



Smart Campus

Modelo y resultados

Autores:

Manuel Alejandro Pastrana Pardo

Ana Milena Rojas Calero

Hugo Alberto González López



Smart Campus

Modelo y resultados

ISBN: 978-958-5167-04-9
Primera edición noviembre de 2020

Autores

Manuel Alejandro Pastrana Pardo
Docente Facultad de Ingenierías,
Institución Universitaria Antonio José
Camacho – Grupo de Investigación
GRINTIC

Ana Milena Rojas Calero
Docente Facultad de Ingenierías,
Institución Universitaria Antonio José
Camacho – Grupo de Investigación
GRINTIC

Hugo Alberto Gonzáles López
Rector Institución Universitaria
Antonio José Camacho – Grupo de
Investigación GRINTIC

Colección:
50° aniversario Institución
Universitaria Antonio José Camacho

Gestión editorial
Biblioteca Universitaria Centro
Cultural Jairo Panesso Tascón
Decanato Asociado de
Investigaciones

Corrección de estilo:
Angélica María Grajales R.

Diagramación e impresión:
Ingeniería Gráfica S.A.S.
Institución Universitaria
Antonio José Camacho
Av. 6N No. 28N-102 Tel.: 665 28 28

www.uniajc.edu.co
Cali Valle Colombia

El contenido del presente libro es responsabilidad exclusiva de su(s) autor(es) y en ningún momento representa el pensar de la Institución Universitaria Antonio José Camacho. No está permitida la reproducción total o parcial de este libro por cualquier medio sin el permiso previo y por escrito del editor o de los autores.

Consejo Directivo

Rector

Hugo Alberto Gonzáles López

Vicerrectora Administrativa (invitada)

Mónica Leonor Gómez

Representante del Ministerio de Educación

Raquel Díaz Ortiz

Representante de Ex Rectores

Martha Inés Jaramillo Leiva

Representante de Autoridades Académicas

María Isabel Afanador Rodríguez

Representante Estudiantil

Derly Vanessa Vásquez

Representante Docentes

Gustavo Adolfo Girón

Representante Egresados

Franklin Mantilla

Representante Sector Productivo

Martha Lucía Bernal Marulanda

Secretario General

Luis Guillermo Betancourt

Consejo Académico

Rector

Hugo Alberto González López

Vicerrectora Académica

Zoraida Palacio Martínez

Decana Facultad de Ciencias Empresariales

Francia Elena Amelines

Decano Facultad de Ingenierías

Edwin Jair Núñez

Decano Facultad de Ciencias Sociales y Humanas

Octavio Augusto Calvache Salazar

Decana Facultad de Educación a Distancia y Virtual

María Isabel Afanador Rodríguez

Director Departamento de Ciencias Básicas

Víctor Manuel Uribe

Representante Docentes

Luis Alejandro Vanegas

Representante Estudiantil

Jorge Enrique Domínguez Giraldo

Jefe de Bienestar Universitario

Yolanda Ochoa Grajales

Decano Asociado de Investigaciones

Juan Carlos Cruz Ardila

Director de Proyección Social

James Cuesta Mena

Secretario General

Luis Guillermo Betancourt

Misión

La Institución Universitaria Antonio José Camacho es una entidad de carácter público, comprometida con la formación Integral de excelencia en diferentes niveles y metodologías de la educación superior; contribuyendo de manera significativa al avance de la ciencia, la tecnología, la cultura, a la transformación socioeconómica y al desarrollo de la región y del país.

Visión

La Institución Universitaria Antonio José Camacho es una entidad de carácter público, comprometida con la formación Integral de excelencia en diferentes niveles y metodologías de la educación superior; contribuyendo de manera significativa al avance de la ciencia, la tecnología, la cultura, a la transformación socioeconómica y al desarrollo de la región y del país.

Prólogo

La institución Universitaria Antonio José Camacho (UNIAJC) es una entidad pública adscrita al Municipio de Santiago de Cali que no recibe auxilios por parte del Estado, siendo su principal fuente de ingresos la oferta académica que provee a la comunidad. Cuenta con un total de 4 facultades: Educación a Distancia y Virtual, Ingenierías, Ciencias Empresariales y Ciencias Sociales y Humanas, que generan una oferta académica de 38 programas en los diferentes niveles de formación en modalidad presencial y semipresencial en la ciudad Cali.

Dadas las condiciones limitadas en el presupuesto de la institución se requieren medidas que optimicen los recursos con los que cuenta, aprovechándolos al máximo en los servicios que ofrece. Por lo anterior, en pro de la mejora continua de todos sus servicios hacia la comunidad, en su plan estratégico de desarrollo institucional ha apuntado al incremento de proyectos de investigación aplicada que permitan generar dicha optimización de cara a todos los procesos internos, aplicando los resultados de los diversos temas que se puedan investigar, consultar y debatir en el grupo de investigación GRINTIC en torno a la ingeniería de software, las ciencias de la computación y la electrónica.

Así surge el proyecto Smart Campus, dirigido por la investigadora Ana Milena Rojas Calero, Ingeniera de Sistemas de la Universidad Antonio Nariño, Especialista en Gerencia Informática Organizacional y Magister en Informática y Telecomunicaciones de la Universidad ICESI, docente de carrera con una trayectoria de 24 años en la UNIAJC, quien cuenta con el apoyo de los investigadores: Manuel Alejandro Pastrana Pardo, Ingeniero de Sistemas de la Universidad Santiago de Cali, Especialista en procesos para el desarrollo de software, MSc. en Ingeniería de Software de la Universidad San Buenaventura, profesor asociado de la institución con amplia experiencia en la industria, y Fernando Cifuentes, Ingeniero de Sistemas, egresado de la UNIAJC y quien está vinculado en diversos subproyectos dentro de la iniciativa. Adicionalmente, también participan como co-investigadores el Phd Leandro Flórez Aristizábal, el Mg. Juan Carlos Cruz Ardila, Decano de Investigaciones y el Rector de la UNIAJC, el Mg. Hugo Alberto González López.

Acerca de este libro

Este libro expone los resultados obtenidos durante la ejecución del proyecto de investigación aplicada, denominado Ecosistema Smart Campus, detallando el modelo seguido, desglosándolo en categorías y, por cada una de ellas, identificando qué tipo de problemáticas fueron abordados y cómo se les dio solución, siendo pertinente exponer los casos más relevantes y con mayor impacto dentro de la institución. Esto supone que para la comprensión de su contenido es necesario exponer adicionalmente el modelo seguido para el análisis de los problemas, el diseño de soluciones, el desarrollo, pruebas que garantizan la calidad del resultado, la puesta en marcha y, finalmente, la adopción por parte de la Institución.

Es importante para la comunidad académica e investigadora conocer estos resultados, a fin de poner en contexto todos los avances obtenidos en esta investigación que puedan servir para alimentar otras investigaciones o incluso para mejorar el modelo en sí. Siendo la evolución tecnológica y la mejora continua parte de este proceso constante que busca impactar de manera significativa el campus universitario.

Acerca de los asesores de esta edición

Ana Milena Rojas Calero

Recibió su Máster en Informática y Telecomunicaciones de la Universidad ICESI, Cali, Colombia en 2012. Actualmente trabaja como docente de carrera, adscrita a la Facultad de Ingenierías en la Institución Universitaria Antonio José Camacho. Hasta el 2014 se desempeñó como CIO de la Institución por un período de 15 años. Su línea de investigación es Desarrollo de Software y su principal área de investigación es Redes y Telemática. Su interés investigativo gira entorno del desarrollo de los Smart Campus. Ha publicado tres artículos relacionados con su investigación, cuenta con una amplia experiencia en la implementación y gestión de proyectos en TI. Durante los 23 años de experiencia en la academia ha impartido una amplia gama de cursos. Actualmente es la Main Contact de la Academia Cisco y docente investigador del grupo GRINTIC.

Manuel Alejandro Pastrana Pardo

Recibió su Máster en Ingeniería de Software de la Universidad San Buenaventura, Cali, Colombia, en 2017 y su Especialización en Procesos para desarrollo de software de la misma universidad en 2015. Actualmente trabaja como profesor ocasional tiempo completo adscrito a la Facultad de Ingenierías de la Institución Universitaria Antonio José Camacho. Consultor independiente con una trayectoria de más de 12 años en la industria con empresas de TI y desarrollo de software de talla nacional e internacional. Su interés investigativo está orientado dentro de la línea de ingeniería de software y el desarrollo de Smart campus. Actualmente cursa estudios de postgrado en la Universidad del Cauca.

Hugo Alberto González López

Obtuvo su grado de Magíster en Educación de la Universidad de San Buenaventura sede Cali, en el año 2010; Magíster en Administración de la Universidad ICESI en el año 2014 y Master of Business Administration (MBA) de Tulane University (New Orleans, Estados Unidos) en el año 2015. Está vinculado a la Institución Universitaria Antonio José Camacho desde hace 23 años, tiempo en el cual se ha desempeñado como docente universitario y ha ocupado diferentes cargos académico-administrativos como: Coordinador Académico del Programa Jóvenes en Acción, Director del Centro de Formación Técnico Laboral (CEFTEL), Decano de la Facultad de Ingenierías, Vicerrector Académico y Rector.

Desde su llegada a la rectoría formuló en su plataforma rectoral el proyecto de modernización institucional orientado hacia la adopción de un modelo Smart University, para lo cual lideró la formulación del proyecto de investigación Ecosistema Smart Campus UNIAJC y la formulación del Plan Estratégico de Desarrollo 2020-2030, que adopta este modelo de desarrollo universitario como horizonte institucional.

Contenido

Introducción	15
Escenario de motivación	19
Demoras en los trámites y servicios institucionales.....	22
Obsolescencia en los procesos académico-administrativos y desaprovechamiento del talento humano	24
Repitencia y deserción académica	24
Desaprovechamiento de la infraestructura física en jornadas de baja ocupación	25
Mayor demanda de recursos tecnológicos y conectividad	26
Antecedentes	29
Modelo Propuesto	33
Equipo de trabajo Ecosistema Smart Campus -UNIAJC.....	35
Proceso de desarrollo de software Ecosistema Smart Campus.....	36
Roles dentro del proyecto	36
Análisis y planeación	38
Diseño de software	53
Desarrollo de software y calidad	68
Despliegue	74
Proceso de gestión de proyectos Ecosistema Smart Campus.....	76
Resultados del Proyecto	
Estandarización proceso de desarrollo de software	76
Configuración del ambiente de calidad preventiva	79
Instalación y configuración de Jenkins	135
Proyectos realizados por Smart campus.....	138
FASE I.....	138
FASE II.....	140

Publicaciones en los que participo el equipo de investigadores....	146
Capítulos de libro	147
Eventos científicos en los que participo el equipo de investigadores	149
Registros de software	155
Impacto en la comunidad	157
Referencias	165

Introducción

Las universidades, en cierta medida, son comparables con las ciudades. Esto las hace un espacio ideal para explorar alternativas que permitan mejorar la calidad de vida y la experiencia de la comunidad en un entorno capaz de reinventarse constantemente y aprender de sí mismo en pro de la mejora a todo nivel.

Por tanto, deben tener una administración capaz de hacer uso óptimo de los recursos económicos de los que disponen, aprovechando al máximo las tecnologías disponibles para ello. Controles de calidad sobre el aire, la comida, el manejo de residuos y basuras, el uso efectivo de energía y agua, la medición de gastos y su disminución inteligente, los trámites internos, la distribución de espacios y la movilidad son algunos de los factores claves para mejorar la calidad de los servicios y el entorno en donde la comunidad habita.

La institución Universitaria Antonio José Camacho (UNIAJC) cuenta actualmente con 4 facultades y 44 programas en oferta académica en 2 sedes en la ciudad de Santiago de Cali. Su población está constituida por aproximadamente 8.452 estudiantes por semestre académico, sin contar el personal administrativo, docente y egresados. Esto denota que su población genera un uso constante de los espacios y servicios de la institución, haciendo que la universidad tenga que concebir un proceso de mejora continua a partir de su experiencia y generando transformaciones controladas con el ánimo de avanzar hacia una mejor versión de sí misma.

Para la institución esto no es un panorama desconocido. En el Plan Estratégico de Desarrollo (UNIAJC, 2020), el Concejo Directivo y el Concejo Académico junto con la rectoría y vicerrectoría han consolidado unas directrices que apoyan la resolución de las problemáticas expuestas a continuación:

- Limitada respuesta de la oferta formativa de la institución en relación con las necesidades del entorno.
- Planta docente insuficiente e inadecuada para la proyección estratégica de la institución y la demanda de los programas académicos.
- Bajo aprovechamiento de las relaciones existentes con otros actores con respecto a la formación, investigación y proyección social.
- Baja capacidad de respuesta para el desarrollo de la oferta de educación continua y servicios de extensión.

- Recursos financieros limitados para garantizar la sostenibilidad de la institución.
- Infraestructura y espacios insuficientes e inadecuados para el cumplimiento del propósito misional.
- Subutilización de los sistemas integrados de información que soportan la gestión de procesos y obsolescencia frente a las exigencias de la proyección estratégica.
- Deficiencias en la comunicación con impacto negativo en la eficiencia de procesos académicos y administrativos.

Cada uno de estos tópicos mencionados hacen parte de la realidad institucional que deben ser resueltos de alguna manera para así alcanzar los objetivos estratégicos de la visión y la misión de la UNIAJC.

En las grandes instituciones, una de las mejores formas para dar solución a problemáticas complejas que aquejan a la sociedad ha sido el recurrir, en la comunidad académica, al gremio de la investigación, en específico para generar proyectos de investigación aplicada. Esto puede entenderse como la capacidad de invertir en un mejor futuro basado en las condiciones actuales y buscando las mejores posibilidades a la mano para resolver una necesidad, generando impacto y transformación.

Así, la comunidad investigativa de la UNIAJC, desde el grupo de investigación GRINTIC, genera una propuesta acorde a la iniciativa, liderada por la investigadora Mg. Ana Milena Rojas Calero, quien propone el proyecto Ecosistema Smart Campus en el año 2017 con la colaboración de investigadores: Mg. Hugo Alberto González, rector de la institución; Phd (C) Leandro Flórez Aristizábal; Mg. Juan Carlos Cruz, decano de investigaciones de la UNIAJC y MSc Manuel Alejandro Pastrana.

Un Smart Campus puede ser concebido, según Francisco Maciá Pérez en su libro *Smart University*, como un modelo que mejora la calidad de vida mediante el uso eficiente de todos los recursos disponibles para interconectar servicios y actores en la comunidad. Por tanto, este modelo propende por la creación de proyectos más pequeños aplicando el concepto de *divide y vencerás*, de manera que la institución pueda obtener victorias tempranas sobre las problemáticas mencionadas.

Aquí el potencial de los investigadores es demostrado a través de los resultados expuestos en este libro, generando a la fecha dos fases de este proyecto que ha permitido obtener 8 publicaciones en revistas indexadas, 4 registros de software, 30 tesis dirigidas de las cuales 10 fueron exaltadas como meritorias. Adicionalmente, se ha presentado 3 artículos de divulgación y se ha participado en 11 eventos nacionales e internacionales.



Escenario de motivación

La UNIAJC es una entidad pública del Municipio de Santiago de Cali, que no recibe recursos de financiación de parte de ningún ente territorial. Por lo anterior, requiere de un modelo de sostenibilidad óptimo, alimentado principalmente por sus servicios educativos presenciales y virtuales. Alineado a esto, su visión genera un compromiso social al orientar los mismos a los grupos poblacionales más vulnerables y con mayores limitaciones socioeconómicas, gracias una oferta académica de bajo costo.

Para lograr el anhelado modelo sostenible es necesario un eficiente proceso económico-administrativo de la institución, centrado en la disminución del impacto financiero de los costos fijos, siendo el principal su infraestructura física. Por lo que las estrategias planteadas deben estar alineadas con maximizar el uso de su espacio físico, reducir el consumo de energía y agua, manejo eficiente de residuos y basuras, manejo eficiente de recursos tecnológicos y mejor distribución del personal docente para los cursos. Adicionalmente, otras necesidades relevantes para el planteamiento y que permiten reducir los costos fijos están orientados a la operación diaria de la universidad. Mejorar los procesos internos de la institución para atender solicitudes de manera rápida y oportuna, medición de indicadores internos como apoyo a la toma de decisiones, índice de deserción académica pueden ayudar a comprender el contexto actual y abrir la puerta a nuevos enfoques de una mejor universidad.

La institución ha experimentado entre los años 2005 al 2015 un crecimiento¹ aproximado de 506%, llegando a alcanzar en 2015 una población estudiantil de 7.434 estudiantes, distribuidos en los diversos programas de la institución que son atendidos en sus diferentes sedes. Esta configuración genera una interacción constante y alta entre todos los participantes de la comunidad, como son los estudiantes, personal administrativo, egresados, profesores, investigadores y demás funcionarios.

No obstante, el crecimiento acelerado de la UNIAJC no se ha dado a la par con la preparación de los entes organizacionales para atender a la población en sus diversos trámites y servicios universitarios. Esto genera impactos negativos en las capacidades de gestión organizacional, en la administración de recursos y en la prestación de los servicios que se pueden evidenciar en las siguientes situaciones problemáticas:

¹ Fuente: Estadísticas de crecimiento año 2010 al 2015 del Anuario Estadístico 2015.

- Procesos con baja eficiencia dado las demoras en los trámites y servicios institucionales.
- Obsolescencia en los procesos académico-administrativos y desaprovechamiento del talento humano.
- Incremento en el índice de repetición de materias y deserción académica².
- Infraestructura física no aprovechada al máximo teniendo periodos completamente no utilizada.
- Impactos ambientales negativos debido del incremento del consumo energético.
- Impactos ambientales negativos debido a incrementos del consumo no consciente de agua y/o desperdicio de la misma.
- Impactos ambientales negativos debido a mal manejo de residuos y desechos.
- Mayor demanda de recursos tecnológicos y conectividad por el incremento poblacional.
- Mayor demanda de canales de comunicación y formas de interacción entre los diferentes estamentos.

A continuación, se analizarán dichas situaciones problemáticas con mayor detalle, respaldando las mismas con datos y hechos objetivos, producto del informe final de investigación de la etapa I de este proyecto.

Demoras en los trámites y servicios institucionales

La complejidad sistémica de la UNIAJC generada por el crecimiento acelerado de su población multiestamentaria y la diversificación de los servicios educativos, implica un alto volumen de solicitudes de servicios complementarios, tanto académicos como administrativos. Así mismo, se ve impactado el proceso de gestión documental que en algunos casos es una dolencia latente para la institución. Como consecuencia lógica se ve afectada la prestación de servicios y tiempos de respuesta, generando inconformidad, hecho respaldado en la encuesta de satisfacción del servicio³, realizada a los estudiantes de la

² Es un desertor todo estudiante que no presenta matrícula durante dos periodos consecutivos o más al momento del estudio. SPADIES permite tres diferenciadores de desertores: desertor de programa, desertor de la IES y desertor del sistema.

³ Fuente: Encuesta de Satisfacción del servicio realizada en el año 2015 por la Oficina de Gestión de la Información, adscrita al Departamento de planeación de la Institución Universitaria Antonio José Camacho.

institución, donde se determinó que el 65% tiene una sensación de satisfacción frente a los trámites institucionales y el 19.25% restante no. Esto genera una alerta dado que los valores establecidos por los entes de control indican que deben estar por encima de 80% de satisfacción.

De igual manera, un estudio de eficiencia en la distribución presupuestal del programa de retención estudiantil en la UNIAJC, generado por González López (2014), en el que por medio de 698 llamadas telefónicas se identifican las causales de ausencia intersemestral, arrojó como resultado que las causas de mayor impacto son: económicas, académicas e institucionales (que para este estudio correspondían a infraestructura y atención) como muestra la Figura 1, siendo la segunda causa más relevante la atención al público con 24% de impacto.

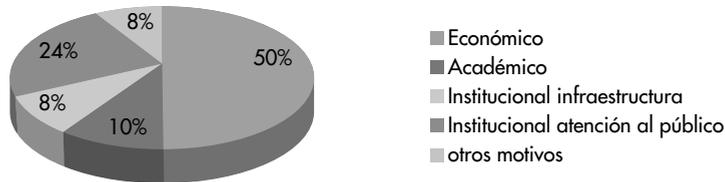


Figura 1. Causas de Ausencia Intersemestral período 1-2013.

Fuente: Informe final proyecto Ecosistema Smart Campus fase I.

Por lo anterior, se hace inminente la necesidad de implementar políticas y normas que exigen el mejoramiento de la eficiencia organizacional, en la prestación de servicios y trámites, enmarcando la eficacia de la gestión documental del sistemas de gestión de calidad, modelo estándar de control interno y políticas de estado de obligatorio cumplimiento como son: eficiencia administrativa y lineamientos de la política de cero papel en la administración pública⁴. gobierno en línea⁵, Ley anti trámites⁶. Todo esto pretende aligerar los procesos internos optimizándolos en alguna medida y generando una mejor sensación de conformidad con la atención al público.

⁴ Directiva Presidencial 04 del 03 de abril de 2012: las entidades deberán identificar y aplicar buenas prácticas para reducir el consumo de papel.

⁵ El Gobierno Electrónico se define como el uso de las Tecnologías de la Información y las Comunicaciones (TIC) para mejorar los servicios e información ofrecidos a los ciudadanos, aumentar la eficiencia y eficacia de la gestión pública e incrementar la transparencia del sector público y la participación ciudadana Consultado en: http://europa.eu/legislation_summaries/information_society/strategies/124226b_es.htm

⁶ Decreto 19 de 2012. Ley Anti-trámites Por el cual se dictan normas para suprimir o reformar regulaciones, procedimientos y trámites innecesarios existentes en la Administración Pública.

