

Aplicación de UML y SCRUM al desarrollo del software sobre control de acceso

Jaime A. Páez^{1,2*}, Jairo A. Cortes^{1,2}, Fredys A. Simanca^{1,2}, y Fabián Blanco^{1,2}

(1) Facultad de Ingeniería, Programa de Ingeniería de Sistemas, Universidad Cooperativa de Colombia, Bogotá – Colombia. (Correo-e: Jaime.paez@campusucc.edu.co, Jairo.cortes@campusucc.edu.co, fredys.simanca@campusucc.edu.co, fabian.blancog@campusucc.edu.co)

(2) Grupo NEOTIC, Bogotá – Colombia.

* Autor a quien debe ser dirigida la correspondencia

Recibido May. 12, 2021; Aceptado Jul. 7, 2021; Versión final Ago. 9, 2021, Publicado Oct. 2021

Resumen

El objetivo del presente estudio es desarrollar una aplicación móvil, que administre y controle la entrada y salida de bicicletas de los estudiantes, docentes y administrativos, de la Universidad Cooperativa de Colombia (UCC) sede Bogotá. Modelos UML (Unified Modeling Language) son aplicados para mejorar la calidad del software mediante el uso adecuado de métricas y requerimientos del software que controla la entrada y salida de bici usuarios por medio de una aplicación móvil construida con tecnología de última generación (Ionic y Angular). Se utiliza el proceso agile Scrum, permitiendo obtener el producto de software para el control de acceso a la universidad, mediante una aplicación web y una app. Se concluye que el desarrollo del software responde a las necesidades de la universidad, mejorando el control sobre el parqueo de bicicletas en sus instalaciones.

Palabras clave: UML; métricas; software; tecnología móvil; Ionic; Angular

Applying UML and SCRUM to develop an access control software

Abstract

The main objective of the present study is to develop a mobile application that manages and controls the entry and exit of bike users (students, faculty, and administrative staff) at the Cooperative University of Colombia (UCC, in Spanish), Bogota campus. Unified modeling language (UML) models are applied to improve software quality by using appropriate metrics and software requirements to control the entry and exit of bike users through a mobile application with up to date technologies (Ionic and Angular). The agile Scrum process is used to obtain the software product to control access to the university through a web application and an app. It is concluded that the software developed here responds to the needs of the university by improving control over on campus bicycle parking.

Keywords: UML; metrics; software; mobile technology; Ionic; Angular

INTRODUCCIÓN

El uso de la bicicleta en Bogotá es cada vez más común según estudio realizado por la Universidad Libre nos habla que en la capital de la republica actualmente se mueven más de 835 mil habitantes en bicicleta convirtiéndola en una de las ciudades con el mayor número de vías para las bicicletas. Respecto a los motivos de los viajes, el 20% de esta población lo hacen para llegar a su lugar de estudio, es allí donde es importante dar una solución tecnológica para el ingreso y control de los bici usuarios de la Universidad Cooperativa de Colombia. Es evidente la necesidad de una herramienta capaz de mejorar dicho proceso en la universidad, ya que se evidencia deficiencia al momento del ingreso a las instalaciones, evidenciando demoras y embotellamientos al momento del ingreso.

Lo anterior genera la necesidad de desarrollar e implementar un software que satisfaga este requerimiento utilizando las últimas tecnologías y buenas prácticas de desarrollo, enfocado en lectura de código de 2D, para optimizar y agilizar el ingreso a las instalaciones de la universidad, dando como valor agregado el mejoramiento de la seguridad en un 90% mediante los diferentes pasos de seguridad que brinda el software (Zalewski et al., 2017) ya sea al momento de ingresar y salir de la universidad, con lo anterior se daría un beneficio a la comunidad universitaria y los administradores de los ingresos de las bicicletas en este caso las personas que prestan la seguridad, brindándoles una herramienta funcional y capaz de solucionar esta problemática evidenciada. La pregunta de investigación que guio el proceso fue ¿Cómo mediante diagramas de UML y Scrum se puede desarrollar de una aplicación móvil que controle los accesos de entradas y salidas de bicicletas?

