aprender muchas cosas antes de comenzar a utilizar la plataforma?
6. ¿No necesito configurar muchas cosas antes de utilizar la plataforma?
7. ¿Considero que una persona aprendería a usar la plataforma muy rápidamente?
8. ¿Considero que es sencillo utilizar la aplicación móvil para control a distancia del sistema de bombeo mediante conexión vía Bluetooth o SMS?
9. ¿Considero que es fácil de utilizar el dispositivo electrónico ubicado en la instalación de la bomba?
10. ¿Considero que es sencillo de utilizar la aplicación Web para visualizar los datos de temperatura obtenidos a través de los sensores?

Participantes

La evaluación de usabilidad está orientada a la persona responsable del sistema de bombeo del agua en la comunidad de Huixtitla cuya responsabilidad es encender y apagar diariamente el sistema, además de monitorear su funcionamiento y realizar el mantenimiento; así como a personas que conocen o realizaron tareas relacionadas con el sistema de bombeo del agua de la comunidad. El objetivo de la evaluación de usabilidad es comprobar que cualquier persona sin habilidades de programación y sin conocimientos de aplicaciones Web, aplicaciones móviles o Internet de las cosas implementen la plataforma integral para el control y monitoreo del sistema de abastecimiento de agua de manera fácil, segura y rápida. Cabe mencionar los participantes no tenían conocimiento del System Usability Scale por lo que se les dio una introducción del SUS y del instrumento empleado para la evaluación.

Procedimiento de recopilación de datos

Para la evaluación de usabilidad se les pidió a los participantes que usarán todos los componentes de la plataforma integral de la manera que lo harían en un día normal de trabajo; posteriormente a su uso se aplicó a los participantes un cuestionario basado en la escala de SUS para evaluar la usabilidad de la herramienta, el cuestionario evaluó aspectos como facilidad de uso, entre otros aspectos con una escala de respuesta del 1 al 5 que equivale del “Totalmente en desacuerdo” al “Totalmente de acuerdo”, los participantes tuvieron un lapso de 5 minutos para contestar el cuestionario.

Resultados de la evaluación de usabilidad

En esta sección se presentan los resultados de la evaluación de usabilidad de la plataforma integral para el control y monitoreo del sistema de abastecimiento de agua en la comunidad de Huixtitla utilizando la escala de SUS. Los resultados obtenidos del cuestionario de evaluación de usabilidad que se representan en la tabla 2 corresponden a la media aritmética de las respuestas de los participantes, una representación de los resultados se presenta en la Figura 6.

Pregunta	Media
¿Me gustó usar los componentes de la plataforma integral?	4.8
¿Me sentí muy seguro usando la plataforma?	4.1
¿Considero que la plataforma es fácil de usar?	4.4
¿Considero que no necesito apoyo de una persona técnica para usar los componentes de la plataforma?	3.6
¿No necesito aprender muchas cosas antes de comenzar a utilizar la plataforma?	3.6
¿No necesito configurar muchas cosas antes de utilizar la plataforma?	3.4
¿Considero que una persona aprendería a usar la plataforma muy rápidamente?	4.5
¿Considero que es sencillo utilizar la aplicación móvil para control a distancia del sistema de bombeo mediante conexión vía Bluetooth o SMS?	4.7
¿Considero que es fácil de utilizar el dispositivo electrónico ubicado en la instalación de la bomba?	4.8
¿Considero que es sencillo de utilizar la aplicación Web para visualizar los datos de temperatura obtenidos a través de los sensores?	4.3

Tabla 2 – Resultados de la evaluación de usabilidad

Los resultados obtenidos en la evaluación de usabilidad fueron favorables como se puede observar en la tabla 2, la media aritmética en la mayoría de las preguntas mayor o igual a 4 de 5 que representa “Totalmente de acuerdo” con un 70% de las preguntas y un 30% de las preguntas tendientes a 3.6 que representa “Normal”; lo que representa una evaluación de usabilidad satisfactoria en la cual los participantes estuvieron de acuerdo en todas las preguntas. Los resultados del SUS en el uso de la herramienta, la media de las puntuaciones arroja un resultado favorable “de acuerdo” (4.22) con una pequeña desviación estándar (0.1878) lo que implica que los participantes tuvieron una experiencia de usabilidad favorable con respecto a su interacción con la plataforma integral para el control y monitoreo del sistema de abastecimiento de agua en la comunidad de Huixtla.