Obsolescencia en los procesos académico-administrativos y desaprovechamiento del talento humano

Pese a que la institución ha realizado un gran esfuerzo en la automatización y en la sistematización a través de la modernización de las plataformas tecnológicas instaladas, todavía hay ausencia de sistemas de información efectivos y tácticos que permitan el apoyo a la toma de decisiones de manera oportuna. Esto hace que el trabajo aún sea muy manual y se desaprovechan las capacidades del talento humano que compone el plantel, lo cual es resultado de la obsolescencia de procesos administrativos basados la comunicación organizacional constante que sincronice de manera adecuada los distintos estamentos de la UNIAJC.

Esto genera una gran alerta para la institución. Al tener procesos académico-administrativos poco eficientes, la operación se ve muy afectada teniendo que afrontar el día a día mediante operaciones que dependen de las personas en su mayoría, impidiendo que la universidad se concentre en una operación más enfocada en la planeación estratégica. Sumado a esto, se genera también una tasa baja de respuesta al estudiante, lo que afecta la imagen de la universidad.

Repitencia y deserción académica

La población académica que compone el cuerpo estudiantil de la UNIAJC en un 82,66% pertenece a los estratos 1, 2 y 3. Esto genera unas condiciones previas un poco desfavorables para su paso a los estudios profesionales con respecto al restante porcentaje poblacional (Ministerio de Educación, 2016). Esta población se caracteriza por presentar deficiencias académicas en su formación como bachilleres, lo que obedece a que en muchos casos fue desarrollada mediante procesos educativos acelerados o en Instituciones de Educación Media (IEM) con bajo desempeño en pruebas Saber 11. Adicionalmente y por diversas razones, algunos se ven obligados a postergar sus estudios empezando de manera tardía. Como una muestra de esta afirmación, se pueden revisar los resultados de pruebas Saber 11 para quienes iniciaron sus estudios en la UNIAJC del año 2015.

El 60.18% registró un resultado Bajo, el 34.5% Medio y solamente el 4.6% logró un resultado destacado como Alto. Esto es una medida que persiste en el tiempo y que la institución debe afrontar como una oportunidad para mejorar la formación académica de sus integrantes, pero a la vez genera un gran reto para la nivelación de conocimientos evitando la deserción por dificultades en el aprendizaje, sobre todo en los primeros semestres.

Esto es una gran apuesta de la UNIAJC por el desarrollo de la región y del país. Aquí la institución afronta la necesidad de crear estrategias académicas y de formación efectivas que permitan generar la permanencia de los estudiantes a lo largo de sus formaciones profesionales, motivándolos e incentivándolos a integrarse dentro de una comunidad que está enfocada en la excelencia y la mejora continua.

Para esto, una de las mejores formas de apoyar las construcciones de estrategias es medir, en alguna manera, diferentes factores claves como son: el índice de estudiantes con dificultades que ingresa semestre a semestre, dificultades más comunes, dificultades menos comunes, temáticas por materias que presentan mayor dificultad, tiempo de desactualización de conocimientos generado por la postergación de los estudios una vez terminada la educación media, para quienes ingresan por primera vez a la institución. Adicionalmente, no se puede dejar por fuera a quienes ya forman parte y están formándose como profesionales, requiriendo todo el apoyo disponible para continuar con sus procesos formativos. Como se mencionó antes, este apoyo es muy necesario en especial en los primeros semestres donde se debe medir constantemente el índice de estudiantes que repiten materias, cuáles son esas materias donde más repiten y qué conocimientos previos se requerían. Todo esto lleva finalmente a buscar tener datos que permitan poder generar estrategias efectivas como cursos intermedios o nivelatorios, planes de curso adaptados a las características poblacionales, didácticas pedagógicas efectivas para el aprendizaje, tutorías constantes, etc. Finalmente, todo esto debe buscar en su máxima posibilidad evitar la deserción, pero en los casos donde no es posible, también se deben medir las razones del por qué ha sucedido y plantear estrategias que intenten reintegrar a los estudiantes.

Desaprovechamiento de la infraestructura física en jornadas de baja ocupación

La UNIAJC cuenta con espacios donde confrontar lo teórico en la práctica, estos son los laboratorios y las salas de sistemas. La problemática radica en el aprovechamiento de estos espacios, debido a las limitaciones de disponibilidad y espacio. Para poder hacer uso de un laboratorio o de una sala, las materias que lo requieren fijan unos espacios de clase y separan su horario durante el semestre. Sin embargo, cuando el estudiante requiere un proceso de refuerzo, o generar un trabajo no dirigido, se dificulta el uso de estos espacios por la misma asignación de clases. Por otra parte, los cursos

que tienen demasiados estudiantes presentan dificultades para que todos puedan interactuar de manera adecuada en sus prácticas con los elementos disponibles, a lo que se suma un bajo control del inventario. Claramente la institución afronta un reto y debe buscar estrategias que permitan hacer un uso de la infraestructura disponible maximizando su potencial.

El no uso óptimo de los espacios de la institución genera impactos ambientales como: consumo energético incrementado e innecesario, mayor producción de residuos, consumo incrementado de agua e incremento de residuos de todo tipo. En este sentido, la universidad desde su Plan Estratégico de Desarrollo también tiene un compromiso con la conciencia ambiental y busca minimizar el impacto a través de estrategias formativas, pero también correctivas.

Mayor demanda de recursos tecnológicos y conectividad

El crecimiento poblacional de la institución Universitaria Antonio José Camacho frente a sus diversos servicios académicos ha generado una demanda constante de mayores recursos tecnológicos tanto para la operación como para la conectividad. Es importante recordar que la institución cuenta con una sede en el sur, ubicada en el Km 7 de la vía Cali-Jamundí, junto a Parquesoft, y una sede principal, ubicada en la avenida sexta con 28N en el Norte, sector donde también cuenta con las edificaciones Estación 1 y Estación 2, además de su presencia en la Fundación para el Desarrollo de la Institución Universitaria Antonio José Camacho y en otros municipios. Por tanto, la UNIAJC requiere de más infraestructura para sus laboratorios y salas de práctica, ampliación de las zonas Wifi, ampliar la velocidad de conexión a internet y conexión con redes académicas de investigación disponibles a la comunidad como RUAV (Red Académica Avanzada del Valle del Cauca) y RENATA (Red Nacional Académica de Tecnología Avanzadas).

La Institución consiente de esto ha caracterizado en diversas oportunidades este tema, siendo la más reciente la generada por la Oficina Asesora de Planeación (2015) que ha generado una estadística en la cual, entre otros factores, se muestra la medición de evolución de los recursos tecnológicos en la institución a través del tiempo. Ver Ilustración 2.

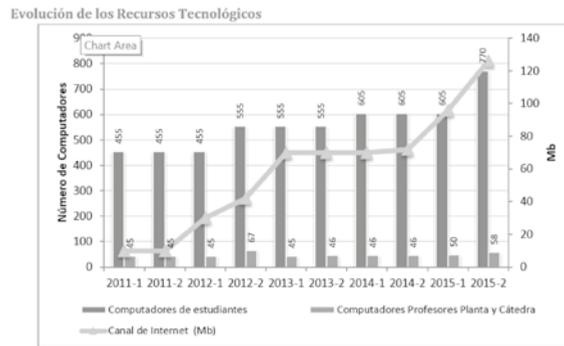


Figura 2. Evolución de los recursos tecnológicos, conexión a internet, computadores de estudiantes en los últimos 5 años.

Fuente: Anuario Estadístico 2015. Oficina de Planeación Institucional de la UNIAJC

La ilustración muestra la respuesta de la institución en su esfuerzo por cubrir esta necesidad, pero es claro también que la demanda es constante y los esfuerzos no cesan, pudiendo ser sobrepasados en cualquier momento. Una de las necesidades más grandes radica en la parte operativa de la institución, que requiere mayor demanda de canales de comunicación y formas de interacción entre los diferentes estamentos, compartiendo información de manera automática y efectiva para agilizar sus procesos, lo que es común con varios de los retos y problemáticas mencionadas. Por otra parte, la prestación de servicios debe ser capaz de responder a la oferta académica institucional compuesta por 31 programas académicos de educación superior, 4 de formación para el trabajo y desarrollo humano, formación en segunda lengua y programas de extensión, distribuidos en diversas sedes e incluso municipios.

Acorde a lo mencionado hasta este punto sobre las diversas problemáticas, como respuesta la UNIAJC en define el Plan Estratégico de Desarrollo para la vigencia 2020-2030 (UNIAJC, 2020), donde se consignan como objetivos estratégicos encaminados a responder a estas necesidades los siguientes:

- Establecer una oferta académica alineada con las exigencias del desarrollo productivo, social y cultural de la región, con base en la pertinencia y la calidad académica.
- Fortalecer las capacidades institucionales para desarrollar investigación pertinente y relevante para la comunidad académica y la sociedad en general

- Incrementar el impacto de la proyección social aprovechando el reconocimiento de la institución en el entorno
- Desarrollar el potencial de la capacidad institucional para el bienestar de la comunidad universitaria.
- Consolidar un cuerpo profesoral suficiente y cualificado para la docencia, la investigación y la proyección social, comprometido con la alta calidad y las metas institucionales.
- Fortalecer la gestión administrativa y financiera para la consolidación económica en un marco de eficiencia organizacional.
- Desarrollar la infraestructura física, tecnológica y medios educativos para responder a las actividades misionales, de bienestar y administrativas.
- Modernizar la estructura organizacional y los procesos académicos y de gestión a través un modelo de universidad inteligente, soportada en servicios y recursos digitales.

Siendo consecuentes con esto, la Mg. Ana Milena Rojas Calero plantea el proyecto Ecosistema Smart Campus, encaminado a dar respuesta a estas problemáticas desde la consecución de soluciones que permitan alcanzar los objetivos estratégicos propuestos y apoyada en un equipo de profesores investigadores, becarios y estudiantes capaces de afrontar estos retos.



Antecedentes

La búsqueda de la mejora continua a nivel de la comunidad académica está presente en diversos procesos investigativos. Uno de ellos evoca la posibilidad de que, adoptando la tecnología como eje de la evolución, las instituciones educativas puedan transformar su modelo actual de servicios Maciá, Berná, Sánchez, Lozano, & Fuster, (2016). Así, la Universidad de Alicante, consciente de que no existe un modelo estándar para realizar esta labor, decide dar sus primeros pasos entendiendo el concepto de Smart como la aplicación del uso de las tecnologías en pro del bienestar de la comunidad.

En complemento, Maciá, Berná, Sánchez, Lozano, & Fuster (2016) proponen, unido al término anteriormente descrito, una definición de Smart University como: "un modelo de universidad que mejora la calidad de vida haciendo uso intensivo, global, eficiente y sostenible de las TI para interconectar todos los actores y servicios en beneficio de toda la comunidad". Esto abre un panorama ideal para el campo investigativo, debido a la diversidad de proyectos que pueden resultar producto de esta definición.

Asimismo, el trabajo de Maciá, Berná, Sánchez, Lozano, & Fuster (2016) propone que un campus universitario puede funcionar como una réplica pequeña de una ciudad, donde distintos entes interactúan en pro de la comunidad buscando la sostenibilidad, eficiencia y calidad de los servicios brindados a la comunidad. Por lo anterior, y en concordancia con European Commission (2010), quien promueve el desarrollo de las comunidades inteligentes y la inversión en tecnología para potenciar el desarrollo social y humano, la Universidad de Alicante decide dar el aval e iniciar con esta monumental tarea.

Si bien existen muchas tecnologías aplicables a casi cualquier contexto, Maciá, Berná, Sánchez, Lozano, & Fuster (2016) determinó por dónde debía iniciar la investigación y qué aspectos debían ser cubiertos de manera prioritaria. Uno de los aspectos iniciales a abordar, dada la variedad de proyectos que pueden surgir, fue la conectividad pertinente a nivel de infraestructura. La tecnología como eje fundamental del éxito de los proyectos requiere de una infraestructura física preparada para soportar una alta conectividad de los recursos, información, personas y aplicaciones que empiecen a interactuar en esta comunidad. Por lo anterior, la escalabilidad, mantenibilidad y operatividad del sistema estará soportada bajo la premisa de una alta conectividad, un diseño correcto de la conexión entre sistemas, un gobierno federado de aplicaciones y una amplia capacidad de mantenimiento de los

canales de integración y comunicación. Por ello, Maciá, Berná, Sánchez, Lozano, & Fuster (2016) señalan que las principales ventajas de una correcta implementación a nivel de infraestructura radican en la facilidad de acceso a los datos, la disponibilidad de la información cuando es requerida, el procesamiento rápido de la misma como apoyo a la toma de decisiones, la transmisión rápida, el almacenamiento confiable de información con capacidad de respaldo y recuperación a fallos, el análisis constante y la interactividad. Todo esto incrementa significativamente la capacidad de tomar decisiones de manera oportuna, minimizando el reproceso producto de labores con mucha interacción humana de por medio. Así mismo, se genera una mejor manipulación de la información, haciéndola más segura y confiable.

Como mencionan Maciá, Berná, Sánchez, Lozano, & Fuster (2016), hablar de tecnología no sólo implica la infraestructura que soporta el proyecto, también es necesario hablar de los aplicativos, plataformas y sistemas de información que apoyan la gestión de la institución. Este podría ser un campo muy grande para explorar, pero los mismos autores proponen una forma adecuada de abarcar los múltiples proyectos que pueden resultar, para quienes deseen seguir sus pasos. Ellos proponen 6 ejes en los que se puede dividir el trabajo de acuerdo a unos intereses específicos de impacto sobre la comunidad. A continuación, se listan, pero en la siguiente sección es donde se dará un breve detalle:

- Smart People: se enfoca en impactar a las personas, su interacción y los distintos medios de comunicación que tienen actualmente.
- Smart Government: busca apoyar la transparencia en las decisiones del gobierno de la institución, facilitando el acceso a la información (Open Data).
- Smart Mobility: se enfoca en la mejora de las condiciones de movilidad para la comunidad.
- Smart Economy: se enfoca en el apoyo a la toma de decisiones económicas, fomentando nuevos modelos de negocio, refinando los actuales, impulsando la mejora continua a partir de la disponibilidad de la información.
- Smart Living: se enfoca en los estilos de vida de la comunidad y en entregar herramientas que mejoren su calidad de vida.



Modelo Propuesto

El modelo propuesto se basa en el trabajo de Maciá, Berná, Sánchez, Lozano, & Fuster (2016), que separa los proyectos a abordar dentro del Smart Campus en los 6 ejes descritos anteriormente. Antes de exponer los proyectos propuestos por eje, es necesario abordar el modelo de trabajo del macroproyecto Ecosistema Smart Campus de la UNIAJC y a sus colaboradores.

Equipo de trabajo Ecosistema Smart Campus-UNIAJC

El proyecto está encabezado por la Magister en Informática y Telecomunicaciones de la Universidad ICESI, Ana Milena Rojas Calero, docente de carrera, perteneciente a la Facultad de Ingenierías e investigadora del grupo Grintic. Junto con ella se encuentran trabajando durante la fase I: Mg. Hugo Alberto González López, rector de la institución; Mg. Juan Carlos Cruz Ardila, decano de investigaciones y PhD. Leandro Flórez Aristizábal, director del grupo de investigación Grintic. Durante la misma fase, de manera no oficial, participó el MSc. Manuel Alejandro Pastrana, profesor de la misma facultad, quien en segunda fase pasa a ser parte del equipo de trabajo.

En este equipo, se encuentra además un grupo de becarios, quienes participaron durante las dos etapas del proyecto. A continuación, se mencionan como agradecimiento a su valioso esfuerzo:

- Angie Melissa Tobar Bolaños
- Camilo Andrés Muñoz González
- Fernando Andrés Cifuentes Calderón
- Jonathan Arley Rodríguez Pacheco
- Manuela Cardona Arias
- Nasly Katherine Escobar Ortiz
- Nicolás Bermúdez Rodríguez
- Nicolás Grisales Villanueva
- Nicolás Ortiz Acevedo
- Raúl Andrés Hernández Ocampo
- Ronal Andrés Tamayo Zapata
- Rubens Díaz Pullí

Proceso de desarrollo de software Ecosistema Smart Campus

La metodología de desarrollo de software de Smart Campus toma, a criterio de los investigadores, la mejor configuración de roles, técnicas, eventos, artefactos, herramientas y buenas prácticas de diversos marcos de trabajo como XP, SCRUM, DevOps y de metodologías como PMI para la gestión de proyectos. Inicialmente, en el trabajo de Pastrana, Ordoñez, Rojas, & Ordoñez (2019) Springer Nature Singapore Pte Ltd. *Compliance with the client requirements is a key factor for the success of projects. However, this compliance is not always easy to monitor and verify. SCRUM framework flexibility makes it possible to include a variety of techniques and good practices from other methods to help software development teams to achieve their goals. Some of these tools can track historical changes in the code versioning* (El cumplimiento de los requisitos del cliente es un factor clave para el éxito de los proyectos. Sin embargo, este cumplimiento no siempre es fácil de monitorear y verificar. La flexibilidad del marco SCRUM permite incluir una variedad de técnicas y buenas prácticas de otros métodos para ayudar a los equipos de desarrollo de software a lograr sus objetivos. Algunas de estas herramientas pueden rastrear cambios históricos en el control de versiones del código), se describe el proceso de desarrollo basado en scrum, que utiliza las buenas prácticas de XP orientadas a la implementación de algunas herramientas que garantizan la calidad preventiva mediante el enfoque DevOps y, finalmente, se describen los artefactos y prácticas tomadas de PMI para garantizar el éxito de los proyectos.

Scrum es un marco de trabajo muy útil a la hora de ejecutar proyectos de todos los tamaños y niveles de complejidad. Se centra específicamente en generar una cultura de trabajo colaborativo, donde la comunicación entre todos los interesados es el eje fundamental para obtener retroalimentación lo más rápido posible y aprender constantemente como lograr mejores resultados. A su vez, aunque es muy fácil de contar como funciona, es un poco complejo de implementar (Schwaber & Sutherland, 2017). Cada equipo tiene una configuración cultural, una experiencia propia y una forma de trabajo. Por lo mismo, puede resultar complicado implementar todo scrum dentro de un proceso de desarrollo. Por lo anterior, Schwaber & Sutherland (2017) denominan a scrum como un marco de trabajo y no una metodología, debido a que quien busca la implementación de scrum dentro de un proceso de desarrollo, debe tener la suficiente habilidad para determinar qué incluir o no dentro del proceso por cada fase del ciclo de desarrollo descrito por Pressman & Maxim (2015). A continuación, se detallarán los roles, artefactos, eventos y prácticas tomadas de scrum y las buenas prácticas de desarrollo de software de XP.

Roles dentro del proyecto

El Ecosistema Smart Campus adopta los roles basados en scrum, con responsabilidades específicas bien definidas, de acuerdo a lo descrito por Schwaber & Sutherland (2017) y un rol extra, que es el gerente de proyecto, quien cumple con orientar el proyecto a fin de que este tenga los resultados esperados de acuerdo a la visión institucional. A continuación, se describen cada uno:

Product Owner

Según el enfoque de Schwaber & Sutherland (2017), este rol es ejecutado por quien tiene más clara la visión del producto. La responsabilidad principal es la de representar a todos los stakeholder a través de su opinión y visión. Centra su esfuerzo en ayudar a que todos los integrantes del proyecto entiendan las necesidades funcionales a resolver, la importancia o prioridad que tienen para el cliente; ayudar a pactar los compromisos, ordenar el trabajo, a realizar y proveer a los otros roles de todo lo requerido para que puedan realizar el proyecto de manera adecuada cuando lo que necesitan está del lado del cliente. Este rol no es un gerente de proyecto, ni un dueño del equipo. Tan sólo es quien ayuda a orientar la visión para que se cumpla.

Scrum Master

Es un integrante del scrum development team, con una responsabilidad específica: "eliminar cualquier impedimento para lograr las metas del sprint", como indican Schwaber & Sutherland (2017). Este rol es quien mejor conoce scrum y hace que se cumpla. Se encarga de organizar todos los eventos de scrum (sprint, planning meeting, daily scrum, sprint review, sprint retrospective) y apoyar a que se cumplan adecuadamente. Apoya al equipo con sus problemáticas como solicitudes de información al cliente, aclaración de dudas, pero también con cosas simples como ubicación cómoda del espacio de trabajo, material para trabajar, e incluso preparación técnica. Adicionalmente, corrobora que los artefactos fundamentales de scrum se construyan de manera adecuada (product backlog, sprint backlog y burndown chart).

Scrum Development Team

Schwaber & Sutherland (2017) indican que este rol es quien construye el producto con la ayuda de los otros dos roles. Su objetivo principal es lograr el cumplimiento de los objetivos del sprint. Se conforma por equipos auto-gestionables que

organizan su trabajo de manera adecuada y son capaces de trabajar enfocados en compromisos y no en horas. Son los responsables de construir junto con el product owner y bajo la orientación del scrum master los artefactos del proyecto, incluyendo la unidad desplegable para poner en producción.

Gerente del Proyecto

Este rol está enfocado en dirigir de manera adecuada el Proyecto, supervisando el cumplimiento de los compromisos de acuerdo a la planeación entregada por el equipo y supervisando que las entregas cumplan con lo esperado por la institución. Este rol normalmente lo cumple el director de proyecto de grado, quien es uno de los coinvestigadores inscritos dentro del proyecto.

Análisis y planeación

Según Pressman & Maxim (2015), esta etapa consiste en la identificación de las necesidades funcionales de los sistemas a construir, que formalmente se denomina elicitación de requisitos. Aquí los encargados de recopilar la información utilizan diversas técnicas que permiten formalizar lo indagado en un artefacto (documento que cumple con características específicas de acuerdo a un estándar o a una convención).