Como caso de aplicación y objetivo general está en desarrollar una aplicación móvil, que administre y controle la entrada y salida de las bicicletas de los estudiantes, docentes y administrativos, de la UCC sede Bogotá. Se realizó un estudio de campo el día 12 de febrero del 2019 (Amézquita et al., 2016) en el cual se tomaron tiempos, y en un periodo de 2 horas ingresaron 40 bicicletas obteniendo en promedio un ingreso de 00:41,44 segundos por estudiante. Se realiza una solicitud mediante correo electrónico al área de infraestructura, la cual suministra fotos del formato de registro de visitantes y vehículos (Márquez et al, 2011) donde se evidencio los datos que deben registrar los estudiantes al momento de ingresar a la universidad con una bicicleta. En el mismo ejercicio se realizan unas capturas fotográficas sondeo que muestra un cuello de botella en el momento de la llegada, al mismo tiempo varios estudiantes con bicicleta generando una fila para realizar el registro ya antes mencionado.

Realizando una revisión de la información de varias investigaciones relacionadas con Scrum (Santos et al., 2018), con los diagramas de UML con sus métricas y las tecnologías Ionic (Chen et al., 2019; Salleh et al., 2020) y Angular (Santos et al., 2016; Wautelet et al., 2016), estas han permitido que se desarrolle el software de control de acceso con eficiencia y calidad, así como la conformación y desarrollo de la arquitectura de software. Este proyecto está basado en patrones arquitectónicos (Gamma et al., 2003) y bajo la metodología ágil (Hernández et al., 2019) para el desarrollo de software Scrum (Babativa et al., 2016), de la cual se usaron algunas de sus características tales como (Carvalho et al, 2012): Backlog de Producto, Planificación del sprint, Reunión Scrum semanal para dar el informe y avances del proyecto, Burndown Chart.

METODOLOGÍA

Para el proceso de abstracción del mundo real se utilizaron los diagramas del lenguaje de modelado unificado de sus siglas del inglés UML (Wei et al., 2014; OMG, 2017) en particular los casos de uso, se desarrolló el modelo de datos y clases, se utilizaron tecnologías como Angular (Escott, 2019) y Ionic (Tafur et al., 2020), los versionamientos del sistema se controlaron mediante el repositorio de uso general como lo es GitHub. El software desarrollado se realizó en ambientes Web (González et al., 2019) y se consolidó una App específica para mostrar los resultados vía aplicaciones en telefonía móvil, a su vez se utilizó Nodejs (Lee et al., 2019) y JavaScript (Krohn, 2012), para el control de las interfaces. Como estructura metodológica se utilizó Scrum (Vidal et al., 2013) donde se definen los diferentes pasos que se debe seguir como un método ágil de desarrollo y de trabajo en equipo, en este proyecto está definido de acuerdo con el Project Charter (Sáenz, 2018), donde se establecen las metas por desarrollar como criterios de aceptación, el desarrollo de la App móvil, el desarrollo del ambiente web, entregas de documentación, fechas de inicio, fin y todos los hitos de control, y la definición del equipo de trabajo como el scrum master, producto owner y el development team, nombres en ingles tomados de la estructura de Scrum (Elallaoui et al., 2016b; Schwaber et al., 2013). Algunas fases descritas son:

Backlog de producto

En esta fase se realiza un proceso de recolección de información aplicando diversas técnicas como la observación, la entrevista, la revisión documental para establecer los requerimientos funcionales y no funcionales de lo que será la aplicación a desarrollar, al igual que la identificación de riesgos asociados a cada acción o requerimiento funcional, valorándolos con prioridad baja, medio y alto, con planes de mitigación

específicos. En la investigación se desarrollaron cincuenta y nueve requerimientos funcionales, nueve requerimientos no funcionales de la app, tres requerimientos no funcionales del ambiente web y 19 requerimientos de sistema, como muestra de este proceso se puede evidenciar en la tabla 1 sobre los requerimientos.