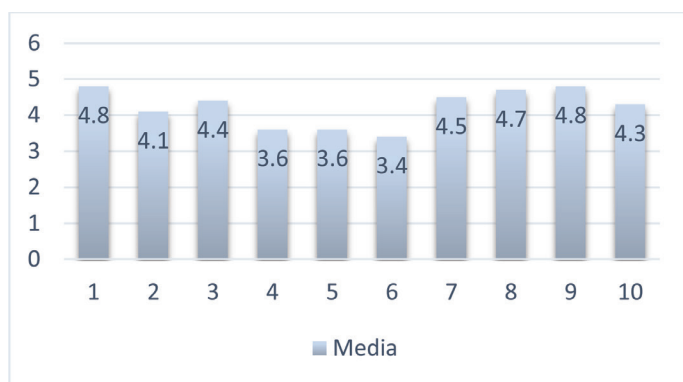


Figura 6 – Representación gráfica de la evaluación de usabilidad

Las preguntas que obtuvieron un mejor puntaje fueron: ¿me gustó usar los componentes de la plataforma integral? Y ¿considero que es fácil utilizar el dispositivo electrónico ubicado en la instalación de la bomba? La razón es porque estos elementos contienen una interfaz intuitiva para el usuario, las medidas cómodas para instalar y operar.

La evaluación realizada por los habitantes arroja que la plataforma integral es fácil de usar, no se necesita tener un conocimiento amplio de las tecnologías, cualquier persona que sepa leer y escribir puede aprender a usar los componentes, la aplicación móvil y Web son sencillos de operar, los colores y espacios dentro de la plataforma les gustó a los usuarios. Estas características fueron aceptadas y aprobadas gracias a la etapa de evaluación y modificación en la que el cliente dio precisiones acerca de las interfaces gráficas.

Las preguntas con menor puntaje fueron: ¿no necesito configurar muchas cosas antes de utilizar la plataforma? Y ¿no necesito aprender muchas cosas antes de comenzar a utilizar la plataforma? Con valoración de “Normal”, esto de acuerdo a los comentarios que dieron los usuarios, ya que recientemente están trabajando con una computadora y con un sistema así, necesitaron de una capacitación inicial pero con la práctica diaria mejoraron sus conocimientos.

Dentro de las 10 preguntas, se consideran 2 como las más importantes ya que esto determina las decisiones a futuro de la plataforma, cada uno tuvo diferentes porcentajes en los resultados de la evaluación. La figura 7A representa la facilidad de uso de la aplicación móvil para controlar el sistema de bombeo, el 70% está totalmente de acuerdo ya que contiene interfaz intuitivo y la operatividad es práctica.

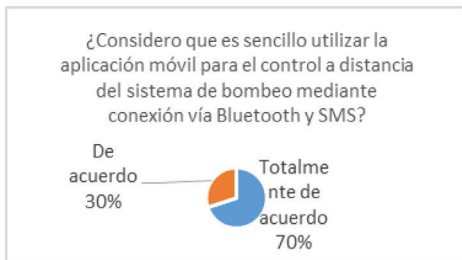


Figura 7A – Gráfica representativa de la facilidad de uso de la aplicación móvil para controlar el sistema de bombeo

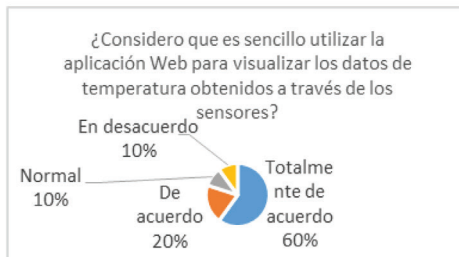


Figura 7B – Gráfica representativa de la facilidad de uso de la aplicación Web para visualizar los datos de la temperatura obtenidos a través de los sensores

La Figura 7B muestra la gráfica acerca del uso de la aplicación Web para visualizar los datos que monitorean los sensores. El 60% está totalmente de acuerdo, pero también se puede notar que hay un 10% en desacuerdo y normal, lo que con llevó analizar la razón, una vez conversando con los usuarios comentaron que no todos saben manejar una

computadora, seleccionar datos en la gráfica para mayor información y que necesitan de mayor familiarización.

Conclusiones

La plataforma integral para el control y monitoreo del sistema de abastecimiento de agua en la comunidad de Huixtitla cumple como una solución tecnológica a cada una de las necesidades especificadas por el cliente, gracias a la implementación de Internet de las cosas.