Las técnicas descritas por el mismo autor son la entrevista, la encuesta, la lluvia de ideas, pero Smart Campus, basado en el trabajo de (Pastrana, Ordóñez, Ordóñez, & Merchan (2017), decide adoptar la técnica inception deck.

Técnica de elicitación de requisitos utilizada

La técnica creada por Rasmusson (2010) consiste en una serie de 10 dinámicas que permiten al auditorio unificar la visión del producto a partir del consenso. A continuación, usando el trabajo de Pastrana et al. (2017), se describe cada etapa:

Initial activity: Who is in the auditorium?

I. Why are we here? And who's in the audience? Estas dos dinámicas están centradas en identificar si todos conocen la razón de la reunión y tienen algún grado de conocimiento sobre la visión del producto. Adicionalmente, confirma que estén los interesados que mayor aporte puedan generar a la reunión.

II. Creating an Elevator Pitch. En caso de no conocerse la visión del producto se debe crear un pequeño discurso que resalte en menos de 5 minutos las ventajas

y beneficios del proyecto, él porqué es necesario y el a quién está dirigido.

III. Design a product box. Esta dinámica permite centrarse en el nombre del producto, una frase que lo identifique y el posible logo. Se parte del símil con la frase: "Si el producto estuviera terminado y puesto en una vitrina de ventas como se vería".

IV. Create a list of "what is not". Esta dinámica está diseñada para delimitar el alcance del proyecto en todos los sentidos posibles, desde funcional hasta tecnológico. Siempre que pensamos en negativo nuestro cerebro instintivamente debe recurrir a identificar que sí es para poder delimitar esta visión.

V. Meet your neighborhood. En caso de que las personas del auditorio no conozcan quienes estarán participando dentro del proyecto, se realiza una breve presentación de cada uno indicando su rol, aspectos a aportar, cosas que no gustan del producto y datos de contacto.

VI. What keeps us up at night? Preocupaciones. Son los riesgos que están desde el inicio del proyecto y deben ser atendidos cuanto antes. Vital para un buen comienzo.

VII. Show the solution: 4 formas de mostrar la solución, donde cada una tiene un enfoque particular.

- a. Give the app some personality
- b. Let's make it flow
- c. Wireframing
- d. Stories map

VIII. What's it going to take? Compromisos de los participantes para que el Proyecto salga a flote.

IX. How much will it cost? ¿Qué retos existen en aspectos no económicos? ¿Pruebas de concepto? ¿Investigación? ¿Cambios internos de la empresa? Son preguntas constantes que deben hacerse para resolver este punto.

X. Summary. Se presenta el resumen de los resultados

Artefacto de la etapa de análisis: Product Backlog

El resultado final de la técnica aplicada es el artefacto fundamental para el proceso de desarrollo de software denominado Product Backlog. Según Schwaber & Sutherland (2017), creadores de scrum, este documento recopila la información funcional con el detalle suficiente a través de un modelo de historias de usuario que permite su escritura. Es importante resaltar que para la ingeniería de software no existe un modelo estándar de historias de usuario hasta la fecha, por tal razón, el equipo decide adoptar el modelo de (Cohn, 2004), que a través de la siguiente frase permite fácilmente seguir una estructura para la escritura de la información: yo como [ROL]+ requiero/espero/deseo/necesito [una funcionalidad] +para [motive]+con los siguientes [Criterios de aceptación]. Con base en esto se crea la siguiente estructura base de plantilla para la recopilación de las historias de usuario:

Tabla 1: Modelo inicial de Product Backlog. C.A = Criterios de Aceptación

ID	Rol	Funcionalidad	Motivo	C.A.
	Yo como [Rol]	Requiero [Funcionalidad]	Para [Motivo]	

La identificación del ID permite referenciar rápidamente la HU sin tener que entrar a leer todo el detalle. Esto facilitará ejercicios como la estimación y la priorización en la construcción de la planeación de sprint backlogs. La columna Rol indicará al equipo rápidamente quien ejerce la acción. Es importante no confundir el sistema como un rol ya que este no lo es. La columna funcionalidad expresa siempre la acción a ejecutar. Una regla común a la hora de crear un lenguaje común es que la funcionalidad siempre se exprese como un verbo en infinitivo (terminado en ar, er, ir). Esto facilita garantizar que la historia de usuario es atómica, es decir, que solamente responde por una funcionalidad a la vez. Frecuentemente, una confusión que se da respecto a este aspecto es si se pueden utilizar o no verbos como gestionar que agrupa 4 funcionalidades dentro de sí (crear, modificar, consultar, eliminar). La recomendación que da tanto Cohn (2004) como Pastrana et al. (2017) es que no, debido a que esto incrementa el número de criterios de aceptación haciendo muy difícil de leer y comprender la HU, además de aumentar la complejidad de la misma en el momento de realizar la estimación.

Cohn (2004) sugiere que las HU deben llevar en su estructura una justificación o motivo que lo da el para. Cuando un interesado del proyecto puede justificar

por qué necesita la HU significa que tiene claro su valor de negocio, pero cuando le cuesta trabajo significa que realmente no está muy claro qué quiere hacer y es deber del equipo ayudarlo a encontrar dicho valor o descartar la HU. Finalmente, los criterios de aceptación dan el detalle adecuado para poder comprobar si se ha cumplido con una HU. Algunas reglas útiles para escribir criterios de aceptación son: ¿dónde sucede la acción?, ¿qué componentes de pantalla hay?, ¿campos, botones, listas, paneles colapsables, imágenes, etc?, ¿tienen alguna restricción estos campos como tamaño, formato archivos a utilizar, columnas a mostrar en una tabla?, ¿tienen obligatoriedad?, ¿qué mensajes de error y de éxito se deben mostrar? Recordar que no existe un nivel de detalle estándar para los criterios, pero entre más claro sea, más fácil será su implementación.

Una vez este modelo base de la documentación del análisis de requisitos es construido, se da paso a su completitud, adicionándole los campos requeridos para la estimación y priorización bajo técnicas ágiles y un campo extra de observaciones que permite durante la etapa de refinamiento de las historias adicionar comentarios para ser resueltos previos a la estimación.

Tabla 2: Modelo final de Product Backlog. C.A = Criterios de Aceptación. C = Complejidad. P= Priorización. Obs = Observaciones

ID	Rol	Funcionalidad	Motivo	C.A.	C	P	Obs
	Yo como [Rol]	Requiero [Funcionalidad]	Para [Motivo]				

Lo anterior deja entrever que las historias de usuario son susceptibles a refinamiento constante, debido a que están escritas en lenguaje natural y su interpretación puede variar por quien las lee a menos de que se construyan bajo un lenguaje común y pactado previamente entre todos los interesados del proyecto. El impacto de la ambigüedad está contemplado y reducido en alguna medida, a partir del uso de la técnica inception deck y del modelo de historias de usuario utilizado. Adicionalmente, bajo la sugerencia de Pastrana et al. (2017), en algunos proyectos con gran complejidad, el uso de modelamiento de procesos de negocio se utiliza como complemento de las HU para garantizar una mayor claridad de la información.

Adicionalmente, también es posible entender a este punto que la actividad de estimación y priorización es complementaria y posterior a la elicitación de requisitos. Por lo que el equipo primero debe entender el contexto y todo

lo que se requiere previo a afrontar cómo podrían planear los compromisos del proyecto y las entregas parciales. A continuación, se da un ejemplo de una historia de usuario construida para un proyecto aplicando todo lo indicado anteriormente:

Id	Rol	Funcionalidad	Motivo	C:A.	C	P	Obs
	Yo como administrador	Requiero ingresar a la plataforma de manera segura	para garantizar que las acciones del administrador no las puede ejercer ningún otro rol	1. la pantalla contiene el campo usuario. Obligatorio 2. La pantalla contiene el campo contraseña. Obligatorio 3. la pantalla tendrá un botón "Acceder" que permite enviar los datos para ser validados 4. Al presionar el botón, el sistema valida que los campos usuario y contraseña no estén vacíos, en caso contrario, el sistema desplegará el siguiente mensaje "Favor ingresar campos obligatorios". Adicionalmente, los campos dejados en vacío se resaltarán con color rojo y un asterisco que indique son esos los campos a corregir. 5. Cuando los datos son correctos el sistema los envía al servicio de autenticación de smart campus y espera la validación. si es correcta da acceso con el rol administrador. En caso contrario indica el siguiente mensaje: "Usuario o contraseña incorrecto, por favor volver a intentar"			Usuario y contraseña son los mismos de Academusoft

Figura 3. Ejemplo de HU aplicando. C:A = Criterios de Aceptación. C = Complejidad. P= Priorización. Obs = Observaciones

Se recomienda al lector dar una revisada al **anexo a**, adjunto a este libro para que pueda tener un modelo a seguir de acuerdo a la plantilla generada por el equipo de trabajo.

Técnica de estimación de requisitos utilizada

El proceso de estimación bajo la perspectiva de los marcos de trabajo ágil se enfoca en entender que el tiempo es una variable que cambia tanto como los

diversos factores con los que un equipo afronta el proyecto. Aspectos como la experticia del equipo de desarrollo, que tanto se conocen entre sí, que tanto dominan las tecnologías del proyecto, que tan buenas máquinas de desarrollo tienen, que tanta calidad deben incluir, si conocen las tecnologías de calidad, las técnicas y las medidas de calidad, etc., generan variabilidad en el tiempo que toma construir un proyecto. Por tanto, los modelos ágiles indican que las medidas de tiempo que se utilizan en los métodos predictivos intentan predecir algo que sin una continuidad de datos históricos formalizados a través de fórmulas no se podrían predecir.

Para dar un ejemplo simple de la pretensión del agilísimo, se plantea el siguiente escenario.

La distancia entre un punto A cualquiera y B es de 20 km.

$\overline{\hspace{1.5cm}}$
 a 20 KM B

Si una persona decidiera recorrer esa distancia a pie, caminando podría tomarle unas 3 horas aproximadamente. Pero si dispone de buen estado físico y decide trotar, podría tomarle alrededor de 1,5 horas. La mitad del tiempo inicial. Si decidiera, en lugar de trotar, tomar una bicicleta y recorrer esa misma distancia, probablemente en 45 minutos lo haría. Claro está, si lo hace en un vehículo más rápido como un carro o una moto, su tiempo sería mucho menor.

Así mismo funcionan los proyectos de desarrollo de software. El vehículo sería la tecnología a usar: lenguajes de programación, frameworks, generadores de códigos, IDE o entornos de desarrollo y herramientas de calidad preventiva. Entre más dominio tengamos de estos, el tiempo de desarrollo será menor, pero si desconocemos varios de estos el tiempo de desarrollo será mucho mayor.

Por lo anterior, los modelos ágiles no se centran en calcular el tiempo sino la dificultad a resolver bajo una percepción subjetiva que solamente puede ser dada por el equipo de desarrollo.

Con respecto al tiempo que requieren los modelos tradicionales de gestión de proyectos y los cálculos gerenciales de métricas de control de productividad, hay una solución muy simple y es el uso de compromisos fijos por sprints.

Según Schwaber & Sutherland (2017), un sprint se le denomina a un periodo de tiempo fijo que puede tener una duración de 1, 2, 3 o 4 semanas. Ese periodo de tiempo es en el que se divide el proyecto a realizar y es el mismo durante todo el proyecto, por lo que puede ser inferior de lo mencionado, que un proyecto que está compuesto por varios sprints.

Para poder afrontar la estimación, Ecosistema Smart Campus UNIAJC adopta póker scrum. Primero se explicarán las barajas posibles a utilizar y posteriormente se detallará el proceso de estimación para que pueda ser replicado. La técnica consiste en una baraja con unos rangos de número y 3 cartas comodines. Existen dos tipos de baraja muy utilizados a nivel de la industria del desarrollo de software. La figura 3 refleja la baraja original, que utiliza la serie de Fibonacci para la sucesión de números. Es importante recordar que la serie de Fibonacci indica que el valor siguiente es la suma de los dos anteriores. Los valores reflejan la percepción de complejidad asociada a la funcionalidad o en palabras más simples, en esa escala de valores qué tan difícil considera que es hacer la funcionalidad requerida. Por parte de las cartas comodines tenemos tres opciones importantes a revisar. La primera es el símbolo de pregunta (?). Cuando alguien del equipo utiliza esta carta significa que la funcionalidad expuesta no contiene detalle suficiente para comprenderla y poder dar una apreciación de la estimación. Por otra parte, la carta infinita (∞) significa que la funcionalidad expuesta tiene tanta información que su complejidad tiende a ser muy grande. Ambas cartas sugieren al equipo de desarrollo scrum que deben refinar la historia que se está afrontando y darle suficiente detalle adecuado para la percepción de la estimación. Es importante resaltar que refinar una historia de usuario (HU) puede llevar a dos situaciones muy comunes. La primera es que la HU le falte detalle en los criterios de aceptación. La segunda es que la HU tenga tanto detalle en los criterios que la historia probablemente lleve más de una funcionalidad incluida y, por tanto, para conservar su atomicidad esta debe ser descompuesta en varias HU. Se sugiere tener presente que al descomponer un HU en varias, su estimación no debe dar la suma de la estimación anterior, sino que cada HU nueva se debe estimar de manera independiente, como una funcionalidad sin dependencia de otras.

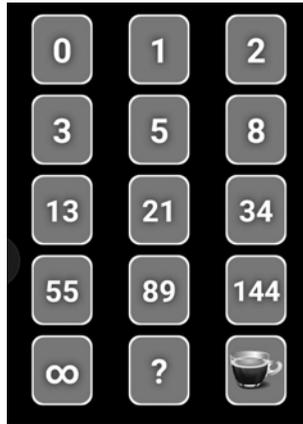


Figura 4. Baraja de Ppker Scrum basado en Fibonacci.

Fuente: Aplicación scrum poker disponible en Google Play

La figura 4 muestra la baraja más aceptada por la industria. Esta baraja contiene la misma definición de la serie de Fibonacci, pero con un rango más acotado de valores que van del 0 al 100. Esta última baraja es la que es adoptada por el equipo de trabajo y utilizada en los proyectos por parte del Ecosistema Smart Campus UNIAJC.



Figura 5. Baraja de Poker Scrum Standar.

Fuente: Aplicación scrum poker disponible en Google Play

Pasos para la estimación por poker scrum

Una vez comprendida la baraja a utilizar, es importante exponer la técnica a fin de tener un pleno conocimiento del proceso de planeación bajo el modelo de trabajo del equipo de Smart Campus. A continuación, se exponen los pasos requeridos para llevar a cabo la técnica con éxito:

Pasos previos

- El product backlog debe contener todas las historias de usuario refinadas y con el detalle suficiente para que el equipo pueda realizar el proceso.
- El equipo de desarrollo scrum debe leer previo a la reunión el product backlog y anotar todas sus dudas o sugerencias previas a la reunión.
- Si hay sugerencias o dudas estas deben ser expuestas al equipo para ser resueltas previas a la reunión.
- El product owner debe estar disponible para consultas en caso de requerirlo.

Inicio de la estimación

Se debe fijar un moderador de la reunión, que normalmente es el scrum master.

Se leen todas las historias de usuario en su definición épica. Recordar que Cohn (2004) indica que una HU épica es la primera parte de la estructura sugerida por él y que no incluye criterios de aceptación (Yo como [rol]+ requiero [funcionalidad] +Para [motive]).

I. Se selecciona la HU que parece la más fácil de hacer y la más difícil. Esto se denomina fijar el piso y el techo de la estimación.

II. Se leen los criterios de aceptación de las HU piso y techo para corroborar que efectivamente son las más fácil y difícil de realizar. Si todo el equipo está de acuerdo se seleccionan.

III. Se estima la HU más fácil o piso. Para la estimación el moderador corrobora que la HU es completamente comprendida. Si no es comprendida, la historia de usuario debe ser refinada inmediatamente. Si es comprendida, cada integrante del equipo de manera secreta selecciona un valor de la baraja del póker scrum, y no comenta este valor hasta que el moderador lo indique. Esto se hace con el fin de no sesgar la opinión de los demás, sino fomentar la participación de todos y llegar a un consenso a través de la diversidad de opiniones. Cuando el moderador indica que todos muestren su selección, todos los participantes muestran la carta de la baraja que han seleccionado. Aquí pueden suceder varias situaciones. La primera y la ideal es que todos tengan el mismo valor en la votación. En este caso

significa que todos están de acuerdo con que el valor de estimación es ese. El segundo caso es que algunos integrantes tengan un valor y otros otro (división del equipo), por lo que el moderador solicita que se expongan las opiniones de todo para poder llegar a un consenso. Puede suceder que en algunos casos el equipo decida adoptar una de las votaciones como el valor de complejidad o dificultad de la HU. En otros casos, donde aún no se han puesto de acuerdo, se solicita volver a estimar. El moderador debe intentar que el máximo número de votaciones que se haga sobre la misma HU no supere el valor de 3. Hay que tener presente que el valor de complejidad asignado a la HU siempre es el consenso del equipo, nunca se hace promedio de los valores. De esta manera queda asignado el valor de complejidad de la historia de usuario más fácil, que se denomina mínimo de la estimación. No existirá un valor más pequeño que el de esta HU.

IV. De la misma manera que se realizó la estimación para la HU más simple se realiza para la máxima.

V. Una vez fue fijado el valor máximo y mínimo de la estimación, se procede a seleccionar un valor intermedio entre esa escala. A ese valor se le denomina pivote. Cuando un equipo tiene dudas sobre una HU, si está sobre estimada o sub estimada, el equipo debe compararla contra el pivote. Con esto tiene un punto intermedio para poder identificar si la historia se acerca más al pivote o se aleja más del pivote hacia alguno de los extremos.

VI. Con los valores de máximo, mínimo y pivote fijados, se procede a estimar todas las demás HU del product backlog.

VII. Con todas las HU estimadas se suman los pesos de todas las HU para realizar el cálculo del peso del proyecto. En la tabla 3 se puede apreciar un breve ejemplo:

Tabla 3. Ejemplo cálculo del peso del proyecto

ID	Estimación
HU-01	1
HU-02	3
HU-03	8
HU-n	13
Total	25

VIII. Con el peso de todo el proyecto calculado, el equipo de trabajo ya puede proceder a calcular el número de sprints que tomará realizar el proyecto. Para esto lo primero que se hace es fijar el tamaño del sprint. Recordar que de acuerdo a Schwaber & Sutherland (2017), un sprint es un periodo de tiempo que va de 1 hasta máximo 4 semanas. Con el mismo tamaño durante todos los sprints del proyecto. Esto quiere decir que, si se define que los sprint serán de 2 semanas, todos los sprints del proyecto tendrán ese tamaño. Con el tamaño del sprint seleccionado, el equipo debe fijar un compromiso de puntos a cubrir durante ese periodo de tiempo. Para identificar el valor más pequeño de compromiso posible a asumir se sugiere que se multiplique el tamaño del equipo por el valor de la historia de usuario más pequeña y este valor por el número de semanas que implica el sprint. Es decir, que si la historia de usuario más pequeña tiene una complejidad de 2, el equipo de trabajo está compuesto por 5 personas y cada sprint dura dos semanas; un valor mínimo de compromiso podría ser $2 * 5 * 2 = 20$. Esto implica que cada persona se compromete a realizar dos puntos de complejidad por sprint. Claro está, este es el escenario de compromiso más bajo posible. Otra forma de calcularlo es utilizando una fórmula similar a la anterior, pero sin multiplicar por el número de semanas del sprint y con la historia de usuario más difícil. En este caso, si por ejemplo la estimación máxima da un valor de 20, existen 5 integrantes; el valor del compromiso del sprint será de 100 puntos a cubrir. Una tercera forma de asumir el compromiso, y que es la más utilizada, es colocar un valor por encima del compromiso calculado con la historia de usuario mínima e ir generando un incremento a medida que el equipo aumenta la velocidad de desarrollo.

IX. Calcular el número de sprints. Cuando se divide el tamaño del proyecto por el compromiso se obtiene el número de sprints que tomará hacer el proyecto realidad.

A continuación, se da un ejemplo más detallado de cómo aplicar la fórmula, suponiendo la siguiente tabla como el resultado de la estimación de un proyecto:

Tabla 4. Ejemplo de product backlog estimado sin detalle de las HU.

ID	Estimación
HU-01	5
HU-02	5
HU-03	8
HU-04	13
HU-05	8
HU-06	5
HU-07	5
HU-08	5
HU-09	5
HU-10	8
HU-11	13
HU-12	20
HU-13	5
HU-14	5
HU-15	13
HU-16	20
HU-17	8
HU-18	13
HU-19	13
HU-20	13
HU-21	13
HU-22	13
HU-23	13
HU-24	5
HU-25	5

Peso total del proyecto: 239 puntos.

Peso HU Mínima: 5

Peso HU Máxima: 20

Pivote: 8

Duración de los sprints: 2 semanas

Compromiso del equipo: 36 puntos

Número de sprints para construir el

proyecto: $\text{Peso total del proyecto} /$

$\text{Compromiso del equipo} = 239 / 36 = 6.63$

Recomendaciones sobre la estimación

Dos recomendaciones muy útiles para los equipos de desarrollo que apenas empiezan a abordar la técnica son los siguientes:

1. A la hora de estimar una historia de usuario, pensar que todo lo que se debe realizar es vital. Como mínimo toda historia de usuario está dividida en backend y frontend, es decir, en la lógica que componen la funcionalidad y la pantalla. Para poder construir esa funcionalidad que se está estimando, es necesario realizar una serie de actividades que involucran tener en cuenta: desde cuántos elementos hay en pantalla que capturan o exponen

datos, validaciones en pantalla, mensajes de error o de éxito; el paso de esta información hacia la lógica y el retorno de la respuesta desde la lógica hacia la pantalla, la estructuración de clases de la lógica regida por patrones como MVC, MVP, MWM, BloC, etc.; las consultas, si estas pueden ser heredadas de un método genérico o deben ser implementadas de manera nativa; el número de relaciones de las tablas para las consultas, etc. Adicionalmente, el versionamiento, las pruebas unitarias, las pruebas de integración, las revisiones y refinamiento producto del análisis estático de código, las revisiones funcionales que aseguran que la HU está cumplida. Esto hace que la percepción real a la hora de estimar no sea como se acostumbra, pensando sólo en líneas de código, sino pensando en un panorama mucho más completo.