Tabla 1: Requerimientos.

<i>Código</i>	<i>Nombre</i>	<i>Descripción</i>
RF-APP-001	Registrar Usuario	La aplicación debe permitir que el usuario realice un registro por primera vez a través de su correo personal o institucional
RF-APP-002	Enviar Clave Temporal	La aplicación debe contar con sistema de envío de correos, por medio del cual se entreguen las credenciales de ingreso por primera vez
RF-APP-003	Completar Información de Registro	La aplicación debe detectar cuando es la primera vez que se inicia sesión para solicitar la información personal al usuario.
RF-APP-004	Inicio sesión	La aplicación debe permitir que el usuario inicie una sesión única, brindando la seguridad de la información de los datos del mismo.
RF-WEB-059	Listar Barrios	La aplicación debe permitir listar los barrios
RNF-APP-001	Generar Token Seguridad	La aplicación debe tener la capacidad de crear un Token de seguridad con un tiempo prudente de vencimiento, después de validar si los datos de inicio de sesión son correctos, para asegurarle seguridad al usuario final de la aplicación
RD-WEB-004	FrontEnd Reportes	Ajuste Visual de los Reportes de la web administrativa

Arquitectura

El diseño arquitectónico es de tipo cliente– servidor de tres capas (Alsaadi et al., 2019) donde el contenedor de microservicios mediante Nodejs provee varios servicios como la universidad, los usuarios, la configuración y las bicicletas, relacionados con la base de datos de almacenamiento. El servidor web utiliza un server Gateway para establecer las conexiones con los clientes mediante acceso móvil como remoto. La arquitectura de infraestructura utiliza un servidor VM_Azure con el front-end de la web administrativa, seguridad de MS Azure, relacionado con los servicios de Nodejs y base de datos MySql.

Para el desarrollo del software se realiza el proceso de las historias de usuario y se hace una especificación de los requerimientos, se utilizan formatos estandarizados como el de la IEEE-830 para la documentación tanto funcionales como no funcionales, se realizan los prototipos que permite hacer su proceso de validación, seguidamente se hace un proceso de modelación del mundo real mediante el uso de diagramas de UML (Carvalho et al., 2013), como son los de caso de uso, secuencia, dominio, componentes, clases entre otros. Para el presente artículo se visualizan algunos casos de uso utilizados para el desarrollo del software.

Casos de uso bici usuario

El caso de uso general del bici usuario contempla todas las operaciones básicas que registran la información del usuario como ver el perfil del usuario, editar, cambiar la contraseña, registrarse por primera vez como bici usuario, consultar las bicicletas almacenadas, se podrá modificar la información registrada de las bicicletas y la generación de un código de seguridad como es su QR.

Casos de uso celador

El caso de uso del celador o vigilante tiene asociado varias funciones que realizan para el control de acceso como el registrar la entrada, registrar las salidas, realizar consultas de bicicletas con su información asociada, registrar a los visitantes, con las consultas respectivas, igualmente es importante en este caso de uso registrar apropiadamente la salida de los visitantes que poseen este medio de transporte.

Casos de uso usuario plataforma Web

Para el caso de uso de la plataforma web, se generan las operaciones que pueden realizar sobre los datos registrados como el inicio de sesión, las operaciones básicas sobre las facultades, las puertas de acceso,

registro de información de afiliación a empresas de salud, correo electrónicos validados, actualización de claves, generación de reportes, registro de cargos y departamentos, así como todas las funciones asociadas a los celadores o vigilantes de las diversas sedes.

Casos de uso súper administrador web

El caso de uso del administrador tiene acceso a todas las opciones del software de control como poder hacer recuperaciones de información relacionadas con las contraseñas, bici usuarios, administradores, más la gestión sobre el usuario administrador, esta función solo es administrada por el área de tecnologías de la Universidad, para garantizar mayor control e integridad sobre la información que se administra en el aplicativo.