El trabajo realizado define la manera en que dos tipos de aplicaciones en ambientes Web y móvil interactúan con dispositivos electrónicos como la placa Arduino Uno y sensores de temperatura para controlar el encendido y apagado de una bomba de agua, donde se implementó servicios Web alojados en la plataforma ThingSpeakTM.

Beneficios obtenidos

Durante las pruebas funcionales se obtuvo la conexión estable de los dispositivos, el envío de datos y la visualización de información de manera directa, ahorrando tiempo y trabajo de los usuarios y beneficiando la operatividad del sistema de bombeo.

Los usuarios de las aplicaciones comentaron el gran beneficio que obtuvieron al utilizar la tecnología desarrollada y aplicada en este caso de estudio, ya que realizan el encendido y apagado de forma remota y segura con un solo click sin exponer la integridad de ningún usuario, disponen de un sistema de agua automatizado, práctico e intuitivo que puede ser utilizado en horarios poco adecuados para los operadores del servicio.

Una arquitectura escalable, esta plataforma, además de ayudar a las comunidades rurales, también sirve como un marco de referencia en el desarrollo de aplicaciones basadas en el IoT, ya que la arquitectura diseñada, al ser completamente modular, con un bajo acoplamiento y alta cohesión, permite agregar o quitar funcionalidades para controlar nuevos dispositivos electrónicos y sensores.

Trabajos a futuro

Como trabajo a futuro se realizará la solicitud de registro de patente de un modelo de utilidad ante el Instituto Mexicano de la Propiedad Intelectual.

Este trabajo apoyará a futuros proyectos e investigaciones (como seguridad y análisis de riesgos, predicción de mantenimientos y daños mediante técnicas de IA aplicadas a la domótica, entre otras) referentes a la programación en Arduino y Android, también en la conexión de módulos como lo son la Shield SIM900 y Bluetooth mediante la publicación de los resultados en este artículo.

Por otro lado, se pretende escalar la funcionalidad de plataforma para el control y monitoreo de otros servicios para la comunidad, como el alumbrado público, en ese sentido, se pretende llevar a cabo convenios con Ayuntamientos para la implementación de AMEYALI en sus comunidades.

Agradecimientos

A los habitantes de la comunidad de Huixtla, Veracruz por otorgar las facilidades y accesos a las instalaciones donde se ubica el sistema de abastecimiento de agua y permitir el desarrollo de este proyecto.

Al Instituto Tecnológico Superior de Zongolica por el apoyo y los espacios proporcionados para realizar este trabajo.

Al Dr. Emmanuel de Jesús Ramírez Rivera por los conocimientos aportados para la escritura de este artículo.

Referencias

- Android Developers. (2018). Obtenido de <https://developer.android.com/studio/releases/>
- Calderón, C. A. (2018). Monitoreo y automatización del proceso de bombeo y almacenamiento de agua potable aplicado a una Planta de Tratamiento de Agua. In: *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1-6. Cáceres: IEEE.
- Calle-Zambrano, J. X.-V. (2017). Automatización de un sistema de riego con monitoreo local usando una touch y control remoto inalámbrico vía GSM (Arduino Open Source) para el mejoramiento dentro del campo agrícola en Asofrut de la Ciudad de Ambato. Riobamba: ESPOCH.
- Chueca, P. (2017). Automatización de una planta de tratamientos de aguas residuales. Retrieved from: <http://hdl.handle.net/10251/88981>.
- CONAGUA. (2007). Manual de Agua Potable, Alcantarillado y Saneamiento. Coyoacán, D.F., México.
- Evans, D. (2011). Internet de las cosas. Cómo la próxima evolución de Internet lo cambia todo. San José, CA.: Cisco IBSG.
- Gauchat, J. D. (2012). El gran libro de HTML5, CSS Y Javascript. Barcelona, España: Marcombo.
- Ingeniería de software. (2017). Recuperado el 04 de abril de 2018, de: http://ingenieriadesoftware.mex.tl/63758_AUP.html
- Brooke, J. (1996). SUS: a quick and dirty, usability scale. London: Taylor and Francis.
- Jeff, S. (2011). Measuring usability with the System Usability Scale (SUS). MeasuringU.
- Kitchenham, B. B.-P. (1997). Evaluation and assessment in software engineering. Information & Software Technology. DOI:10.1016/S0950-5849(97)00024-4
- Kruchten, P. B. (1995). The 4+1 View Model of Architecture (Vol. 12). IEEE Software.