2. Cuando se realiza el cálculo del número de sprints del proyecto y da un valor decimal, como en el ejemplo anterior que da 6.63, lo más recomendable es indicar que el número de sprints es el siguiente valor entero. En este caso 7 sprints tomaría hacer el proyecto. Esto se hace con el fin de no asumir compromisos parciales y utilizar el tiempo restante del compromiso para incrementar las prácticas de calidad.

Técnica de priorización de requisitos utilizada

En los marcos de trabajo ágiles como scrum y XP se abordan diversos modelos de priorización. El objetivo principal de esta labor es dar un orden de atención a las historias de usuario que se realizarán en la distribución por sprints. Uno de los modelos más utilizados en la industria por su simplicidad y que es el adoptado por Ecosistema Smart Campus UNIAJC es MoSCoW.

La palabra MoSCoW utiliza las siglas MSCW para referenciar frases que determinan el orden de importancia a través de la palabra deber en inglés.

M = Must Have = Debe tenerse

S = Should Have = Debería tenerse

C = Could Have = Podría tenerse

W= Won't have by now = No se tendrá por ahora

Estas siglas recuerdan que todas las historias de usuario marcadas con la letra M tienen el más alto grado de prioridad porque, de acuerdo a la visión del equipo junto con el product owner, son parte del core o núcleo del proyecto. Son las que más valor de negocio le aportan al mismo. Por tanto, si o si deben estar lo antes posible. Es decir, se verán reflejadas en los primeros sprints del proyecto. Las historias marcadas con S deberían estar una vez atendidas las de mayor prioridad. No son parte del núcleo, pero son un complemento excelente del mismo. Pueden esperar a sprints posteriores del proyecto. Las historias marcadas como C son un ideal, un plus del proyecto. Son un complemento que incrementa la satisfacción del cliente, pero no son vitales para el proyecto. Normalmente estas HU son abordadas al final del proyecto. Toda HU marcada como W indica que en la fase actual no son contempladas dentro del alcance, pero pueden marcar el camino para una nueva.

Pasos para la priorización por MoSCoW

- I. Una vez se ha completado la estimación, el equipo debe volver a leer el product backlog. Esto se realiza como filtro de calidad. Se identifica si hay claridad sobre las HU o existe alguna duda y se resuelve.
- II. Junto con el product owner a cada HU se le asigna un valor de acuerdo a la escala mencionada de la técnica (valores posibles M, S, C, W). Continuando con el ejemplo anterior, la tabla 5 muestra cómo se completa el product backlog con los valores de estimación y priorización.

Tabla 5. Ejemplo de product backlog estimado y priorizado sin detalle de las HU.

ID	Estimación	Priorización
HU-01	5	M
HU-02	5	S
HU-03	8	M
HU-04	13	M
HU-05	8	S
HU-06	5	S
HU-07	5	S
HU-08	5	M
HU-09	5	S
HU-10	8	M
HU-11	13	S

Viene de la página 51

HU-12	20	S
HU-13	5	M
HU-14	5	S
HU-15	13	M
HU-16	20	M
HU-17	8	S
HU-18	13	S
HU-19	13	M
HU-20	13	M
HU-21	13	S
HU-22	13	M
HU-23	13	M
HU-24	5	M
HU-25	5	S

Recomendaciones sobre la priorización

Dos recomendaciones muy útiles para los equipos de desarrollo que apenas empiezan a abordar la técnica son las siguientes:

I. No olvidar preguntarle al product owner (PO) si está de acuerdo con la priorización. En algunas ocasiones el PO no podrá acompañar el equipo en la reunión de priorización, por lo que siempre es importante validar posterior a la aplicación de la técnica si está de acuerdo con el orden de atención que se ha dado.

II. No suponer que todas las HU son M. Es frecuente creer que todas las HU son importantes e inmediatas, pero la realidad es que siempre hay cosas que esperan y cosas que efectivamente son inmediatas. Tener esto presente.

Artefacto de la etapa de análisis: *Sprint Backlog*

La construcción de un sprint backlog supone la división del product backlog en listados más pequeños a realizar por cada sprint. La forma de realizar esta labor es bastante simple. Dado que a estas alturas ya se debe tener el tamaño del sprint, el número de sprints a realizar, el compromiso del equipo y el orden de prioridad de atención de la HU, la división se da casi en un orden natural. Siempre se atienden primero las historias de usuario M, salvo que el número de puntos de complejidad exceda demasiado el compromiso del sprint, en cuyo caso se pueden incluir

historias de complejidad S. La figura 5 muestra un ejemplo resultante de generar el sprint backlog del ejemplo que se ha venido trabajando en esta sección.

Sprint 1			Sprint 2			Sprint 3			Sprint 4		
HU	Complejidad	Prioridad	HU	Complejidad	Prioridad	HU	Complejidad	Prioridad	HU	Complejidad	Prioridad
1	5	M	15	13	M	16	20	M	19	13	M
8	5	M	4	13	M	20	13	M	6	5	S
13	5	M	2	5	S	9	5	S	23	13	M
24	5	M	14	5	S	Total pun	38		7	5	S
3	8	M	Total pun	36					Total pun	36	
10	8	M									
Total puntos	36										

Sprint 5			Sprint 6			Sprint 7		
HU	Complejidad	Prioridad	HU	Complejidad	Prioridad	HU	Complejidad	Prioridad
22	13	M	12	20	S	18	13	S
11	13	S	21	13	S	17	8	S
14	5	S	Total pun	33		5	8	S
2	5	S				Total pun	29	
Total puntos	36							

Figura 6. Ejemplo Sprint Backlog

Se recomienda revisar el **anexo b** de este documento para tener un modelo a seguir para los sprint backlog.

Diseño de software

El diseño, bajo la perspectiva de Pressman & Maxim (2015), constituye una de las etapas más importantes para la correcta ejecución de un proyecto de software. Si bien, durante la etapa de análisis de requisitos se responde el interrogante; ¿qué se hará?, en la etapa de diseño se responde: ¿cómo se hará? Y para dar respuesta a este interrogante se requiere una metodología de diseño adecuada.

El proyecto Ecosistema Smart Campus, se nutre de la experticia del equipo de investigadores, quienes dada su experiencia profesional han afrontado distintos proyectos con múltiples empresas del país y del exterior. Con base en esto, se ha construido una plantilla del Documento de Arquitectura de Software o DAS, como se le conoce por sus siglas. Este es el artefacto fundamental de esta etapa y contiene todas las decisiones arquitecturales que permiten una correcta codificación del proyecto, basado en la formalización de las decisiones a través de diagramas UML.

Artefacto de la etapa de diseño: Documento de Arquitectura de software-DAS

El artefacto está compuesto de tres elementos principales que permiten organizar la información de manera correcta. A continuación, se amplía en detalle cada una.

Información de contexto

Contiene toda la información introductoria y de contexto del documento y del proyecto. Lo componen una introducción del documento, el propósito, el alcance del proyecto, las definiciones, siglas y abreviaturas, referencias y una perspectiva de vista global donde se detallan que vistas utilizará el documento para formalizar las decisiones.

Decisiones arquitecturales basadas en atributos de calidad y patrones de diseño

Está compuesto por la sección metas y restricciones, donde el equipo de desarrollo especifica los atributos de calidad (estado deseado de comportamiento del software) y los patrones de diseño que lo satisfacen (son soluciones eficientes a problemas conocidos que ya están comprobadas que funcionan de manera correcta). Es importante conocer que los atributos de calidad pueden ser tanto observables como no observables. Esto implica que si son observables pueden ser especificados en partes del código y su representación en los diagramas UML es posible. De ser no observables, se verá en comportamientos del sistema como la adaptación de la interface gráfica a diferentes tamaños de pantalla. En esta sección se realiza una tabla por los atributos observables y una por los no observables, donde se detalla atributo de calidad a cubrir, descripción del atributo, patrón de diseño que lo resuelve y dónde se ve reflejada la implementación del mismo.

La correcta selección de atributos de calidad no es tarea fácil. Aunque existen varios modelos que pueden sugerir posibles atributos recomendados para los proyectos como son la norma ISO/IEC 25000 y el modelo FURPS. Siempre es recomendado evaluar las características de cada proyecto para seleccionar cada atributo de calidad de manera adecuada y no equivocarnos por omisión. Si bien la selección de atributos de calidad es el primer paso, una vez seleccionados debemos entender que cubrir por completo el atributo por medio de patrones de diseño y luego evaluar su resultado con diversos escenarios de calidad puede resultar muy costoso. Por este motivo, siempre es aconsejable que se especifique cuales sub categorías dentro del atributo de calidad se van a cubrir dentro del proyecto.

Un ejemplo que puede ayudar al lector a asimilar más este aspecto es el atributo de calidad: *usabilidad*. Este atributo tiene una gran variedad de elementos que lo componen. Por mencionar algunos de ellos está la facilidad

de aprendizaje, la adaptabilidad, la usabilidad universal, la confiabilidad, el ser comprensible, el ser intuitivo, la memorabilidad, entre otros. Cubrir todos estos aspectos es muy difícil, ya que el equipo debería entonces hacer múltiples tipos de pruebas que garanticen la implementación y satisfacción dentro de la solución. Las tablas 6 y 7 muestran el ejemplo de la tabla utilizada dentro de la plantilla.

Tabla 6. Modelo de tabla de atributos de calidad observables

Atributos de calidad observables			
Atributo de calidad	Descripción	Táctica o Patrón de diseño	Aplicación
Atributo de calidad: ·Subcategoría 1 ·Subcategoría 2 ·Subcategoría N	Descripción de las subcategorías	Patrón o patrones de diseño a utilizar para satisfacer cada subcategoría	Se especifica. En lugar del código fuente se ve aplicado.

Tabla 7. Modelo de tabla de atributos de calidad no observables

Atributos de calidad no observables			
Atributo de calidad	Descripción	Táctica o Patrón de diseño	Aplicación
Atributo de calidad: ·Subcategoría 1 ·Subcategoría 2 ·Subcategoría N	Descripción de las subcategorías	Patrón o patrones de diseño a utilizar para satisfacer cada subcategoría.	Se especifica en que momento de la ejecución se puede ver el comportamiento.

Vistas arquitecturales

De acuerdo al trabajo realizado por Kruchten (2006), existen muchos diagramas UML y, para darle un orden, se identificaron características comunes entre ellos y luego fueron agrupados de acuerdo a esas características en vistas. Las vistas están dirigidas para la comprensión específica de un público objetivo y esto facilita saber específicamente qué diagramar.

El primer paso para poder hacer correctamente diagramas UML es entender qué debemos diagramar. Aquí es altamente importante que el equipo haya elaborado las tablas indicadas anteriormente. Los atributos de calidad están especificando las características no funcionales del sistema y los patrones de diseño el cómo resolverlas. La representación de estos patrones a través de UML será lo que conforme el diseño arquitectural del sistema y modele la solución.

Vista Física

De acuerdo con Kruchten (2006), la vista física está encargada de expresar la comunicación entre dispositivos físicos y sus componentes principales. Existen varias maneras de representar con diagramas informales o diagramas UML.

En los diagramas formales a través de la notación UML, se puede representar con nodos los dispositivos físicos que intervienen en la solución, como son los dispositivos móviles o los PC de los usuarios finales, los servidores internos de la UNIAJC y los servidores externos, necesarios por consumos de APIs externas, ver figura 7.

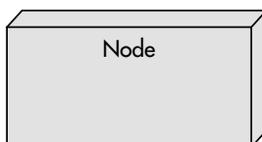


Figura 7. Representación UML de un nodo. Herramienta Visual Paradigm Versión 16

Estos dispositivos cuando permiten que se les hagan peticiones exponen puertos para la comunicación, lo que crea interfaces de comunicación habilitadas entre los nodos. Esto también tiene su representación en UML que puede ser apreciada en la figura 8.

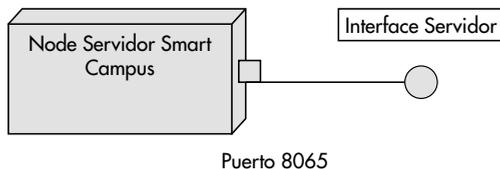


Figura 8. Representación UML de un nodo con un puerto y una interface de comunicación. Herramienta Visual Paradigm Version 16

Finalmente, existen elementos de software representados como componentes que permiten detallar aspectos como servidores de aplicaciones, donde corren las aplicaciones web o las APIs propias del proyecto, bases de datos, sistemas operativos o servicios externos, ver figura 9.

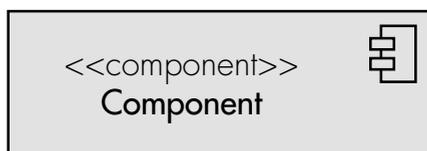


Figura 9. Representación UML de un componente. Herramienta Visual Paradigm Versión 16

La unión de estos elementos permite realizar un diagrama de nodos y componentes orientado a la vista de física. En la figura 10 se puede apreciar un ejemplo más completo sobre la aplicación de los elementos UML mencionados y trabajando en conjunto para reflejar un modelo de solución bajo las condiciones de infraestructura actuales de Smart Campus.

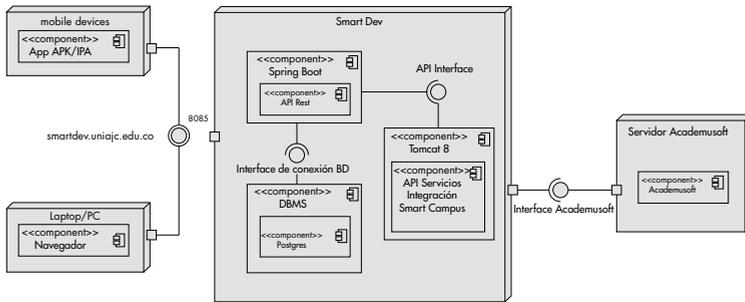


Figura 10. Diagrama de nodos y componentes modelo para proyectos Smart Campus UNIAJC

Vista Lógica

Bajo el enfoque dado por Kruchten (2006), la vista lógica revela al equipo de desarrollo las decisiones arquitecturales principales que dan estructura al aplicativo. Esto quiere decir que la mayor cantidad de patrones de diseño son expuestos aquí, en especial los que se clasifican como estructurales y de comportamiento, aunque algunos creacionales también pueden ser reflejados. Los diagramas principales que pueden ser utilizados por los equipos en esta vista son: el diagrama de clases, el diagrama de comunicación y el diagrama de secuencia. Pero existen otros diagramas como el de paquetes y componentes tanto simple como de capa nivel, que permiten dar un detalle exhaustivo de la solución. Es importante no confundir los diagramas de secuencia y de comunicación que se exponen en esta vista con los de procesos. En esta vista se detallan los elementos que componen la arquitectura y su comunicación entre sí, en cambio en la vista de proceso se detalla a nivel funcional la interacción entre las diversas funcionalidades a construir.

Con respecto a los diagramas de clase es vital comprender que la estructuración de este diagrama no debe exponer un modelo entidad relación con métodos. Por el contrario, debe exponer de manera global toda la estructuración del arquetipo a implementar. A continuación, en la figura 11 se puede apreciar un ejemplo de una arquitectura multicapa para una aplicación web a construir en Java.

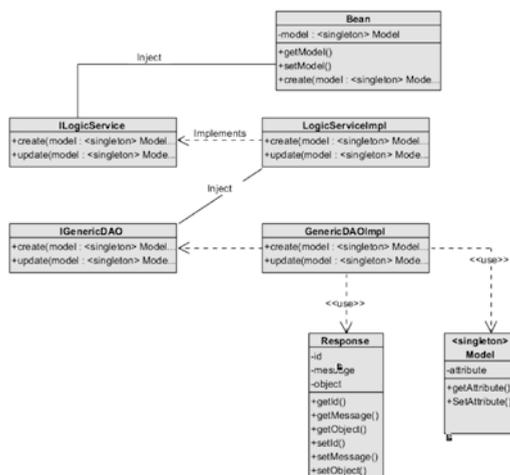


Figura 11. Ejemplo diagrama de clases. Herramienta Visual Paradigm Versión 16

Es importante destacar de la lectura correcta de un diagrama de clase para esta vista, que se debe poder dar una explicación detallada sobre los elementos que lo componen, los atributos de calidad que satisface y los patrones de diseño que implementa. Del diagrama anterior se puede apreciar que la aplicación recibe todas las peticiones a través de una pantalla y esta tiene asociado un bean. Los bean son clases específicas de Java que permiten hacer un binding de los elementos en pantalla con atributos de objetos o variables que se construyen para capturar la información. Los bean de este modelo representan la capa de presentación del aplicativo y se comunican con la siguiente capa, que es la de lógica de negocios a través de la inyección de dependencias a la interfase de esa capa. La inyección de dependencias es un patrón de diseño que permite, de manera muy simple, trabajar bajo la definición de los métodos a utilizar sin tener que reservar espacio en memoria para su uso. Únicamente cuando una petición es requerida, se crea el objeto bajo demanda, se utiliza y al terminar su función se elimina de la memoria. Esto incrementa el uso eficiente de recursos del sistema, haciéndolo dinámico. Estas interfaces que son inyectadas solamente en tiempo de ejecución y bajo la condición de que se haya hecho una petición, invocan a su clase de implementación. De la misma manera, las clases que implementan la lógica usan el mismo concepto para llamar por inyección de dependencias a clases de acceso a datos. Estas clases, implementando el patrón de diseño DAO, son las únicas encargadas de administrar las peticiones a la bd. Esto independiza la capa de lógica de la capa de presentación. Haciendo un diseño arquitectural multicapa. Es relevante para la interpretación apreciar que sólo el DAO utiliza al modelo, por lo que la información que viajó encapsulada por toda la aplicación estará dentro de un objeto de transferencia de datos o

DTO. Así mismo, para no tener que crear una clase DAO por cada modelo, se ha utilizado colecciones de Java para generar una clase DAO genérica, la cual solo requiere de la entidad para ejecutar los métodos globales: si se requiere un método específico, si se requiere la implementación de una clase propia para ese DAO. De igual forma, garantizando que las transacciones son únicas y serializadas, evitando que se sobre escriban, todos los modelos son singleton. Gracias a este modelo, se garantiza que el aplicativo es escalable, mantenible, modificable, confiable y de un rendimiento eficiente.

Otro ejemplo de diagrama a utilizar para esta vista es el de paquetes y componentes, donde el principal objetivo de los paquetes es servir de agrupador de los componentes que integran la arquitectura. En UML los paquetes se representan mediante carpetas. Ver figura 12.

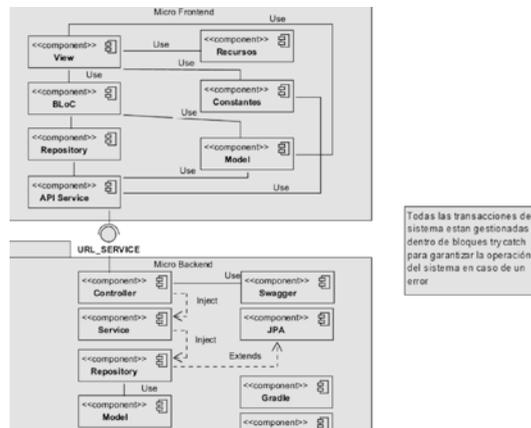


Figura 12. Diagrama de paquetes y componentes. Herramienta Visual Paradigm Versión 16

Es vital para el modelo de trabajo de Smart Campus que cada diagrama vaya acompañado de su respectiva explicación detallada, como se hizo en el caso del diagrama de clases. Se deja al lector la interpretación de la imagen 12.

Vista de Procesos

Esta vista tiene una estrecha relación con el resultado de la etapa de análisis. Aquí se describe el paso a paso funcional del aplicativo mediante la identificación correcta de los roles que intervienen, qué acciones pueden ejecutar y qué reglas de negocio aplican. Los diagramas principales que se utilizan en esta vista, de acuerdo a Kruchten (2006), son los de secuencia, los de colaboración y los de comunicación.

Dentro del Ecosistema Smart Campus, los investigadores siempre están buscando nuevas formas de aproximarse a resultados de alta calidad mediante la innovación y la creatividad aplicada. Así mediante Pastrana, Ordóñez, Ordóñez, Thom, & Merchan (2018), uno de los investigadores propone un enfoque diferente utilizando Business Process Modeling (BPM o Modelado de Proceso de Negocio). Esta técnica utiliza una notación específica, llamada Notación BPM o BPMN por sus siglas en inglés. Esta forma de modelado de procesos contiene una semántica propia que permite de manera detallada definir las funcionalidades del sistema, los roles que las pueden ejecutar, las reglas de negocio, las validaciones lógicas y el tipo de tarea a ejecutar. Lo anterior ayuda a unificar la visión del proyecto debido a que BPMN solo tiene una única interpretación al ser un lenguaje gráfico y, por tanto, reduce en gran medida la ambigüedad producto de escribir las historias de usuario en lenguaje natural.

Business Process Modeling - BPM

Responde a una forma de modelado que facilita la descomposición de un proyecto en los procesos que lo componen, a fin de poder entender plenamente su complejidad, los elementos que lo componen, los roles que interactúan dentro de las funcionalidades y su comunicación entre sí. Su forma de representación es como un workflow o flujo de trabajo. Posee como restricciones que un proceso sólo puede tener un estado de inicio y uno de fin; todas las tareas llevan semántica asociada, todas las compuertas llevan semántica asociada y compuerta que abre debe ser cerrada previa al fin del modelado. A continuación, se detallan los elementos que componen la notación utilizando BPMN y la herramienta Bizagi Modeler.