Casos de uso usuario general de la aplicación móvil

Los casos de uso relacionados con cada uno de los requerimientos funcionales se establecen para poder validar en forma gráfica, cada uno de estos requerimientos principales utilizados son los de usuario relacionados con la aplicación, el bici usuario y el celador como ejemplo se detalla en la figura 1, y en la figura 2 se muestra el inicio de sesión por parte del usuario validando este proceso con el registro previo de la información, creando una sesión y validando mediante token de seguridad los datos almacenados.

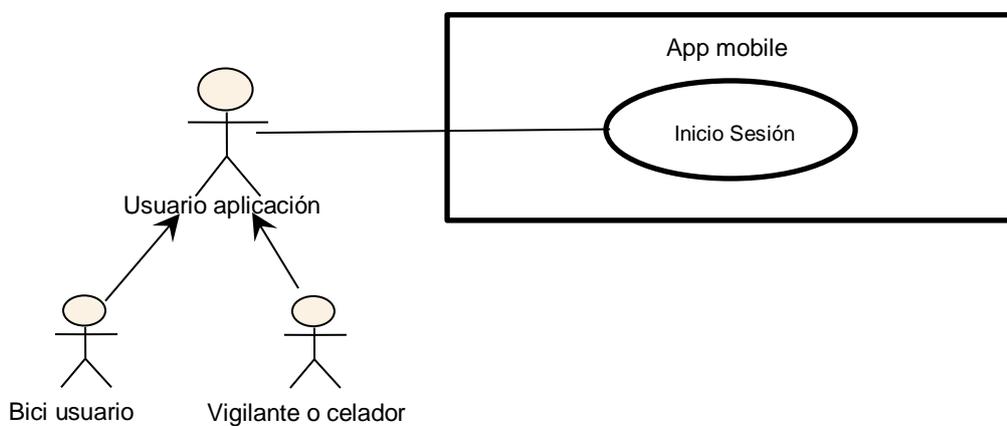


Fig. 1: Funciones generales de usuario

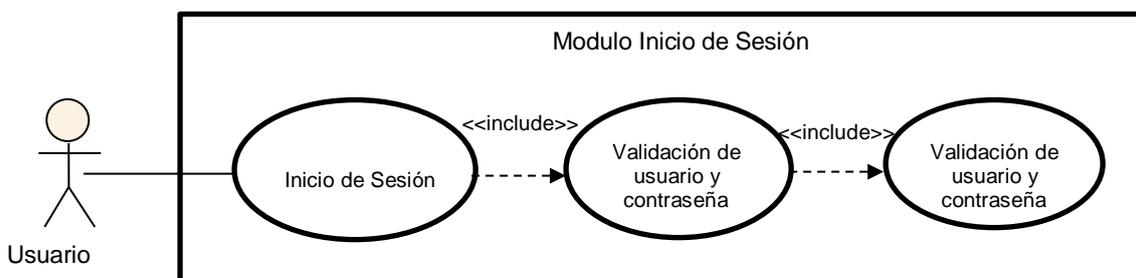


Fig. 2: Inicio de Sesión

Diagrama de clases

El modelo de clases (Elallaoui et al., 2015) diseñado fue diseñado en visual paradigm, generando las clases con sus atributos y métodos necesarios para generar y probar (Elallaoui et al., 2016a) la aplicación que permite controlar los accesos mediante el ingreso a la Universidad utilizando bicicletas. La clase principal denominada usuario mantiene relaciones con la mayoría de clases y acciones del sistema está relacionada con la información de las bicicletas y la oficina que controla el acceso mediante el software a cargo de la empresa que ejerce el control de acceso.

Como descripción del método utilizado anteriormente, se puede describir las fases importantes de Scrum que se encuentran en el Product Backlog que son los requerimientos del sistema ordenados por prioridades, este proceso se descompone de actividades más pequeñas denominadas el Sprint Backlog, estos se desarrollan en periodos de 2 a 4 semanas donde se desarrolla una nueva funcionalidad con control diario denominado Daily Scrum Meeting donde se hacen encuentros de 15 minutos y en donde el scrum master realiza preguntas

Sprint Burndown App

A continuación, se muestran algunos ejemplos como en la figura 3, de lo que se realizó de graficación de los Sprint, en la figura se proyecta el esfuerzo vs los días relacionados con lo planeado vs lo actual, en los que se debe desarrollar cada sprint, esto permite tener un mayor control sobre las actividades que se programan diariamente para cumplir las metas semanales y mensuales.