- López-Benítez, R. (2017). Diseño de automatización del sistema de distribución de agua potable de la ESSAP en la ciudad de Coronel Oviedo. Coronel Oviedo: Universal Nacional de Caaguazu.
- López-Correa, O. J. (2018). Sistema Automático para el monitoreo de servicios de agua potable de la Parroquia Pimocha del Cantón Babahoyo. Babahoyo: Universidad Técnica de Babahoyo.
- Murueta, M. A. (2019). Automatización de planta de tratamiento de aguas con microcontrolador y servicio Web. Cd México: Ciudad Universitaria.
- Martin Fomler, K. S. (1999). UML gota a gota. México: Addison Wesley Longman.
- Pérez, B., Koo, J. M., García, F., & Carmona, J. (2014). Automatización, monitoreo y control remoto de un sistema de riego agrícola con código abierto. LACCEI Latin American and Caribbean Conference for Engineering and Technology, (20), 1-9.
- Quezada, J. C., Bautista, J., & Flores, E. (2014). Diseño e implementación de un sistema de control y monitoreo basado en HMI-PLC para un pozo de agua potable. Ingeniería Investigación y Tecnología, XV(1), 1-10.
- Robayo, F. I., Silva, D. M., & Mosquera, D. J. (2015). Sistema de control automatizado en planta de cargue de agua potable para camiones cisterna. Revista Ingeniería y Región, I(13), 1-10.
- Rodríguez, Y. A., & García, L. E. (2014). Propuesta de automatización de una planta de tratamiento de agua para uso Industrial. Revista Politécnica, X(18), 1.

Critérios Editoriais

A RISTI (Revista Ibérica de Sistemas e Tecnologias de Informação) é um periódico científico, propriedade da AISTI (Associação Ibérica de Sistemas e Tecnologias de Informação), que foca a investigação e a aplicação prática inovadora no domínio dos sistemas e tecnologias de informação.

O Conselho Editorial da RISTI incentiva potenciais autores a submeterem artigos originais e inovadores para avaliação pelo Conselho Científico.

A submissão de artigos para publicação na RISTI deve realizar-se de acordo com as chamadas de artigos e as instruções e normas disponibilizadas no sítio Web da revista (<http://www.risti.xyz/>).

Todos os artigos submetidos são avaliados por um conjunto de membros do Conselho Científico, não inferior a três elementos.

Em cada número da revista são publicados entre cinco a oito dos melhores artigos submetidos.

Criterios Editoriales

La RISTI (Revista Ibérica de Sistemas y Tecnologías de la Información) es un periódico científico, propiedad de la AISTI (Asociación Ibérica de Sistemas y Tecnologías de la Información), centrado en la investigación y en la aplicación práctica innovadora en el dominio de los sistemas y tecnologías de la información.

El Consejo Editorial de la RISTI incentiva autores potenciales a enviar sus artículos originales e innovadores para evaluación por el Consejo Científico.

El envío de artículos para publicación en la RISTI debe hacerse de conformidad con las llamadas de los artículos y las instrucciones y normas establecidas en el sitio Web de la revista (<http://www.risti.xyz/>).

Todos los trabajos enviados son evaluados por un número de miembros del Consejo Científico de no menos de tres elementos.

En cada número de la revista se publican cinco a ocho de los mejores artículos enviados.

Os asociados da AISTI recebem a RISTI gratuitamente, por correio postal. Torne-se associado da AISTI. Preencha o formulário abaixo e envie-o para o e-mail aistic@gmail.com

Los asociados de la AISTI reciben la RISTI por correo, sin costo alguno. Hazte miembro de la AISTI. Rellena el siguiente formulario y remítelo al e-mail aistic@gmail.com



Formulário de Associado / Formulario de Asociado

Nome/Nombre: _____

Instituição/Institución: _____

Departamento: _____

Morada/Dirección: _____

Código Postal: _____ Localidade/Localidad: _____

País: _____

Telefone/Teléfono: _____

E-mail: _____ Web: _____

Tipo de Associado e valor da anuidade:

☐ Individual - 35€

☐ Instituição de Ensino ou I&D/Institución de Educación o I&D - 250€

☐ Outro (Empresa, etc.) - 500€

NIF/CIF: _____

Data/Fecha: __/__/____ Assinatura/Firma: _____



Revista Ibérica de Sistemas e Tecnologias de Informação
Revista Ibérica de Sistemas y Tecnologías de Información

©AISTI 2020 <http://www.aisti.eu>