Proceso

Se representa como un contenedor de todos los elementos que internamente detallarán su lógica. Es el elemento principal para iniciar el modelado. La figura 13 muestra su representación en la herramienta Bizagi.

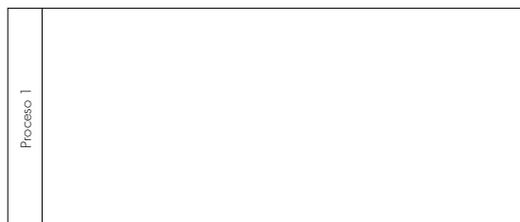


Figura 13. Representación de un proceso en BPMN. Herramienta Bizagi

En el lado izquierdo se aprecia que es donde se indica el nombre del proceso. Es importante mencionar que un proyecto puede tener tantos procesos como sea requerido en el nivel de detalle.

Lane o Carril

Representa el rol que ejecutará la acción. Un contenedor de procesos puede tener tantos roles como sea necesario. Su representación modifica al contenedor de proceso agregando una línea horizontal que separa el carril para indicar que esas acciones sólo son ejecutadas por ese rol. La figura 14 detalla la representación mencionada con un rol y la figura 15 con dos roles.



Figura 14. Representación Lane BPMN con un rol. Herramienta Bizagi



Figura 15. Representación Lane BPMN con dos roles. Herramienta Bizagi

Estado de inicio

Indica el inicio de un proceso. Se representa con una bola de color verde que es situada en el lane o carril que comienza la acción, ver figura 16. Sólo puede existir un estado de inicio por proceso.

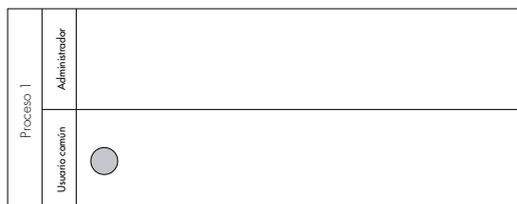


Figura 16. Representación Estado de inicio BPMN con dos roles. Herramienta Bizagi

Estado de fin

Indica el final de un proceso. Se representa con una bola de color rojo que es situada en el lane o carril que finaliza la acción, ver figura 17. Sólo puede existir un estado de fin en un proceso. Asimismo sólo puede haber estado de fin si hay estado de inicio.

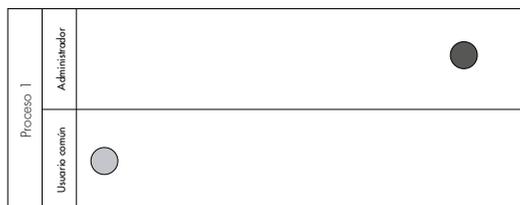


Figura 17. Representación Estado de fin BPMN con dos roles. Herramienta Bizagi

Tarea

Es un elemento fundamental dentro de la representación de las funcionalidades del proceso. Detalla qué acción se va a ejecutar por el rol en donde está puesta. Siempre inicia con un verbo en infinitivo, es decir, terminado en ar, er, ir. Esto se hace para indicar acción. Las tareas tienen una particularidad y es que poseen semántica que indica el tipo de acción a ejecutar, resaltando si es realizada por el usuario (ver figura 18); por el sistema (ver figura 19); si es manual (ver figura 20); si es una tarea de envío de mensaje (ver figura 21) o de recepción del mensaje (ver figura 22); si es un script (ver figura 23) o si es una regla de negocio (ver figura 24). Lo anterior se hace para dar claridad en los elementos utilizados, en la comunicación entre ellos y en la comunicación entre los roles.

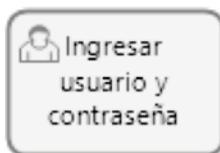


Figura 18. Tarea realizada por el usuario, notación BPMN. Herramienta Bizagi

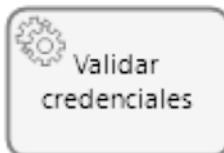


Figura 19. Tarea realizada por el sistema, notación BPMN. Herramienta Bizagi

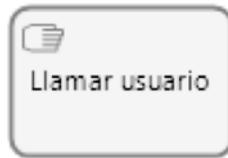


Figura 20. Tarea realizada de forma manual, notación BPMN. Herramienta Bizagi



Figura 21. Tarea de envío de mensaje, notación BPMN. Herramienta Bizagi

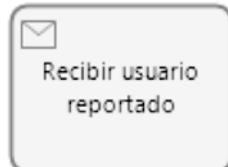


Figura 22. Tarea de recepción de mensaje en notación BPMN. Herramienta Bizagi

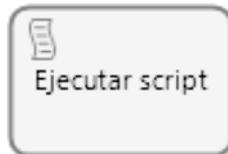


Figura 23. Tarea de ejecución script, notación BPMN. Herramienta Bizagi

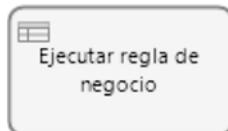


Figura 24. Tarea de ejecución regla de negocio, notación BPMN. Herramienta Bizagi

Compuerta

Las compuertas detallan una regla de negocio o validación a realizarse. Indican una divergencia o convergencia de varios caminos del proceso. A continuación, se da un breve detalle de cada una:

- **Exclusiva:** funciona como una disyunción, es decir, genera una validación donde se toma una alternativa u otra, ver figura 25.
- **Paralela:** funciona como un conector donde dos alternativas o caminos del proceso se ejecutan al tiempo, ver figura 26.

- **Inclusiva:** funciona como una disyunción, es decir, genera que dos condiciones deban ser cumplidas antes de continuar con el flujo normal del proceso, ver figura 27.
- **Compuerta basada en eventos:** en un proceso donde aparece esta compuerta, se indica que hasta no disparar un evento que la invoque, los caminos que desprenden de esta no estarán accesibles para el proceso. Ver figura 28.
- **Compuerta exclusiva basada en eventos:** funciona como un conector donde hay varias opciones con posibilidad de ser escogidos. Solamente cuando se da la selección de uno en específico, los demás caminos se inactivan y no son ejecutados. Un ejemplo simple para su uso son las opciones de menú. Aunque un menú puede tener múltiples opciones, el usuario sólo puede acceder a una al tiempo. Ver figura 29.
- **Compuerta paralela basada en eventos:** funciona como un conector donde al ser disparado por un evento varios caminos son ejecutados al tiempo. Si el evento no es iniciado, estos caminos no estarán disponibles. Ver figura 30.
- **Compuerta compleja:** se utiliza de manera convergente o divergente. Su uso radica en la complejidad de un proceso, donde no se ha podido controlar, por ejemplo, la dependencia de continuidad del proceso en la validación de 2 de 3 respuestas. Ver figura 31.



Figura 25. Compuerta exclusiva, BPMN. Herramienta Bizagi



Figura 26. Compuerta paralela, BPMN. Herramienta Bizagi



Figura 27. Compuerta inclusiva, BPMN. Herramienta Bizagi

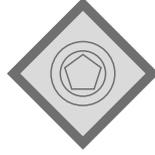


Figura 28. Compuerta basada en eventos, BPMN. Herramienta Bizagi



Figura 29. Compuerta exclusiva basada en eventos, BPMN. Herramienta Bizagi



Figura 30. Compuerta paralela basada en eventos, BPMN. Herramienta Bizagi



Figura 31. Compuerta compleja, BPMN. Herramienta Bizagi

Subproceso

Se representa como un contenedor de proceso. Ayuda a disminuir la complejidad implícita en una visión global y sirve como punto de partida para descomponer desde una macrovisión a un detalle exhaustivo. La figura 32 muestra su representación en la herramienta Bizagi.



Sub proceso 1

Figura 32. Subproceso, BPMN. Herramienta Bizagi

Evento intermedio

Se representa un evento que está en medio de un proceso, por ejemplo, una comunicación asíncrona de mensajes o una espera de tiempo requerida hasta disparar el servicio. Los dos eventos intermedios más utilizados son el de mensaje, ver figura 33 y el de espera, ver figura 34.



Figura 33. Evento intermedio de mensaje, BPMN. Herramienta Bizagi



Figura 34. Evento intermedio de espera, BPMN. Herramienta Bizagi

El ejemplo de la figura 35 ejemplifica el uso del evento intermedio de mensajes.

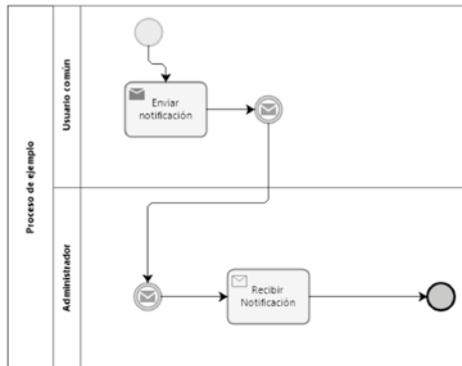


Figura 35. Ejemplo uso evento intermedio, BPMN. Herramienta Bizagi

La integración de estos componentes permite modelar a un detalle completo todos los procesos que integran un proyecto. Esta técnica permite fácilmente validar con todos los stakeholders la comprensión del proyecto y la visión del mismo, además de permitir refinarlo en caso de ser necesario.

Es muy recomendado el uso de esta vista cuando el proyecto a afrontar tiene una complejidad significativa y un número de roles que interviene mayor a 3.

Por ejemplo, en el trabajo de grado de Delgado & Gutiérrez (2020) se construyó una solución para la gestión de las solicitudes de la Facultad de Ingenierías. Una de las principales causas para esta solicitud es que el proyecto tenía mucha tarea manual con intervención de varias personas y pocas alertas tempranas de gestión y control.

Para hacer un correcto análisis de la solución y complementarlo con el diseño de la solución, el equipo de trabajo recurrió al uso del modelamiento de procesos de negocio que le permitiera comprender dónde estaban los mayores problemas y modelar una solución que optimizara de manera significativa el proceso. La figura 36 muestra un ejemplo de la vista global utilizando una gran parte de los elementos mencionados a este punto.

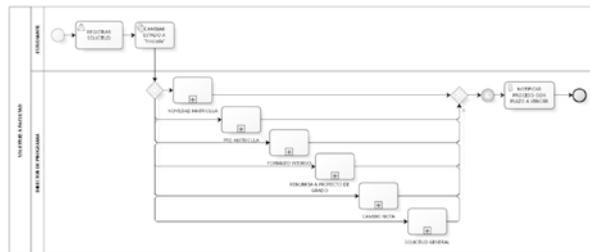


Figura 36. Macro proceso solución sistema de solicitudes Facultad de Ingenierías. Modelado con Bizagi. Fuente: proyecto de grado Delgado & Gutiérrez (2020)

Vista de Despliegue

De acuerdo con Kruchten (2006), la vista de despliegue es muy similar a la vista física. Ambas se encargan de expresar la comunicación entre dispositivos físicos y sus componentes principales. Pero su principal diferencia radica en el detalle interno que tienen los componentes principales de la aplicación. En este diagrama es mucho más detallada la estructuración y se puede percibir una mayor cantidad de patrones de diseño implementados. La figura 37 da un claro ejemplo de esta vista.

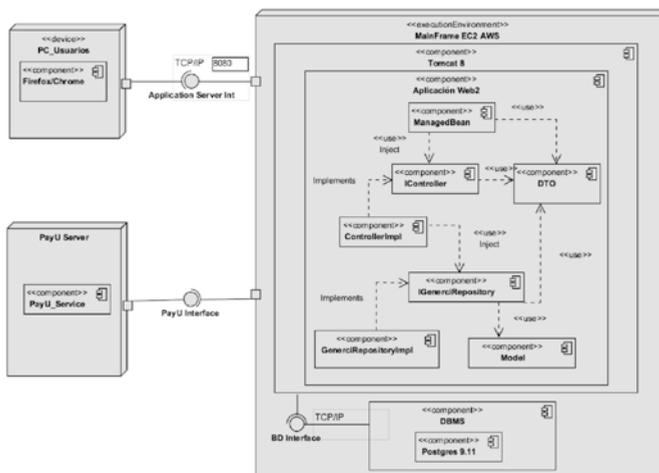


Figura 37. Ejemplo diagrama de despliegue vista de despliegue. Herramienta visual paradigm versión 16

Se recomienda al lector revisar el anexo c adjunto a este documento, para tener una guía más completa de la plantilla utilizada dentro de los proyectos.

Desarrollo de software y calidad

Esta etapa permite materializar el *¿qué se hará?*, descrito en el análisis de requisitos y el *¿cómo se hará?*, especificado en el diseño de la solución mediante el código fuente. Aquí, el scrum development team, de manera organizada y de acuerdo con la planeación, trabaja de manera iterativa incremental para construir unidades de código que permiten hacer un entregable 100% funcional. Estas unidades desplegables conforman una parte del proyecto que, de sprint a sprint, suman al total de compromisos, resolviendo las HU de cada fase. La base fundamental para que el equipo pueda concentrarse únicamente en programar son las decisiones tomadas en el diseño. Gracias a esas decisiones, el equipo de desarrollo sabe exactamente a qué acudir dentro del lenguaje de programación, utilizando el máximo potencial de los frameworks disponibles.

Si las fases de análisis y diseño han sido ejecutadas adecuadamente, se minimiza el reproceso producto de dudas o de decisiones no tomadas a la hora de programar, incrementando la productividad del equipo de desarrollo. Claro está, esto no es suficiente para garantizar que la calidad de cada entregable sea la más alta posible, como indica Pastrana et al. (2019), y por ello, se hace necesario generar una serie de mecanismos y prácticas que resalten la mejora continua producto de la calidad preventiva aplicada sprint a sprint.

Tradicionalmente los proyectos de desarrollo de software separan la etapa de construcción del proyecto de la de calidad, como indica el enfoque inicial de Pressman & Maxim (2015), en el que primero se construye el producto y luego se revisa que tanta calidad tiene para someterlo a un proceso de correcciones y revisiones hasta que cumpla con el nivel aceptado por el revisor. Este enfoque puede resultar costoso dependiendo de la cantidad de reproceso a asumir y la cantidad de tiempo disponible en el cronograma de actividades pactadas con el product owner. Siguiendo el otro enfoque, el agilista, la calidad y el desarrollo deben ir de la mano en todo momento, como lo indica Beck (1999).

Desde la visión de los 12 principios y los 4 valores de Fowler & Highsmith (2001), lo deseable para todo proceso de desarrollo de software es tener un perfecto equilibrio entre los compromisos y la vida personal, esto sólo es posible alcanzarlo si existe un ritmo sostenible de desarrollo de software, donde los mecanismos de alerta temprana permitan al equipo identificar sus buenas y malas prácticas de desarrollo, logrando así, sprint a sprint, mejorar su forma de trabajo y obtener un resultado de calidad más alto y con menor reproceso. Enfoque que coincide con el de Pastrana et al. (2019), quienes centrados en este ideal elaboran una identificación de las principales buenas prácticas de desarrollo detalladas por Beck (1999), pero dándole forma a través de herramientas específicas para poder estandarizar el proceso de desarrollo de software, propio de Smart Campus.

Buenas prácticas de desarrollo de XP adoptadas

Acorde al trabajo realizado por Pastrana et al. (2019), Extreme Programming o XP, como comúnmente se le conoce, describe 12 buenas prácticas con el ánimo de que los equipos de desarrollo puedan obtener información rápida de cómo están llevando la construcción de su aplicativo y puedan corregir a tiempo posibles brechas que con el paso del tiempo se incrementen y terminen siendo un problema para el producto ya puesto en producción. Según Beck (1999), las buenas prácticas recomendadas son:

1. **El juego de la planificación:** para todo equipo de desarrollo es fundamental realizar una correcta planeación de su trabajo, por lo que es fundamental tener claro los compromisos por entrega, las responsabilidades de cada uno de los integrantes del equipo y los objetivos a cumplir de acuerdo a los roles que desempeñen los integrantes.

2. **Entregas pequeñas:** se debe trabajar por periodos de tiempo controlados no mayores a 4 semanas. En concordancia con la definición de scrum a esto se le denomina sprint y tiene como objetivo poder controlar de manera eficiente la evolución del producto.
3. **Metáfora:** una metáfora es una explicación de algo por medio de una representación diferente con la que guarda una relación. Las abreviaciones o nomenclaturas pactadas en un proyecto entre todos los integrantes del equipo son un ejemplo de esto. Para aclararlo un poco más, por ejemplo, el equipo de desarrollo puede indicar que tbl significa tabla y crear esta representación como algo común con el equipo de soporte del proyecto. Esta práctica puede llegar a ser muy difícil de implementar.
4. **Diseño simple:** arquitecturalmente un proyecto debe mantener el diseño de su solución lo más simple posible para poder controlar los cambios estructurales, de comportamiento o creacionales de una manera más eficiente y mantener en orden el crecimiento evolutivo del proyecto.
5. **Refactoring:** todo proceso de desarrollo de software tiene implícito algún grado de reproceso, producto de malas prácticas o de errores humanos. Por lo anterior, es necesario tener presente siempre un tiempo adecuado para realizar estas correcciones o mejoras posibles que permitan disminuir la deuda técnica.
6. **Pair Programming o programación en parejas:** es una práctica poco utilizada debido a la complejidad de la misma. Se utiliza principalmente cuando queremos nivelar la curva técnica del equipo al tiempo que supervisamos la calidad del entregable. En esta práctica dos personas del equipo trabajan juntos la misma historia de usuario. Primero, uno de los integrantes toma el control del desarrollo y empieza la codificación, a esta persona se le denomina navegante, ya que es quien está directamente afectando la unidad de código. Su acompañante es quien aprende del trabajo que este está realizando y le ayuda a comprobar la concordancia de lo que se escribe a nivel de código frente al documento de arquitectura de software y a los criterios de aceptación de la historia de usuario. Esto permite que, al tiempo, la unidad de código sea puesta a prueba constantemente.
7. **Propiedad colectiva:** tanto la información del proyecto, como el código fuente debe estar disponible en todo momento para cualquiera de los

integrantes del equipo. Por lo anterior, deben existir mecanismos que permitan garantizar esta práctica en todo momento.

8. **Integración y despliegue continuo:** esta práctica apoya al equipo como una alerta temprana que permite identificar si el incremento que se está realizando afecta de manera negativa a la unidad desplegable generando impedimentos para alcanzar los objetivos del sprint. Aquí las herramientas que se implemente propenden por identificar de manera inmediata si se ha producido un error sintáctico o técnico que impida avanzar y notificarlo de manera inmediata para que todo el equipo esté enterado y tome acciones correctivas de manera inmediata.
9. **Semana de 40 horas:** un equipo cansado es un equipo que empieza a generar errores de manera inconsciente. Esto crea no sólo problemas de rendimiento sino también de calidad. Por lo anterior, no se recomienda que un equipo esté dedicado a un proyecto más de 40 horas por semana.
10. **Cliente in situ:** esta práctica invita a que, si es posible, se trabaje lo más cerca posible del cliente, preferiblemente en el mismo lugar, así se garantiza la comunicación constante y cara a cara. En caso de no ser posible, al menos se recomienda tener contacto constante con el product owner del proyecto.
11. **Estándar de programación:** el secreto de mantener la misma visión de desarrollo es que todo el equipo hable el mismo idioma. Por lo anterior, no se debe trabajar un proyecto de desarrollo sin haber construido ese idioma común que represente el nombrado de clases, variables, constantes, objetos, tablas, etc.
12. **Pruebas:** representan la comprobación de que tanto funcional como técnicamente el software está listo para responder a las necesidades del proyecto. Por lo anterior, el equipo de desarrollo debe generar todos los mecanismos necesarios para garantizar el sello de calidad de manera preventiva y no correctiva. Mecanismos como las pruebas unitarias, pruebas de integración, revisiones por pares, son obligatorias antes de determinar que una historia de usuario ha sido finalizada.

Si bien, de acuerdo a la explicación de este mismo capítulo, algunas de estas prácticas recomendadas por Beck (1999), como son *el juego de la*

planificación, entregas pequeñas, diseño simple y pruebas ya son tomadas en cuenta dentro del proceso de scrum, descrito por Schwaber & Sutherland (2017), existen otras que resultan en un complemento ideal que ayuda a reforzar el proceso de aseguramiento de calidad mediante el aprendizaje y la mejora continua, como indica Pastrana et al. (2019). Por ejemplo, *la propiedad colectiva del código* toma alta relevancia para los equipos de desarrollo de software mediante su implementación a través de un versionador de código. Esta herramienta permite administrar en un repositorio central para todo el equipo, todos los cambios generados producto de la evolución del entregable. Aquí el histórico del código fuente es importante no sólo para identificar la trazabilidad histórica, sino también para garantizar que el código fuente está disponible para todo el equipo de trabajo en cualquier momento. Adicionalmente, brinda la oportunidad de recuperar rápidamente un cambio errado que afecte el código de todos los participantes simplemente con devolverse a la versión anterior estable.

Otra de las prácticas que toma mucha importancia, según Pastrana et al. (2019), es *la integración continua*. Las herramientas de integración continua permiten generar junto con el versionador de código fuente una sinergia ideal para revisar si los cambios evolutivos del código están siendo guardados sin problemas sintácticos o si, por el contrario, estos cambios están afectando la unidad desplegable. Una alerta temprana de este tipo ayuda al equipo a controlar la evolución del aplicativo de manera adecuada, impidiendo que pase mucho tiempo entre la construcción de una funcionalidad y su empaquetado. Así se identifica de manera inmediata si los cambios son adecuados o no y el equipo entra a corregir de manera eficiente. En complemento a lo anterior, una forma adecuada de conservar una mantenibilidad ideal del proyecto es tener *estándar de código* definido para el equipo de trabajo. Esto da lugar a una única forma de escribir el código, a la vez que la curva de aprendizaje está asociada únicamente a entender la lógica implícita en la funcionalidad y no a entender la sintaxis con la que fue construida. Los proyectos de Smart Campus para hacer uso de esta práctica implementan un analizador estático de código que garantice el cumplimiento del estándar internacional de codificación, asociado a las buenas prácticas de programación conocidas para los lenguajes orientados a objetos más comunes. Esta herramienta también va de la mano del versionador de software y del integrador continuo.