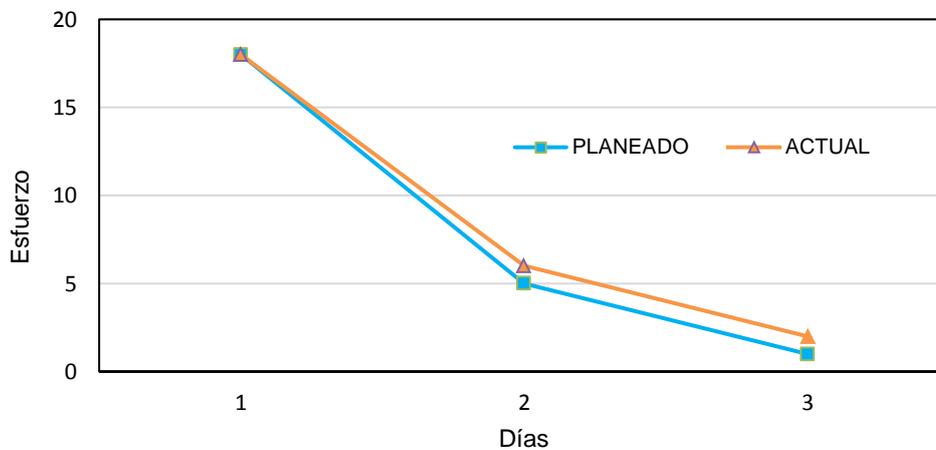


Fig. 3: Sprint 1 App

RESULTADOS Y DISCUSIÓN

En el proceso de recolección de información, la aplicación de la ingeniería de requerimientos, los procesos de observación y análisis de la información permiten proponer una solución software a nivel web y app para el control de acceso de los bici usuarios en la Universidad que tiene múltiples entradas a una sola sede. El desarrollo del software es producto de una investigación con el apoyo de la empresa de vigilancia de la Universidad Cooperativa de Colombia, que permitieron acceder a sus registros sirvieron como insumo para el desarrollo del software. El software para el control de acceso por parte de los bici usuarios deriva tres ambientes uno del usuario, otro del personal de vigilancia y el de administrador, de los cuales se mostraran algunos resultados así:

Ingreso primera vez a BiciparkingApp

Cuando se da el primer acceso en el App, se verá la siguiente pantalla la cual se debe deslizar de derecha a izquierda, la cual conducirá a unos sliders que dan una introducción breve al App, igualmente se solicitará el ingreso del correo electrónico institucional, donde se le enviara un mensaje con la dirección previamente ingresada.

Recepción de correo electrónico

El usuario debe ingresar a su cuenta de correo electrónico previamente ingresada, allí encontrará un correo enviado por la aplicación en el cual se dará la bienvenida a la aplicación y las credenciales temporales para el acceso al aplicativo.

Completar registro y acceso a la información almacenada

Se implementó el sistema para acceder y controlar la información de las bicicletas y lo bici usuarios. Posterior al registro, en el dashboard, tiene varias opciones, como el menú de la aplicación, el nombre de usuario, al presionar en la imagen de huella digital, la aplicación lo dirigirá al código único QR, se puede crear una nueva bicicleta, visualizar las bicicletas creadas, y por último estadísticas de los últimos ingresos y salidas. Otras opciones que se desarrollaron son a continuación así 1) Presione para ver la bicicleta seleccionada, 2) Información de la bicicleta y fotografía de esta 3) Información del propietario e información de este.