Lo anterior busca como objetivo, mantener un ritmo sostenible bajo una jornada eficiente de máximo 40 horas a la semana.

Ambiente de calidad preventiva

Con base en lo anterior, se puede identificar claramente que el objetivo de Smart Campus dentro de la etapa de desarrollo y calidad, es hacer lo más eficiente este aspecto mediante la implementación adecuada de un ambiente de calidad preventiva que entregue al equipo de desarrollo alertas tempranas de mejora y que, a su vez, le permita al equipo mejorar continuamente su cultura de desarrollo.

Alineado el trabajo de Pastrana et al. (2019) Springer Nature Singapore Pte Ltd. Compliance with the client requirements is a key factor for the success of projects. However, this compliance is not always easy to monitor and verify. SCRUM framework flexibility makes it possible to include a variety of techniques and good practices from other methods to help software development teams to achieve their goals. Some of these tools can track historical changes in the code (versioning con Pastrana, Ordóñez, & Cobos (2019), un esquema de calidad preventiva adecuado debe estar orientado al modelo DevOps, que permite garantizar que en todo momento la evolución y operación del proyecto está controlado y medido a través de mecanismos que implementen las buenas prácticas mencionadas. Así, el equipo de trabajo diseña un ambiente compuesto por un versionador de código, un integrador continuo y un analizador estático de código, ver figura 38.

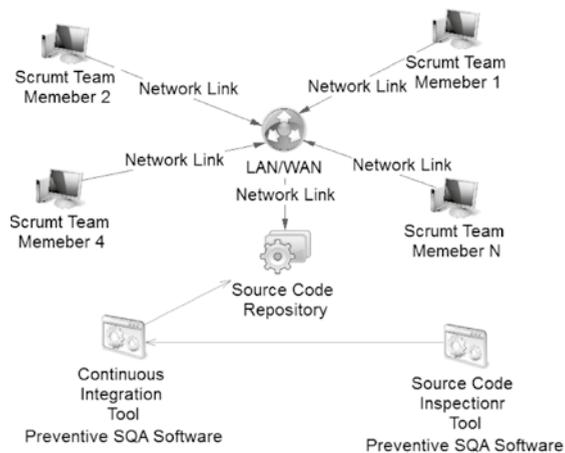


Figura 38. Esquema ambiente de calidad preventiva

Cada vez que uno de los integrantes del equipo de desarrollo suba un cambio al versionador, el integrador continuo detectará este cambio del histórico. Al saber que hay una nueva versión, utilizando el esquema configurado para el proyecto,

primero deberá correr las pruebas unitarias y, si todas salen bien, procederá a generar la unidad desplegable. Esto se hace con el ánimo de garantizar que existe una mínima revisión de calidad previa por parte del desarrollador de software. Si el intento de generar la unidad desplegable falla, la herramienta notifica al equipo que ese cambio subido presenta un error y que debe ser corregido a tiempo. En caso contrario, notifica al analizador estático de código que puede hacer su revisión.

Actualmente, el versionador utilizado por Smart Campus es gitLab, el analizador estático de código es Sonar y el integrador continuo es Jenkins.

Despliegue

El Ecosistema Smart Campus cuenta con el apoyo institucional del departamento de tecnologías DITIC. Gracias al apoyo, el proyecto logra tener un espacio dentro del data center de la institución y hacer la adquisición de un servidor Dell PowerEdge 2900 con las siguientes características:

- Disco duro: 280 GB
- Memoria RAM: 8 GB
- Procesador: Intel Xeon(R) CPU E5420 @ 2.50GHz Max Vel 3600 MHz de cuatro núcleos
- Sistema operativo: Centos 7
- Dominio del servidor: smartdev.uniajc.edu.co

Este servidor es puesto a disposición de todos los proyectos de desarrollo de software que así lo requieran, para realizar actividades de pruebas y simulaciones de despliegue final que permitan garantizar, desde el sprint 0 de cada proyecto, que no habrá inconvenientes a la hora de la puesta en marcha de cada aplicativo. Los siguientes servidores de aplicaciones y base de datos se encuentran dentro de este equipo:

- Apache tomcat 7.0.59
- Apache tomcat 8.0.9
- Apache tomcat 8.5.30
- PostgreSQL 9.5
- PHP 5.6.40

- Java 1.8.0_222
- Nodejs 9.11

Así mismo, el ecosistema cuenta con un ambiente final de producción, servidor denominado smartcampus, que es administrado por DITIC y en donde se despliegan las aplicaciones producto de los resultados de los diversos proyectos. Para el despliegue de las aplicaciones, todas han pasado por un proceso de calidad preventiva y mejora continua, como se describió en la sección anterior, antes de ser aprobadas por el equipo de desarrollo de Smart Campus.

Dentro de este servidor se ha generado la configuración de ambientes de despliegue a partir del uso de contenedores docker. Los contenedores funcionan dentro del sistema operativo aprovechando sus características, pero creando un entorno aislado dentro del él que permite realizar instalaciones y parametrizaciones correspondientes al despliegue, que por fuera de ese contexto no interfieren con otras configuraciones que estén dentro del sistema operativo. De esta manera, fácilmente podemos poner a cohabitar distintas versiones de un lenguaje de programación sin mayor problema, como por ejemplo tener java 7, java 8, php 5 y php 7, todos en la misma máquina sin configuraciones complejas y aisladas unas de otras, pero con la posibilidad de que las aplicaciones interactúen entre sí, de ser requerido.

Así, actualmente el servidor marca HP ProLiant DL120 Gen9 cuenta con las siguientes características:

- Disco duro: 2 discos duros de 1 TB cada uno.
- Memoria RAM: 8 GB DDR4 FB-DIMM con velocidad 2400 MT/s
- Procesador: Intel(R) Xeon(R) CPU E5-2603 v4 @1.70GHz Max Vel 4000 MHz de seis núcleos
- Sistema operativo: Centos 7
- Dominio del servidor: smartcampus.uniajc.edu.co

Las aplicaciones para despliegue instaladas principalmente en esta máquina son:

Apache tomcat 7.0.88 - port-http:8888/ port-http:8443

PostgreSQL 9.5

Java 1.8.0_222

Apache/2.4.6 httpd

Los contenedores desplegados dentro de esta maquina son:

- Php7
- Php56
- SonarQube
- Gitlab
- Jenkins
- MySql

Proceso de gestión de proyectos Ecosistema Smart Campus

La gestión de proyectos dentro del Ecosistema Smart Campus se da orientada por algunos de los lineamientos y recomendaciones del PMBOK (2017), donde los elementos principales del control y seguimiento de la información se encuentran dados por:

- Acta de inicio: artefacto que delimita el alcance del proyecto, determina las personas involucradas y fija el acuerdo de entregas por sprint, para tener un cronograma tentativo a seguir, salvo una eventualidad justificada que por mutuo acuerdo permita mover una fecha de entrega.
- Acta de confidencialidad: documento de aspecto legal que detalla los compromisos adquiridos legalmente, sobre la responsabilidad del acceso y la manipulación de la información presente en el proyecto.
- Matriz de riesgos: artefacto que permite la correcta identificación de riesgos desde el inicio del proyecto y determina las posibles formas de impedirlo o mitigarlo. Es un documento dinámico que sprint a sprint debe estar siendo alimentado para mantener un seguimiento sobre posibles eventualidades.
- Matriz de interesados: artefacto que detalla la información de contacto de las personas involucradas en el proyecto. Sirve de mecanismo de identificación del canal de comunicación y siempre está disponible para todos los interesados.
- Seguimiento plan o informe diario del proyecto: artefacto que permite evidenciar el avance del proyecto sprint a sprint, determinando si se está avanzando a la velocidad deseada y conforme a las metas del sprint. El cálculo de la velocidad del equipo se da mediante la métrica burndown chart, característica del marco de trabajo ágil scrum.



Resultados del Proyecto

**Estandarización proceso
de desarrollo de software**

Gracias a la gestión de la investigadora principal, Mg. Ana Milena Rojas Calero, el proceso de desarrollo de software, explicado en detalle en el capítulo anterior, y todos sus artefactos resultantes de cada etapa fueron estandarizados en el sistema de gestión de calidad de la institución. En la siguiente tabla se relaciona cada artefacto, su descripción y código del documento por etapa

Etapa	Artefacto	Descripción	Código
Gestión de proyecto	Matriz de Riesgo de Proyectos	Documento donde se registra sprint a sprint los riesgos identificados y sus estrategias de control y mitigación.	GIF035 Matriz de Riesgo de Proyectos
	Matriz de Interesados	Documento donde se registran quienes son las personas involucradas en el proyecto, su influencia y sus datos de contacto.	GIF036 Matriz de Interesados
	Acta de Inicio de Proyecto	Documento donde se delimita el alcance del proyecto y se fijan los acuerdos que dan inicio a su ejecución.	GIF037 Acta delnicio de Proyecto SC
	Acuerdo de Confidencialidad	Documento legal que protege la información que la universidad expone a través del proyecto.	GIF040 Acuerdo de Confidencialidad DS
	Seguimiento Plan Informe Diario de Proyectos	Documento utilizado para calcular la velocidad del equipo y determinar así si se cumplirá con lo planeado o se requiere de algún ajuste.	GIF041 Segto Planife Inf. Diario de Proyectos
Análisis	Product Backlog	Documento utilizado para recopilar la información de tipo funcional asociada al proyecto a ejecutar.	GIF044 Historia de Usuarios y CriteriosDS
Diseño de software	Documento de Arquitectura de Software o DAS	Documento que contiene las decisiones arquitecturales que dan forma a la solución.	GIF038Dto.Arq.Software
	Manual de base de datos	Documento que describe las tablas construidas y sus relaciones.	ManualBDAUNIAJC_V1_1
Desarrollo	Código fuente y scripts BD	Son las unidades desplegables del producto.	Cómo tal, no tiene un código dentro del sistema, pero son el resultado final, que es puesto en funcionamiento dentro del servidor de Smart Campus una vez completado cada sprint. Es importante mencionarlo porque consolida el resultado de todo lo anterior.
Despliegue	Control de entregas de productos	Documento que contiene los resultados de la entrega del producto y sus recomendaciones.	GIF042ControldeEntregasde-ProductosDS

Configuración del ambiente de calidad preventiva

A continuación, utilizando el material elaborado en conjunto con los estudiantes William Andrés Leitón Muñoz y Brayan Alexis Moya Loaiza, para el trabajo de grado titulado: "Documentación e implementación de buenas prácticas DevOps con herramientas de automatización de pruebas y de despliegue a

través del refinamiento de software de Smart Campus”, dirigido por el profesor Manuel Alejandro Pastrana Pardo, se describe a continuación detalladamente bajo el enfoque del manual de instalación la configuración requerida para replicar el ambiente mencionado.

Instalación y configuración

Pasos previos

Previo a la configuración del ambiente de calidad preventiva, se requiere actualizar el sistema para evitar fallos de compatibilidad con las herramientas. Esto se realiza por medio de la instalación de paquetes complementarios que proporciona Enterprise Linux (EPEL) (EPEL, s.f.). El siguiente comando es utilizado para esta tarea:

- **sudo yum install epel-release**. Las figuras 39 y 40 reflejan los resultados de este comando.

```

developer@localhost ~]$ sudo yum install epel-release
[sudo] password for developer:
Completamos cargados:fastestmirror
Determining fastest mirrors
 * base: mirror.ula.edu.ec
 * extras: centos.br.isanet.com.br
 * updates: centos.br.isanet.com.br
base                               | 3.6 kB | 00:00:00
extras                              | 2.9 kB | 00:00:00
updates                             | 2.9 kB | 00:00:00
1/21: centos/7/os_64primary_db      | 154 kB | 00:00:01
2/21: updates/7/os_64primary_db    | 6.7 kB | 00:00:00
Resolviendo dependencias
-> Ejecutando prueba de transacción
-> Paquete epel-release.noarch 8-7-11 debe ser instalado
-> Resolución de dependencias finalizada
Dependencias resueltas
=====
Package      Arquitectura  Versión      Repositorio  Tamaño
-----
Instalando:
epel-release
Sumen de la transacción
Instalar: 1 Paquete
=====
Tamaño total de la descarga: 15 k
Tamaño instalado: 24 k
Is this ok [y/n]:

```

Figura 39. Actualización del sistema 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

```
delta RPM disabled because /usr/bin/applydelta not installed.
obsoletencia: /var/cache/yum/x86_64/epel/packages/epel-release-7-12.noarch.rpm: Encabezado00 RPM/00
56 Signature: ID de clave: 352c446c: NOKEY
No se ha instalado la llave pública de epel-release-7-12.noarch.rpm
1/23: epel-release-7-12.noarch.rpm                1 15 kB 00:00:00
1/23: sudo-1.8.23-4.el7_7.2.x86_64.rpm          1 842 kB 00:00:01
-----
Total: 429 kB/s | 656 kB 00:00:01
Resolviendo claves desde file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importando llave GPG 8c35206415:
UsuarioID : "Fedora EPEL (7) (epel@fedoraproject.org)"
Huella    : 8167 7d7c 4d6e 98f1 713e 898f 6b2f a82c 352c 6465
Paquete   : epel-release-7-11.noarch (NoTras)
Detalle   : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Tipo de acuerdo: local
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Actualizando : epel-release-7-12.noarch                1/4
                sudo-1.8.23-4.el7_7.2.x86_64            2/4
Limpiendo    : epel-release-7-11.noarch                3/4
Limpiendo    : sudo-1.8.23-4.el7_7.1.x86_64            4/4
Comprobando  : sudo-1.8.23-4.el7_7.2.x86_64            1/4
Comprobando  : epel-release-7-12.noarch                2/4
Comprobando  : sudo-1.8.23-4.el7_7.1.x86_64            3/4
Comprobando  : epel-release-7-11.noarch                4/4
Actualizado:
  epel-release.noarch 0:7-12                sudo.x86_64 8:1.8.23-4.el7_7.2
!Listo!
desarrollero@localhost ~$
```

Figura 40. Actualización del sistema 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019)

Una vez completado el comando anterior, se procede a con la instrucción **sudo yum update** para actualizar todos los paquetes del sistema y una vez terminado, se reinicia el sistema con el comando **sudo reboot** para que se apliquen correctamente las modificaciones instaladas.

Herramientas necesarias para el proceso

UNZIP

En algunos casos, la información de los instaladores de las herramientas necesarias está en formato comprimido como el ZIP. Por lo anterior, es necesario instalar la herramienta UNZIP, que mediante un comando descomprime estos archivos. Así mismo, la herramienta nos permite también comprimir archivos si lo requerimos. Inicialmente se verifica si está instalado en el sistema con el siguiente comando: **unzip -v**. El resultado puede ser evidenciado en la figura 41.

```
UnZip 6.00 of 20 April 2009, by info-ZIP. Maintained by C. Spieler. Send
bug reports using http://www.info-zip.org/zip-bug.html; see README for details.

Latest sources and executables are at ftp://ftp.info-zip.org/pub/infozip/ ;
see ftp://ftp.info-zip.org/pub/infozip/UnZip.html for other sites.

Compiled with gcc 4.8.5 20150623 (Red Hat 4.8.5-39) for Unix (Linux ELF) on Aug 9 2019.

UnZip special compilation options:
  COPYRIGHT_CLEAN (PKZIP 6.0x unshrinking method not supported)
  SET_DIR_ATTRIB
  SYMBOLS (symbolic links supported, if RTL and file system permit)
  TIMESTAMPS
  UNISOURCE
  USE_EF_UT_TIME
  USE_UNSHRINK (PKZIP/Zip 1.x unshrinking method supported)
  USE_DEFLATED4 (PKZIP 4.x Deflated4(tm) supported)
  UNICODE_SUPPORT (wide-chars, char coding; UTF-8) (handle UTF-8 paths)
  MBCS_SUPPORT (multibyte character support, MB_CUR_MAX = 6)
  LARGE_FILE_SUPPORT (large files over 2 GiB supported)
  ZIP64_SUPPORT (archives using Zip64 for large files supported)
  USE_ZIP64 (PKZIP 4.6+, using help lib version 1.0.6, 6-Sept-2018)
  USE_TEXT_CONV
  (decompression, version 2.11 of 05 Jan 2007)

UnZip and ZipInfo environment options:
  UNZIP: (none)
  UNZIPOPT: (none)
  ZIPINFO: (none)
  ZIPINFOOPT: (none)
desarrollero@localhost ~$
```

Figura 41. Instalación de Unzip 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)

Si no se encuentra instalado, se procede a hacerlo con el comando: **sudo yum -y install unzip**. El resultado puede ser evidenciado en la figura 42.

```
Resolviendo dependencias
-> Ejecutando prueba de transacción
->> Paquete unzip.x86_64 0:6.0-29.el7 debe ser instalado
-> Resolución de dependencias finalizada
Dependencias resueltas
=====
Package      Arquitectura  Versión      Repositorio  Tamaño
Instalando:
unzip        x86_64        6.0-29.el7   base         170 k
=====
Resumen de la transacción
Instalar 1 Paquete
Instalar 1 Paquete
Tamaño total de la descarga: 170 k
Tamaño instalado: 362 k
Downloading packages:
unzip-6.0-29.el7.x86_64.rpm                1 170 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Instalando : unzip-6.0-29.el7.x86_64          1/1
Comprobando : unzip-6.0-29.el7.x86_64        1/1
Instalando:
unzip.x86_64 0:6.0-29.el7
¿listo?
[root@localhost data] #
```

Figura 42. Instalación de Unzip 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)

Finalizada la instalación de la herramienta, esta puede ser verificada mediante el siguiente comando: **which unzip**, como muestra la figura 43, que indicará la ruta donde quedó la herramienta configurada. Finalmente se verifica la versión para garantizar la instalación más actual con el comando: **unzip -v**.

```
[developers@localhost ~]# which unzip
/bin/unzip
```

Figura 43. Ruta de instalación de Unzip 3 en el sistema operativo CentOS 7.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019)

JDK (Java Development Kit)

El kit de desarrollo de java o JDK por sus siglas en inglés, permite la ejecución de algunas de las herramientas requeridas en la configuración indicada por Pastrana et al. (2019), por lo que su instalación en el sistema operativo resulta de gran importancia para el funcionamiento adecuado del ambiente de calidad. Por lo anterior, es necesario verificar si se encuentra o no instalado en el sistema operativo, utilizando el comando: **java -versión**. Si el JDK está instalado, el sistema operativo indicará la versión actual instalada como indica la primera parte de la figura 44. En caso contrario, dará una respuesta de que no reconoce el comando. Así mismo, en la segunda parte de la misma imagen, se evidencia que es posible que cohabiten varias versiones de java dentro del sistema operativo. Gracias al comando: **sudo /usr/sbin/alternatives -config java**, es posible seleccionar una en específico. A modo de ejemplo se selecciona la 1.8 (la única existente para el ejemplo de configuración).

Así mismo, es necesario agregar la ruta absoluta de Java en el mismo archivo en la variable PATH (sin el /bin), para que el sistema detecte el Java y sea posible utilizarlo. La figura 47 muestra el resultado de esta configuración.

```
PATH=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b01-4.el7.7.x86_64:/usr/bin:$PATH
export PATH
```

Figura 47. Ubicación de variable Java.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Completado esto, se hace necesario recargar el **.bash profile** para que el sistema tome los cambios realizados, mediante el comando: **sudo source ~/.bash_profile**. Una forma rápida de validar si la variable de entorno quedó correctamente configurada es ejecutando: **echo \$JAVA_HOME**, que imprime la ruta contenida por la variable, como se ve en la figura 48.

```
root@localhost:~# echo $JAVA_HOME
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b01-4.el7.7.x86_64
root@localhost:~#
```

Figura 48. Variables de entorno 1.

Fuente: (Leiton Muñoz y Moya loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

PostgreSQL 9.2 o versiones más recientes

Para el ejemplo se tomó la versión 9.2, pero es recomendable utilizar versiones más actuales. Esta configuración es necesaria debido a que herramientas como sonarQube lo requieren para guardar el histórico de los análisis estáticos de código realizados. Esto se observará más adelante en las pruebas ejecutadas.