Ingreso a BiciparkingApp

Para comenzar a utilizar el App el usuario debe buscar ingresar en las aplicaciones de su celular en el icono de la aplicación de biciparking donde previamente se debe haber instalado en el celular, al hacer clic en el icono se puede acceder a las funciones que se han descrito en los casos de uso validados con los prototipos de interfaces, como se puede ver en la figura 4.



Fig. 4: Ingreso a la aplicación Bici usuarios

El sistema propuesto se desarrolló en Ionic soportado sobre Angular, esto permite heredar todas las características del framework utilizando wrap nativo y gran variedad de plugins para poder usar todos los elementos nativos que requiere el sistema. Los diagramas de UML utilizados en la investigación se les determinaron las métricas para poder verificar el adecuado uso de los requerimientos funcionales, a manera de ejemplo en el caso de los diagramas de clase se utilizaron las métricas del número de atributos, número de asociaciones, número de operaciones, profundidad y acoplamiento, para el caso de los diagramas de casos de uso se utilizaron las métricas asociadas al número de actores, includes y extends entre otros, con el fin de lograr un desarrollo con eficiencia y calidad en la entrega del producto final (Hernández, 2019).

DISCUSIÓN FINAL

El desarrollo del software responde a las necesidades de la organización, mejorando el control sobre el parqueo de bicicletas en sus instalaciones, con tecnologías Ionic y Angular. El modelo del software diseñado permite su replicabilidad con otros indicadores o métricas de UML (Berhouma et al., 2020), no sólo relacionados con la calidad del software sino en otras organizaciones que utilicen servicio de bici parqueo. A pesar de tener una planilla de registro para el ingreso de estudiantes, profesores y visitantes, no es de fácil acceso a más información o documentos procedimentales para ahondar en la investigación, como se argumentó en la definición del problema donde referenciamos las falencias que existe con este proceso. Al diseñar y desarrollar la aplicación móvil y web sobre Ionic y Angular podemos concluir que no es necesario realizar dos desarrollos para cada uno de los sistemas operativos Android y IOS, dando más practicidad al momento de la codificación al integrarse con estas tecnologías híbridas, dándole al final del desarrollo una posibilidad de mejora visual y rendimiento.

Para la implementación de las tecnologías Ionic y Angular facilito la integración con la arquitectura seleccionada generando una eficiencia notable a nivel de infraestructura por el manejo de micro servicios dando como resultado consumos bajos de las cargas al servidor y base de datos. Al implementar estas dos tecnologías dan la facilidad de ejecutar todo el plan de pruebas a medida que el desarrollo va dando un avance, adicional se aplican a los dos sistemas operativos de manera simultánea, permitiendo realizar análisis de las estadísticas sobre el número de usuario de bicicletas que ingresan a la universidad midiendo el nivel de concurrencia de la información que se administra en la plataforma.

El estudio de campo fue fundamental en el proceso, donde se tomaron tiempos en los accesos e ingresos por las diferentes sedes, analizando el tiempo promedio sin la aplicación contrastada con los tiempos medidos utilizando la aplicación, dando como resultado el uso del software como la opción más eficiente y segura tanto para los usuarios como para la universidad. Algunas recomendaciones en la etapa de producción de la aplicación están en la generación de copias de seguridad del almacenamiento de la información en forma diaria, igualmente se recomienda cuando se está trabajando con las tiendas virtuales de Google o Apple contar con las licencias correspondientes. Otras conclusiones y recomendaciones están relacionadas con las empresas de vigilancia donde se les debe proveer de conectividad a internet, así como los dispositivos móviles con los requerimientos mínimos. Los términos y condiciones deben ser aceptados para el manejo de datos personales y no ir en contravía de las normas establecidas por el habeas data.