Debido a que por defecto los paquetes de instalación de esta base de datos no están en el sistema operativo, se deben adicionar utilizando el comando: **sudo yum install https://download.postgresql.org/pub/repos/yum/9.2/redhat/rhel-7x86_64/pgdg-redhat92-9.2-24.noarch.rpm -y**, para luego poder ser instalados como indica la figura 49.

```
~/pgdg-redhat96-9.6-3.noarch.rpm ~$
Completados cargados: 1 paquetes erróreo
pgdg-redhat96-9.6-3.noarch.rpm
Combinando /usr/lib/yum-root-ca/oidb/pgdg-redhat96-9.6-3.noarch.rpm: pgdg-redhat-repo-42.0-6.noarch
Rescando /usr/lib/yum-root-ca/oidb/pgdg-redhat96-9.6-3.noarch.rpm para ser instalado
Resolviendo dependencias
--> Ejecutando prueba de transacción
--> Paquete pgdg-redhat-repo.noarch.0:42.0-6 debe ser instalado
--> Resolución de dependencias finalizada

Dependencias resueltas
=====
Package      Arquitectura  Versión      Repositorio  Tamaño
-----
Instalado:   pgdg-redhat-repo      noarch      42.0-6      /pgdg-redhat96-9.6-3.noarch      7.5 k
Resumen de la transacción
=====
Instalar 1 Paquete
Tamaño total: 7.5 k
Tamaño instalado: 7.5 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction:
  Instalando : pgdg-redhat-repo-42.0-6.noarch      1/1
  Comprando  : pgdg-redhat-repo-42.0-6.noarch      1/1
Instalado:
pgdg-redhat-repo.noarch.0:42.0-6
El list:
root@localhost ~#
```

Figura 49. Añadiendo repositorio PostgreSQL.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Cuando el repositorio es reconocido por el sistema operativo, es posible instalar la base de datos mediante el comando: `sudo yum install postgresql92 postgresql92-server postgresql92-contrib postgresql92-libs -y`, como evidencia la figura 50.

```
Instalando : postgresql92-contrib-9.6.17-1PGDG.rhel7.x86_64      5/6
Instalando : postgresql-server-9.2.24-2.el7_7.x86_64           6/6
Comprando  : postgresql92-libs-9.6.17-1PGDG.rhel7.x86_64      1/6
Comprando  : postgresql-server-9.2.24-2.el7_7.x86_64           2/6
Comprando  : postgresql-libs-9.2.24-2.el7_7.x86_64             3/6
Comprando  : postgresql92-contrib-9.6.17-1PGDG.rhel7.x86_64   4/6
Comprando  : postgresql92-9.6.17-1PGDG.rhel7.x86_64           5/6
Comprando  : postgresql-9.2.24-2.el7_7.x86_64                  6/6
Instalado:
postgresql-server.x86_64.0:9.2.24-2.el7_7      postgresql92.x86_64.0:9.6.17-1PGDG.rhel7
postgresql92-contrib.x86_64.0:9.6.17-1PGDG.rhel7      postgresql92-libs.x86_64.0:9.6.17-1PGDG.rhel7
Dependencia(s) instalada(s):
postgresql.x86_64.0:9.2.24-2.el7_7      postgresql-libs.x86_64.0:9.2.24-2.el7_7
El list:
root@localhost ~#
```

Figura 50. Instalación PostgreSQL.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para verificar que la instalación se realizó correctamente, se puede ejecutar el comando: `postgres --versión`, que indica la versión instalada de postgres como muestra la figura 51

```
(developers@localhost ~)$ postgres --version
postgres (PostgreSQL) 9.2.24
```

Figura 51. Verificación PostgreSQL.

Fuente. (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para poder hacer uso de la base de datos instalada es necesario iniciar el servicio mediante el comando: `sudo postgresql-setup initdb`. La figura 52 muestra el resultado de ejecutar correctamente la instrucción.

```
root@localhost ~]# postgresql-setup initdb
Initializing database ... OK
```

Figura 52. Inicializando PostgreSQL.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Es recomendable habilitar PostgreSQL para que se inicie automáticamente cada vez que el sistema arranque, de lo contrario, el paso anterior habría que realizarlo cada vez que el sistema operativo se reinicie. Por tanto, las siguientes instrucciones pueden ser aplicadas:

- `systemctl enable postgresql`
- `systemctl start postgresql`

A modo de ayuda extra, se exponen una serie de comandos utilizados frecuentemente para manipular el servicio de PostgreSQL en caso de ser necesario:

- Verificar estado del servicio: `service postgresql status`
- Iniciar servicio: `service postgresql start`
- Detener servicio: `service postgresql stop`
- Reiniciar servicio: `service postgresql restart`

Cuando la base de datos ya se encuentra instalada y el servicio inicializado, se requiere configurar la contraseña del usuario postgres. El comando a utilizar es: `passwd postgres`. Los resultados de la ejecución de la instrucción pueden ser evidenciados en la figura 53.

```
postgres
root@localhost ~]# passwd postgres
Cambiando la contraseña del usuario postgres.
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
passwd: todos los símbolos de autenticación se actualizaron con éxito.
```

Figura 53. Modificación de Password del usuario postgres

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Así mismo, es necesario realizar algunas configuraciones de seguridad para garantizar correcta conexión de las herramientas a la base de datos, de una manera segura. La primera configuración, consiste en habilitar la autenticación

md5, para que la base de datos solo pueda ser manipulada por usuarios con credenciales de acceso. Para esto se debe editar el archivo de configuración de permisos de la aplicación denominado `pg_hba.conf`. Usando nuevamente el editor NANO, se ejecuta el siguiente comando: `sudo nano /var/lib/pgsql/data/pg_hba.conf`. La figura 54 muestra cómo se debe ver el archivo al abrirlo por primera vez.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 ident
# IPv6 local connections:
host all all ::1/128 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication postgres peer
host replication postgres 127.0.0.1/32 ident
host replication postgres ::1/128 ident
```

Figura 54. Configuración de autenticación antigua.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Dentro del archivo, en la columna **METHOD**, se debe cambiar los argumentos *peer* a *trust* y los *ident* a *md5*. como indica figura 55.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication postgres peer
host replication postgres 127.0.0.1/32 ident
host replication postgres ::1/128 ident
```

Figura 55. Configuración de autenticación actual.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Realizada la edición del archivo y guardados los cambios, se debe reiniciar el servicio PostgreSQL para que aplique las configuraciones realizadas. Se recuerda hacer uso del comando: `sudo service postgresql restart`.

Finalmente, como se mencionó al inicio para la instalación de Sonar se requiere de una base de datos donde almacenar la información requerida por la aplicación. Por lo anterior, se recomienda crear la base de datos previo a la instalación. Esto se puede realizar mediante la instrucción SQL: `create database sonar owner postgres` como se ve la figura 56.

```
postgres=# create database sonar owner postgres;
CREATE DATABASE
postgres=# \l
          Listado de base de datos
+-----+-----+-----+-----+-----+-----+
Nombre | Dueño | Codificación | Collate | CType | Privilegio
+-----+-----+-----+-----+-----+-----+
postgres | postgres | UTF8         | es_CO.UTF-8 | es_CO.UTF-8 |
sonar    | postgres | UTF8         | es_CO.UTF-8 | es_CO.UTF-8 |
template0 | postgres | UTF8         | es_CO.UTF-8 | es_CO.UTF-8 | *c/postgres
template1 | postgres | UTF8         | es_CO.UTF-8 | es_CO.UTF-8 | postgres-C*/postgres
(4 filas)
```

Figura 56. Base de datos creada.

Fuente: (Leiton Muñoz y Moya Loaliza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Herramientas de calidad preventiva Smart campus, modelo actualmente implementado

El modelo expuesto por Pastrana et al. (2019) parte de la interacción que tienen los equipos de desarrollo con el versionador de código, garantizando la propiedad colectiva del código para garantizar un crecimiento evolutivo organizado del producto (Venters et al., 2018). A esta herramienta se suman dos, que son: el Jenkins que permite la integración continua y el analizador estático de código Sonar, que es disparado por la herramienta Jenkins cuando genera los empaquetados de la aplicación que está siendo versionada para verificar que los cambios realizados no dañan la unidad desplegable. Varios autores buscan esta sostenibilidad que garantice la calidad del incremento en todo momento, muy alineados de la mano de prácticas como las recomendadas por DevOps, como plantean Luz, Pinto, & Bonifácio (2019). A continuación, se detalla en orden las instalaciones y configuraciones de cada herramienta:

GIT

Por defecto, CentOS 7 ya tiene integrado los paquetes de instalación de la herramienta Git, por lo que sólo se requiere instalarlos usando el comando: `sudo yum install git`, como se evidencia en la figura 57.

```
perl-Fill-Temp          x86_64           0.23-01.3.e17         httpd      56 kB
perl-Filter            x86_64           1.49-3.e17            httpd      76 kB
perl-Gentoo-Layout     x86_64           2.48-3.e17            httpd      66 kB
perl-Git               x86_64           1.0.3-1-21.e17_7     upstream  525 kB
perl-HTTP-Tiny         x86_64           0.033-3.e17           httpd      82 kB
perl-PackTool          x86_64           3.48-5.e17            httpd      21 kB
perl-Pod-Escapes       x86_64           1.1.0-294.e17_6      httpd      87 kB
perl-Pod-Parser        x86_64           3.28-4.e17           httpd      216 kB
perl-Pod-Style         x86_64           1.63-3.e17           httpd      27 kB
perl-Scalar-List-Utils x86_64           1.27-240.e17         httpd      286 kB
perl-Stockfish         x86_64           2.018-4.e17          httpd      49 kB
perl-StrayCat          x86_64           2.42-3.e17           httpd      77 kB
perl-TermReadKey       x86_64           2.38-28.e17          httpd      31 kB
perl-Text-Favorites    x86_64           1.039-2.e17          httpd      13 kB
perl-Time-HiRes        x86_64           4.1.1-922-3.e17     httpd      45 kB
perl-Time-Locales      x86_64           1.2288-2.e17        httpd      24 kB
perl-ccoolant          x86_64           1.27-2.e17           httpd      13 kB
perl-File              x86_64           4.5.10-3-294.e17_6  httpd      680 kB
perl-Image             x86_64           4.5.10-3-294.e17_6  httpd      44 kB
perl-JSON              x86_64           1.9.220-204.e17     httpd      12 kB
perl-JSON-XS           x86_64           2.5.1-3.e17         httpd      112 kB
perl-Thread            x86_64           1.07-1.e17          httpd      19 kB
perl-threads-shared    x86_64           1.43-6.e17         httpd      33 kB
rpm-gpg                 x86_64           3.1.2-6.e17_6.1     httpd      494 kB

Resumen de la transacción
-----
Instalar 2 Paquetes (+38 Paquetes dependientes)
Espacio total de la descarga: 16 M
Espacio instalado: 59 M
Is this ok [y/n]:
```

Figura 57. Instalación de Git.

Fuente: (Leiton Muñoz y Moya Loaliza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para verificar la correcta instalación basta con aplicar el comando: **git--versión**, que debe mostrar un resultado similar al de la figura 58.

```
[root@localhost ~]# git --version
git version 1.8.3.1
[root@localhost ~]#
```

Figura 58. Versión actual de Git.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Una vez instalado el versionador, se requiere configurar el usuario y correo global, con el fin de poder autenticarse de manera remota. Para esto es necesario utilizar los comandos:

- `git config --global user.name "tu nombre"`
- `git config --global user.email "tucorreo@ejemplo.com"`

Para confirmar que las configuraciones han surtido efecto se debe ejecutar el comando: **sudo git config --list**, como muestra la figura 59.

```
[root@localhost ~]# git config --global user.name "baoya"
[root@localhost ~]# git config --global user.email "baoya17@hotmail.com"
[root@localhost ~]# git config --list
git: 'config' is not a git command. See 'git --help'.
Did you mean this?
config
[root@localhost ~]# git config --list
user.name=baoya
user.email=baoya17@hotmail.com
[root@localhost ~]#
```

Figura 59. Verificación de configuraciones.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Es recomendable instalar el paquete `libcurl-devel` mediante el comando: **sudo yum install curl-devel**. Esto se hace para evitar, como sucede en algunos casos, problemas con comandos como: `git clone`. Para que la biblioteca quede configurada es necesario ejecutar la sentencia `--with-curl`. El comando a ejecutar es siguiente:

- `sudo ./configure --prefix=/usr --with-curl`

SonarQube 7.1.

El analizador estático de código seleccionado, según el trabajo de Pastrana et al. (2019), para el Ecosistema Smart Campus es SonarQube. Lo primero que se debe realizar es obtener los paquetes necesarios para su instalación, con el siguiente comando:

- `sudo wget -O /etc/yum.repos.d/sonar.repo http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo`

Descargados e incluidos los paquetes dentro del sistema operativo, se procede a hacer la instalación mediante el comando: `sudo yum -y install sonar`. Así mismo, una vez instalado, es necesario saber dónde queda configurada la herramienta para futuras configuraciones. Por lo tanto, para buscar la ruta se puede utilizar la siguiente instrucción: `sudo find / -name sonar`. El resultado de la ejecución del último comando puede ser evidenciado en la figura 60, que indica que el directorio de Sonar quedó ubicado en la ruta: `/opt/sonarqube-7.1` para este caso.

```

[developers@localhost ~]$ sudo find / -name sonar
/etc/rc.d/init.d/sonar
/usr/lib/yum/repos/x86_64/7/sonar
/usr/lib/jenkins/plugins/sonar
/usr/lib/jenkins/plugins/sonar/META-INF/maven/org.jenkins-ci.plugins/sonar
/usr/cache/yum/x86_64/7/sonar
/usr/bin/sonar
/opt/sonar
[developers@localhost ~]$
display all 148 possibilities? (y or n)
[developers@localhost ~]$
display all 148 possibilities? (y or n)
[developers@localhost ~]$

```

Figura 60. Búsqueda de directorio SonarQube.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para facilitar su manipulación, es posible cambiar el nombre del directorio si así se desea. En este ejercicio se realiza el cambio del nombre del directorio `sonarqube-7.1` por `sonar` mediante el comando: `sudo mv /opt/sonarqube-7.1 /opt/sonar`. El resultado puede ser evidenciado en la figura 61.

```

[root@localhost data]# mv /opt/sonarqube-8.2.0.32929/ /opt/sonarqube
[root@localhost data]# cd /opt/sonarqube/
[root@localhost sonarqube]# ls
bin conf COPYING data elasticsearch extensions lib logs temp web
[root@localhost sonarqube]# _

```

Figura 61. Modificación del nombre de directorio SonarQube.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Una vez instalada la aplicación se debe proceder con las configuraciones respectivas de SonarQube para su adecuado funcionamiento. Para esto, es necesario manipular el archivo **sonar.properties**, con el editor de archivos nano usando el siguiente comando: **sudo nano /opt/sonar/conf/sonar.properties**. Dentro de este archivo, se deben realizar las siguientes modificaciones, en caso de que algunas de las instrucciones estén comentadas descomentarlas:

- Configurar el usuario y password de acceso a la base de datos. En este caso se debe utilizar el usuario y password de postgres configurado en los pasos previos. Para ello se deben modificar las líneas de la siguiente manera: **Sonar.jdbc.username=postgres** **Sonar.jdbc.password=admin**. La figura 62 muestra el archivo en el que se modifican las dos líneas anteriores.

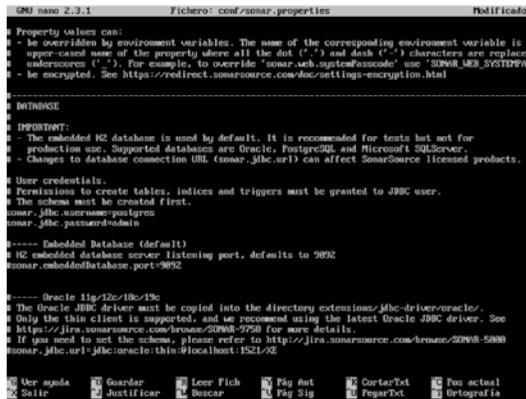


Figura 62. Configuración de password y usuario de PostgreSQL en SonarQube.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Del mismo modo es necesario identificar y modificar el nombre de la base de datos por la que fue creada en los pasos previos con el nombre de sonar. La línea a modificada debe quedar así: **sonar.jdbc.url=jdbc:postgresql://localhost/sonar**, ver figura 63.



Figura 63. Modificación de nombre de base de datos PostgreSQL.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Sonar al igual que Jenkins requiere de una IP y un puerto específico para poder acceder a este servicio desde otras máquinas. Por defecto se ejecuta en el puerto 9000. Por tanto, es necesario adicionar este puerto o el que deseemos colocar disponible para este servicio, dentro de las configuraciones del firewall del servidor. Esto es posible mediante la siguiente instrucción: **sudo firewall-cmd --zone=public --permanent --add-port=9000/tcp**. Una vez realizado, es necesario reiniciar el firewall, con el comando: **sudo firewall-cmd --reload**, para que tome los cambios aplicados. La figura 64 refleja los cambios realizados.

```
firewall-cmd: error: argument --add-port: not allowed with argument --add-service
[root@localhost system]# Firewall-cmd --zone=public --permanent --add-port=9000/tcp
success
[root@localhost system]# Firewall-cmd --reload
success
[root@localhost system]#
```

Figura 64. Configuración de permisos de puerto 9000.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Adicionalmente, es necesario agregar las siguientes variables en el archivo de configuración **sonar.sh**:

- Ruta: **/opt/sonar/bin/Linux-x86-64/sonar.sh**
- Variable 1: **SONAR_HOME=/opt/sonar**
- Variable 2: **PLATFORM=Linux-x86-64**
- **WRAPPER_CMD="\${SONAR_HOME}/bin/\${PLATFORM}/wrapper"**
- **WRAPPER_CONF="\${SONAR_HOME}/conf/wrapper.conf"**

Es importante resaltar que es necesario configurar la versión de Java a utilizar dentro del Wrapper para que así la aplicación pueda ser ejecutada, ver figura 65. Esta configuración se realiza dentro del archivo **wrapper.conf**, que se encuentra en este caso en la ruta: **/opt/sonar/conf/wrapper.conf**. La línea a modificar bajo la configuración realizada en los pasos previos es:

- **wrapper.java.command=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64/jre/bin/java**

```
bin/wrapper.jar command-path/bin/wrapper.jar  
bin/wrapper.jar command-path/bin/wrapper.jar 1.0.0 openjdk 1.8.252 64-bit 7.506 64-bit/bin/wrapper.jar
```

Figura 65. Ruta del Java en la configuración del Wrapper.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para que el sistema operativo tome todos los cambios realizados, Sonar debe ser reiniciado mediante el comando: **sudo service sonar restart**. El resto de la configuración de la aplicación Sonar se realiza directamente accediendo desde un navegador web. Esto abrirá un menú de instalación de las configuraciones básicas que finalmente lleva al ingreso de credenciales, las cuales por defecto son: admin para el usuario y la contraseña. Esto debe ser cambiado después de iniciar sesión la primera vez, por razones de seguridad.

Una vez ingresado en el sistema es posible crear proyectos para que sean analizados. Cuando se crea un proyecto la herramienta genera un token para identificar el proyecto a analizar, como indica la figura 66. Esta llave es fundamental para que Sonar pueda interactuar con otras aplicaciones como Jenkins de manera segura o con versionadores de código en la nube como bitbucket, github y gitlab.

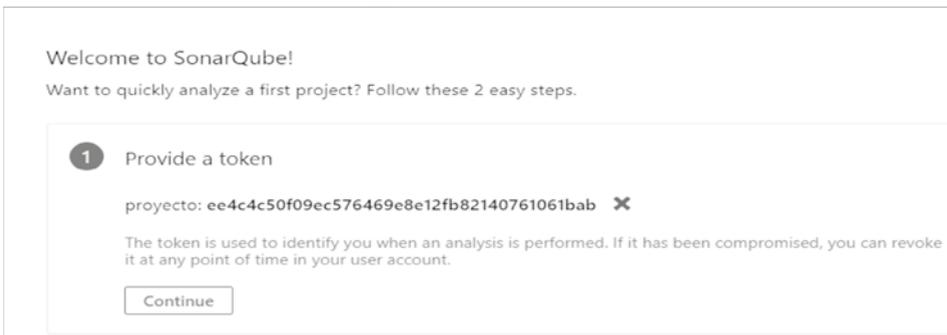


Figura 66. Inicio SonarQube.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Otro requisito importante cuando se crean proyectos, es que la herramienta **solicita** indicar el **lenguaje de programación del proyecto** que se analizará. Existen 3 opciones: Java, C# o VB.Net y otros. Así mismo solicitará la llave y el sistema operativo como indica la figura 67.

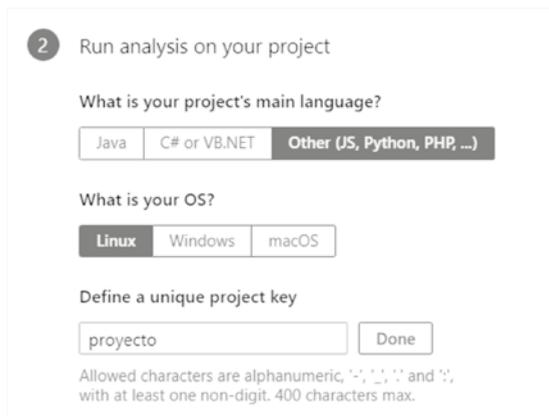


Figura 67. Configuración inicial de proyecto.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

La aplicación recomendará descargar sonar-scanner para el análisis de código, ver figura 68. Más adelante, se procederá con la instalación y configuración, por el momento en este punto se recomienda cerrar la ventana y continuar.



Figura 68. Requerimientos de SonarQube.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Adicionalmente, entre las configuraciones del proyecto también es importante **crear un usuario específico del proyecto** para, de este modo, no trabajar todo con el usuario administrador. Para realizar esto, se debe acceder a la *pestaña de Administración*, que se encuentra en el panel de navegación superior. Dentro de esta interfaz, seleccionar la opción *Seguridad e ingresar a Usuarios*. Ahí se debe presionar el botón azul de *Crear usuario* para realizar esta labor, como indica la figura 69.



Figura 69. Menú Usuarios en SonarQube.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En esta pantalla se deben llenar los datos del usuario como muestra la figura 70.

The image shows the 'Crear usuario' form in SonarQube. It contains several input fields: 'Iniciar sesión *' with the value 'bmoya', 'Nombre *' with the value 'Brayan Moya', 'Correo electrónico' with the value 'bmoya17@hotmail.com', and 'Contraseña *' which is masked with dots. There is an 'Añadir' button for 'Cuentas SCM'. Below the form, a note states: 'El inicio de sesión y el correo electrónico se consideran automáticamente como cuentas SCM'. At the bottom right, there are 'Crear' and 'Cancelar' buttons.