CONCLUSIONES

De acuerdo al trabajo presentado y a los resultados obtenidos, se pueden plantear las siguientes conclusiones principales: 1) Algunos aspectos importantes de usar Scrum fue el trabajo en equipo, el desarrollo rápido de las funcionalidades del sistema, la integración apropiada de los usuarios como los vigilantes y estudiantes; algunas dificultades radicaron en la disponibilidad de la información y la generación del producto en el área de producción. 2) Las métricas permitieron utilizar adecuadamente los diagramas de UML que para esta investigación se validaron cada uno con los requerimientos propuestos para lograr calidad en el producto final. 3) El método utilizado fue el adecuado por que permitió obtener los resultados esperados en la investigación y satisfacer los requerimientos del usuario. 4) Con el producto desarrollado se logró aplicar Scrum, UML con sus métricas y las tecnologías asociadas como Ionic y Angular logrando un control de acceso en la Universidad.

REFERENCIAS

- Alsaadi, M., Lisitsa, A., Khalaf, M., y Qasaimeh, M. Investigating the Capability of Agile Processes to Support Medical Devices Regulations: The Case of XP, Scrum, and FDD with EU MDR Regulations. https://doi.org/10.1007/978-3-030-26766-7_53. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 11645 LNAI, 581-592 (2019)
- Amézquita, L.L., Durán, D.F., y Fajardo, D.H. Matriz Origen-Destino y Eficiencia en Modos de Transporte Urbano: Un Análisis de la Movilidad de Bogotá. *Semestre Económico*, 19(39), 91-111. <https://doi.org/10.22395/seec.v19n39a4> (2016)
- Babativa, A., Briceño, P., Nieto, A., y Salazar, O. Desarrollo ágil de una aplicación para dispositivos móviles. Caso de estudio: Taxímetro móvil. *Ingeniería*, 21(3), 260–275. <https://doi.org/10.14483/udistrital.jour.reving.2016.3.a01> (2016)
- Berhouma, H. A Generic Model for Software Documentation and its Application in Embedded Systems Developed with Scrum. <https://doi.org/10.1145/3436829.3436858>. ACM International Conference Proceeding Series. 33-36 (2020)
- Carvalho, B., y Mello, C. Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. *Gestão & Produção*, 19(3), 557-573. <https://dx.doi.org/10.1590/S0104-530X2012000300009> (2012)
- Carvalho, S.C., Motta, F.R., Marques, A., y Zanetti, L.P. Comparative research between SCRUM and RUP using real time embedded software development. <https://doi.org/10.1109/itng.2013.112>. Proceedings of the 2013 10th International Conference on Information Technology: New Generations, ITNG 2013 6614397, 734-735 (2013)
- Chen, P.-S., Chen, G.Y.-H., Lien, S.-F., y Huang, W.-T. Using Scrum and unified modelling language to analyze and design an automatic course scheduling system. <https://doi.org/10.1080/02533839.2019.1613930>. Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers, Series A. 42(6), 534-543 (2019)
- Elallaoui, M., Nafil, K., y Touahni, R. Automatic generation of TestNG tests cases from UML sequence diagrams in Scrum process. <https://doi.org/10.1109/CIST.2016.7804972>. Colloquium in Information Science and Technology, CIST ISSN 23271884, 0, 7804972, 65-70 (2016a)
- Elallaoui, M., Nafil, K., y Touahni, R. Automatic generation of UML sequence diagrams from user stories in Scrum process. <https://doi.org/10.1109/SITA.2015.7358415>. 2015 10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015. IEEE 2015, 7358415 (2015)
- Elallaoui, M., Nafil, K., Touahni, R., y Messoussi, R. Automated Model Driven Testing Using AndroMDA and UML2 Testing Profile in Scrum Process. <https://doi.org/10.1016/j.procs.2016.04.119>. Procedia Computer Science. 83, 221-228 (2016b)
- Escott, K., y Noble, J. Design patterns for Angular Hotdraw. <https://doi.org/10.1145/3361149.3361185>. ACM International Conference Proceeding Series, Article number a35, 24th European Conference on Pattern Languages of Programs, EuroPloP 2019; Irsee; Code 155742 (2019)
- Gamma, E., Helm R., Johnson, R., y Vlissides J. *Patrones de Diseño*. 1ª edición, 79-317. Addison Wesley. Madrid, España (2003)
- González, J., Ortiz, J., y Zarate, H. Sistema de Toma de Decisiones y Negociación para Nodos Potenciales de un Sistema de Cómputo Ad Hoc. *Ingeniería y competitividad*, 21 (2), 7481. Epub 21 de septiembre de 2019. <https://doi.org/10.25100/iyc.v21i2i.7481> (2019)
- Hernández, G., Martínez, A., Jiménez, R., y Jiménez, F. Métricas de productividad para equipo de trabajo de desarrollo ágil de software: una revisión sistemática. <https://doi.org/10.22430/22565337.1510>, Tecnológicas. 22(spe), 63-81 (2019)
- Krohn, H. Programación de buscadores en JavaScript para diccionarios digitales. *Cuadernos de Lingüística Hispánica*, (34), 109-130. <https://doi.org/10.19053/0121053X.n34.2019.9410> (2019)
- Lee, S., y Paik, J. An Asynchronous-Driven Node.js Based Intermediary-free Direct Deal Distribution Platform Converged with Cloud Service. <https://doi.org/10.3837/tiis.2019.08.022>, Ksii Transactions on Internet and Information Systems, 13(8), 4212-4226 (2019)

- Márquez, L., Gallo, L., y Chacón, C. Influencia del costo de parqueo en el uso del auto en Bogotá. Ingeniería y Universidad, ISSN: 0123-2126 15 (1), 105-124 (2011)
- OMG, Object Management Group, Unified Modeling Language, <https://www.omg.org/spec/UML/2.5.1/PDF> (2017)
- Sáenz, F., Gutiérrez, F., y Ramos, J. Conformación de equipos ágiles para el desarrollo de software: revisión de literatura. Dimensión Empresarial, 16(2), 39-54. <https://doi.org/10.15665/rde.v16i2.1042> (2018)
- Salleh, S.F., Ujir, H., Sapawi, R., y Hashim, H.F. Accreditation document tracking system using scrum approach. <http://doi.org/10.11591/ijere.v9i1.20418>. International Journal of Evaluation and Research in Education. 9(1), 153-161 (2020)
- Santos, N., Fernandes, J.M., y otros 4 autores. Using scrum together with UML models: A collaborative university-industry R&D software Project. http://doi.org/10.1007/978-3-319-42089-9_34. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2016)
- Santos, N., Pereira, J., y otros 3 autores. Deriving user stories for distributed Scrum teams from iterative refinement of architectural models. <https://doi.org/10.1145/3234152.3234165>. ACM International Conference Proceeding Series. Part F147763, a40 (2018)
- Schwaber, K. y Sutherland, J. La guía de Scrum. Scrumguides. Org , 1 , 21 (2013)
- Tafur, L., Zorrilla, A., y otros 8 autores. Development of MyCheckTime® software for perioperative safety based on Toyota's Lean Methodology. Rev. colomb. anesthesiol. [Internet]. 2020 Mar [cited 2021 Apr 04]; 48(1): 12-19. Available from: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-33472020000100012&lng=en. Epub Feb 28, 2020. <https://doi.org/10.1097/cj9.000000000000148> (2020)
- Vidal, C. L., López, L. P., Rivero, S. E., y Meza, R. O. Extensión del Diagrama de Secuencias UML para el Modelado orientado a Aspectos. Información tecnológica, 24(5), 3-12 (2013)
- Wautelet, Y., Heng, S., y otros 3 autores. Bridging user story sets with the use case model. https://doi.org/10.1007/978-3-319-47717-6_11. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 9975 LNCS, 127-138 (2016)
- Wei, Q., Danwei, G., y otros 4 autores. Research on software development process conjunction of scrum and UML modeling. Proceedings <https://doi.org/10.1109/IMCCC.2014.206>. - 2014 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2014. 6995176, ISSN 00002014, 978-982 (2014)
- Zalewski, J., y Gonzalez, F. Evolution in the Education of Software Engineers: Online Course on Cyberphysical Systems with Remote Access to Robotic Devices. International Journal of Online and Biomedical Engineering, eISSN: 2626-8493, 13(8), 133-146 (2017)

Página en blanco