Figura 70. Creación de Usuario en SonarQube.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Una vez realizado esto, es recomendable configurar los permisos globales de Sonar para entregar acceso granular a los roles, evitando que cualquier usuario pueda ejecutar ciertas tareas en los proyectos creados y solamente el administrador tenga permisos totales sobre la aplicación. Dentro de la misma interfaz de Administración, seleccionar la opción Seguridad y después *Permisos globales*. Ver figura 71. En el buscador se debe ingresar el usuario creado en el paso anterior y seleccionar los permisos que se le quieren dar.

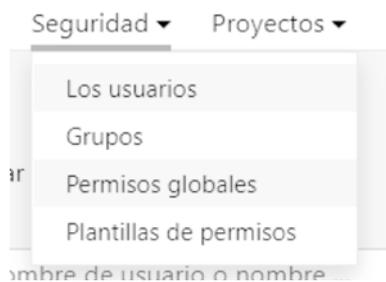


Figura 71. Configuración los permisos globales 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Con esto las configuraciones básicas de permisos quedan finalizadas y se puede proceder a revisar el proyecto. Para ello, es necesario dar clic en la opción *Proyectos* e ingresar a *Gestión de proyectos*. Dentro de esta interfaz, se observará el proyecto creado con su nombre, llave y último análisis (si tiene). Del lado derecho de la columna Último análisis hay un botón en forma de engranaje. Seleccionar y después ingresar en **Editar permisos**. Ver figura 72.



Figura 72. Configuración de permisos del proyecto 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Dentro de la interfaz de Permisos se debe colocar el proyecto como Privado, para que sólo usuarios con permisos de acceso a este proyecto puedan verlo. Luego en la opción Todos, dentro del buscador, se debe ingresar el usuario creado y darle los permisos correspondientes que tendrá en este en el proyecto, como indica la figura 73.

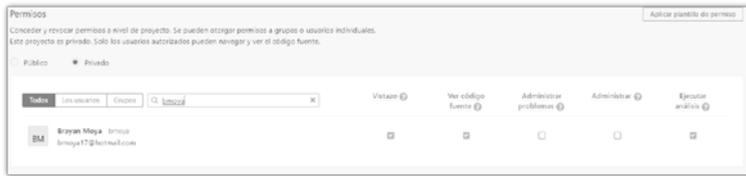


Figura 73. Configuración de permisos del proyecto 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En caso de que Sonar no se inicie correctamente en el navegador, se puede intentar las siguientes posibles soluciones:

- Verificar que el estado del servicio con el comando: **sudo service sonar status**. Si no ha sido iniciado utilizar: **sudo service sonar start**
- Verificar que se haya configurado correctamente el archivo **properties**. Revisar las variables agregadas y la ruta bin de Java en el **wrapper.conf**. Modificarlas de ser necesario.
- Verificar que la IP sea la correcta al igual que el puerto en el que se ejecuta Sonar. Esto se puede revisar en el archivo properties de Sonar haciendo uso del editor de línea de comandos nano mediante la instrucción: **sudo nano /opt/sonar/conf/sonar.properties**, aquí revisar en la línea: **sonar.web.port=** y verificar que sea el mismo con el que se está intentando acceder. Si es una máquina verificar la IP que proporciona.
- Verificar si se encuentra abierto el puerto para acceso público. Si hay dudas con respecto a esto, ejecutar la siguiente línea: **sudo firewall-cmd --zone=public --query-port=9000/tcp**. Si el puerto está agregado, la línea de comandos indicará Yes. En caso contrario, se debe añadir el puerto al firewall de manera correcta, como se explicó al inicio de las configuraciones.
- Si el error persiste, verificar el archivo de log de errores de Sonar. Esto permite revisar otros errores no tan comunes al ejecutar el servicio. Cuando el servicio funciona correctamente, en el archivo se debe de ver una línea que indica SonarQube is up. El archivo puede ser editado con el comando: **sudo nano /opt/sonar/logs/sonar.log**

Sonar Scanner

Una vez configurado Sonar, es necesario instalar y configurar la herramienta Sonar Scanner para SonarQube que permite realizar el análisis estático

de código sobre los proyectos. Para instalar esta herramienta es necesario descargar mediante el siguiente comando el archivo empaquetado: **sudo wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.2.0.1873-linux.zip**. Para descomprimirlo usar el comando: **sudo unzip sonar-scanner-cli-4.2.0.1873-linux.zip**

Para que la herramienta funcione adecuadamente es necesario actualizar la configuración global para apuntar al servidor SonarQube configurado anteriormente. Esto se realiza editando el archivo de propiedades de Sonar Scanner con la instrucción: **sudo nano /opt/sonar-scanner-4.2.0.1873-linux/conf/sonar-scanner.properties**. Modificar como indica la figura 74.



Figura 74. Actualización de configuración global.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Adicionalmente, se deben agregar las variables de entorno de Sonar y Sonar Scanner en el PATH del sistema (figura 75), mediante los siguientes comandos:

- SONAR_HOME=/opt/sonar/bin/
- SONAR_SCANNER_HOME=/opt/sonar-scanner-4.2.0.1873-linux/bin/

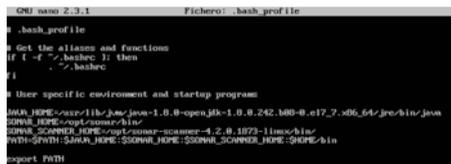


Figura 75. Configuración de las variables de entorno de Sonar y Sonar Scanner.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Al finalizar se deben guardar los cambios y luego recargar las configuraciones mediante el comando: **sudo source ~/.bash_profile**.

- Parar servicio: **sudo service jenkins stop**
- Reiniciar servicio: **sudo service jenkins restart**

```
(developers@localhost ~)$ sudo service jenkins start
Starting jenkins (via systemctl):
[ OK ]
(developers@localhost ~)$ sudo service jenkins status
jenkins.service - LSB: Jenkins Automation Server
Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)
Active: active (running) since jue 2020-05-20 20:53:02 -05; 3min 25s ago
Docs: man:systemd-sysv-generator(8)
Process: 1127 ExecStart=/etc/rc.d/init.d/jenkins start (code=exited, status=0/SUCCESS)
Group: /system.slice/jenkins.service
└─1586 /etc/alternatives/java -Dcom.sun.akuma.Daemon-daemonized -Djava.out.headless=tr...

may 20 20:52:55 localhost.localdomain systemd[1]: Starting LSB: Jenkins Automation Server...
may 20 20:52:55 localhost.localdomain rusermgr[1153]: pam_unix(rusermgr:session): session opened ...
may 20 20:53:02 localhost.localdomain jenkins[1127]: Starting Jenkins [ OK ]
may 20 20:53:02 localhost.localdomain systemd[1]: Started LSB: Jenkins Automation Server.
Hint: Some lines were ellipsized, use -l to show in full.
(developers@localhost ~)$ sudo service jenkins stop
Stopping jenkins (via systemctl):
[ OK ]
(developers@localhost ~)$
```

Figura 77. Estado del servicio Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Jenkins utiliza por defecto el puerto 8080 al instalarlo por primera vez. Bajo el enfoque del top 10 de OWASP, trabajar con puertos por defecto no es recomendado, por tanto se cambia en esta configuración por el 9090. Para hacer el cambio basta con ingresar a editar el siguiente archivo de configuración **Jenkins** ubicado en la ruta: **/etc/sysconfig/Jenkins**. Su resultado puede ser evidenciado en la figura 78. El comando utilizado para editar el archivo es: **sudo nano /etc/sysconfig/Jenkins**.

```
jenkins-9090.service
# Whether to skip potentially long-running clean up at the
# ENDING type location. Do not enable this "true", unless
# you know what you're doing. See ENDING-25273.
#
ENDING_INITIAL_SKIP_CLEAN="false"
#
# Type: string
# Default: ""
# ServiceRestart: jenkins
#
# Options to pass to java when running Jenkins.
#
ENDING_JAVA_OPTIONS="-Djava.out.headless=true"
#
# Type: list(see(8))
# Default: ""
# ServiceRestart: jenkins
#
# Don't Jenkins is listening on:
# Set to -1 to disable
#
ENDING_PORT="9090"
#
# Type: string
# Default: ""
# ServiceRestart: jenkins
#
# IP address Jenkins listens on for HTTP requests.
# Default is all interfaces (0.0.0.0).
#
ENDING_LISTEN_ADDRESS=""
```

Figura 78. Cambio de puerto „8080“ por „9090“.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para que los cambios surtan efecto, se requiere el reinicio del servicio Jenkins mediante el comando: **sudo service jenkins restart**, como muestra la figura 79.

```
root@localhost developers]# service jenkins restart
Restarting jenkins (via systemctl): [ OK ]
root@localhost developers]# _
```

Figura 79. Reinicio de Jenkins.

Fuente. (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para hacer uso de Jenkins, es necesario invocarlo desde un navegador a través de la IP del servidor y el puerto disponible que permiten su acceso. Igual que en el caso de la configuración de SonarQube, se hace necesario permitir el tráfico entrante en el puerto que se configure autorizandolo en el firewall del servidor. En este caso, por cuestiones de seguridad, no se revelará el puerto del servicio real de Smart Campus y se trabajará con un puerto de ejemplo para la configuración. Los comandos utilizados y su resultado puede ser evidenciado en la figura 80:

- `sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp`
- `sudo firewall-cmd --reload`

```
root@localhost ~]# firewall-cmd --zone=public --permanent --add-port=8080/tcp
Warning: ALREADY_ENABLED: 8080:tcp
success
root@localhost ~]# firewall-cmd --reload
success
root@localhost ~]# _
```

Figura 80. Permitir acceso de visitantes a Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

La primera vez que se realiza el ingreso a la herramienta se pedirá la contraseña inicial de Jenkins, la cual se encuentra en la ruta: `/var/lib/jenkins/secrets/initialAdminPassword`. Nuevamente utilizando el editor de texto nano, se accede a la ruta con el siguiente comando:

- `sudo nano /var/lib/jenkins/secrets/initialAdminPassword`

Se ingresa al archivo, se copia la clave que aparece y se pega en el campo de Administrator password, como muestra la figura 81.



Figura 81. Prueba de Jenkins en navegador 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Luego de realizar el paso anterior, la aplicación redirige a un login donde el usuario y contraseña serán admin por defecto, ver figura 82.



Figura 82. Prueba de Jenkins en navegador 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En caso de fallos al iniciar el servicio, se recomienda intentar algunas de las siguientes recomendaciones:

- Verificar el estado del servicio en el servidor con el comando: **sudo service jenkins status**
- En caso de no estar corriendo, iniciarlo mediante la instrucción: **sudo service jenkins start**
- Verificar si Java está instalado en el sistema operativo usando el comando: **java -version**
- Verificar que la IP pública proporcionada por el servidor sea la misma con la que se está ingresando en el navegador. Si es una máquina virtual, verificar la IP que proporciona.

- Verificar el puerto de acceso a Jenkins y que sea el mismo que se invoca desde el navegador. Recordar que esa configuración se encuentra en el archivo accesible con el comando: **sudo nano /etc/sysconfig/Jenkins** en la línea **JENKINS_PORT=**
- Verificar si el puerto que utiliza Jenkins está abierto. Ejecutar la siguiente línea: **sudo firewall-cmd --zone=public --query-port=9090/tcp**. Debe indicar como resultado yes, en caso contrario volver al paso en el que se añade el puerto a las reglas del firewall.
- Si el servicio no inicia ya habiendo ejecutado las anteriores posibles soluciones, se recomienda verificar el archivo log de Jenkins, y así controlar los errores que arroje. El archivo se encuentra accesible con el comando: **sudo nano /var/log/jenkins/jenkins.log**

Para realizar la configuración de plugins necesarios para integrar las herramientas previamente instaladas, se requiere primero ingresar al servicio de Jenkins desde el navegador y en la interfaz principal seleccionar *Administrar Jenkins*, que se encuentra en el menú al lado izquierdo como se muestra en la figura 83.

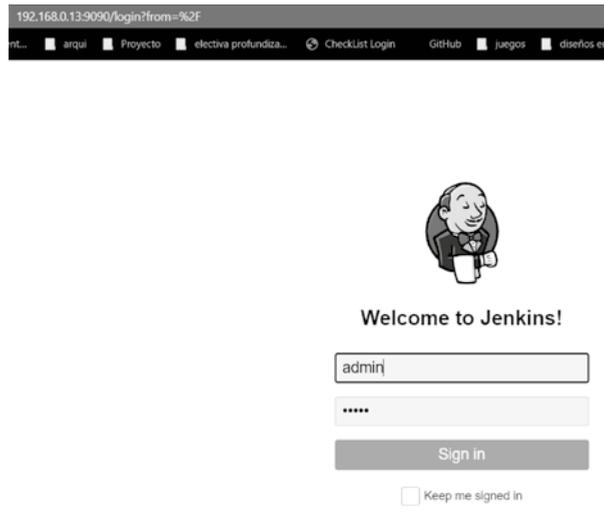


Figura 83. Instalación de plugins en Jenkins 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Dentro de la interfaz de *Administrar Jenkins* entrar a *Administrar plugins* como se ilustra en la figura 84:



Figura 84. Instalación de plugins en Jenkins 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Después ingresar a la opción, en lado derecho, parte superior, se encuentra un buscador. Introducir los nombres de los plugins que se desean descargar, en este caso GitHub y SonarQube Scanner para seleccionarlos, como se muestra en las figuras 85 y 86 respectivamente.

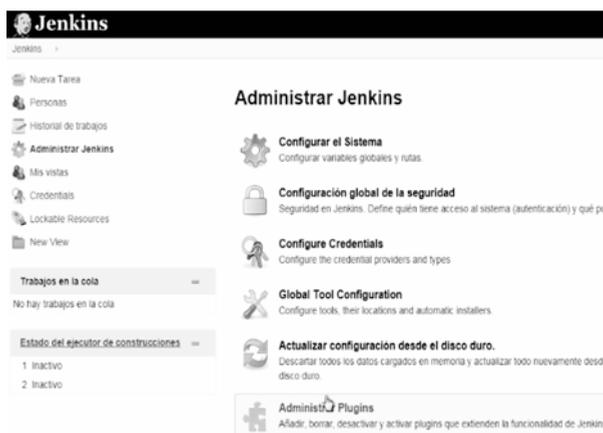


Figura 85. Instalación de plugin GitHub en Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019



Figura 86. Instalación de plugin Sonar en Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Adicionalmente, se instalarán tres plugin más. Uno es el plugin de **JUnit** para ver los reportes en formato XML, ver figura 87. Otro es el plugin de katalon, que se encargan de mostrar los reportes que proporciona **Katalon** (herramienta para automatización de pruebas) luego de su ejecución. Por último, el **Test Results Analyzer Plugin** sirve para generar reportes gráficos de la ejecución del proyecto de Jenkins.



Figura 87. Plugin JUnit.

Fuente: (Leiton Muñoz Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Teniendo los plugins seleccionados, presionar en *Descargar ahora e instalar después de reiniciar* para instalar los plugins, ubicado en la parte inferior como se muestra en la figura 88. Realizado esto se mostrará la interfaz de la figura 89 donde se visualizará el proceso de descarga e instalación de los plugins seleccionados. Seleccionar el campo *Reiniciar Jenkins cuando termine la instalación y no queden trabajos en ejecución*. Esto permite reiniciar automáticamente tanto el servicio de Jenkins como el navegador (ver figura 90).

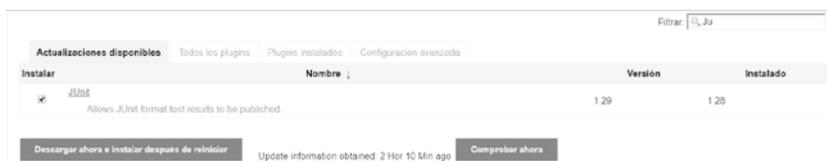


Figura 88. Instalación de plugins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019



Figura 89. Instalación de plugins en Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Instalando/Actualizando plugins

Preparación

- Probando conectividad con Internet
- Probando conectividad con jenkins-ci.org
- Correcto

Mailer

Descarga correcta. Se activará en el próximo arranque.

⇒ [Volver al inicio de la página](#)
(puedes empezar a usar los plugins instalados inmediatamente)

⇒ Reiniciar Jenkins cuando termine la instalación y no queden trabajos en ejecución

Figura 90. Reinicio de servicio y de navegador después de instalación de plugins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Configuración de herramientas en Jenkins

Dentro de la interfaz de Administrar Jenkins, se selecciona la opción de *Configurar el sistema*. En la sección de *JDK*, se desactiva la casilla *Instalar automáticamente*, y se recomienda adicionar la que se configuró previamente. Para esto se debe dar un nombre (puede ser la versión que se esté usando) y se añade en el campo *JAVA_HOME* la misma ruta que se configuró anteriormente para el sistema como variable de entorno *JAVA_HOME*, ver figura 91. Además, se le adiciona el directorio *jre/bin/java* a la ruta indicada.



Figura 91. Configuración JDK.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En la sección Git *desactivar la casilla Instalar automáticamente*. Se agrega la ruta del directorio bin de Git del sistema y se le da un nombre a esta, como se evidencia en la figura 92.



Figura 92. Configuración Git.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En la sección de *SonarQube Scanner*, se selecciona *Añadir SonarQube Scanner* (ver figura 93), desactivando la casilla *Instalar automáticamente* y agregando la ruta donde quedó instalado Sonar (ver figura 94).



Figura 93. Configuración SonarQube Scanner.

Fuente: (Willian Andres Leiton Muñoz Brayan Alexis Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019



Figura 94. Configuración SonarQube Scanner 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En la sección Instalaciones de SonarQube, se agrega la ruta donde esta instalado el servicio de Sonar, como indica la figura 95.

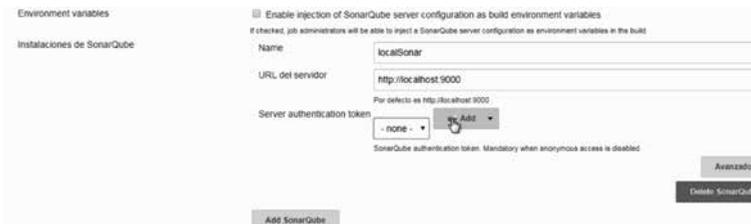


Figura 95. Configuración SonarQube Scanner 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Una vez dentro de esta configuración es necesario darle las credenciales con las que se inicia el servicio de sonar, ver figura 96.

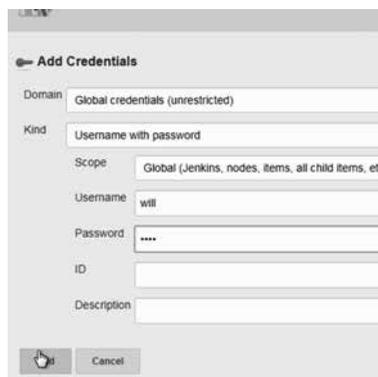


Figura 96. Configuración de SonarQube Scanner 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Terminado lo anterior, guardar los cambios realizados presionando el botón save, como muestra la figura 97.



Figura 97. Configuración de SonarQube Scanner 4.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Creando y configurando un proyecto

En este caso se realizará la configuración de ejemplo de un proyecto. Estos pasos servirán para cualquier tipo de proyecto, donde la única variación es el tipo de lenguaje utilizado de codificación. Para esto, en *la interfaz principal de Jenkins*, ingresar a la opción de *Nueva Tarea* ubicada en el menú, lado izquierdo, ver figura 98.



Figura 98. Configuración de Jenkins para proyecto de prueba en CentOS 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Dentro de esta opción se debe elegir el tipo de proyecto a crear, en este caso será un proyecto de estilo libre, debido a que se seleccionarán qué herramientas externas se integrarán y se tendrá un modo de ejecución personalizado. Adicionalmente, deber darle un nombre y presionar en el botón OK que se encuentra en la parte inferior izquierda, como se evidencia en la figura 99.



Figura 99. Configuración de Jenkins para proyecto de prueba en CentOS 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Una vez creado el proyecto, se habilitará un ítem con el mismo nombre que se ingresó en el paso anterior, dar clic en él como se muestra en la figura 100.



Figura 100. Configuración de Jenkins para proyecto de prueba en CentOS 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Esto abrirá una interface donde se debe ingresar a la pestaña *Configurar*, ubicada en el menú de lado izquierdo, ver figura 101. Esto permitirá la integración entre las herramientas previamente instaladas y su ejecución desde Jenkins.

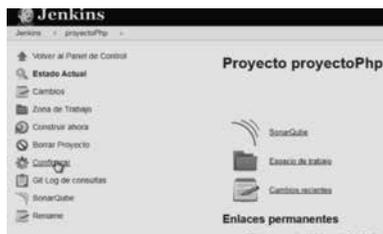


Figura 101. Configuración de Jenkins para proyecto de prueba en CentOS 4.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Dentro de estas configuraciones se debe dar clic en añadir (*Add*), y dentro de esta interfaz, como se puede ver en la figura 103. Aquí es necesario *configurar la dirección y las credenciales de acceso del repositorio Git*, donde está versionado el aplicativo, indicando la rama a la cual debe apuntar, en este caso master, como se evidencia en la figura 103.



Figura 102. Configuración de credenciales Git 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019



Figura 103. Configuración de credenciales Git 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Gracias a esto, el proyecto detectará cambios realizados en la rama del versionador y disparará automáticamente la ejecución de Jenkins para hacer las revisiones pertinentes de IC, y de las otras herramientas configuradas. Es importante resaltar que la herramienta permite realizar revisiones periódicas también. Para esto se debe configurar cada cuanto se realiza la revisión dentro de la sección *Disparadores de ejecuciones*. Es necesario seleccionar la casilla *Consultar repositorio (SCM)*, y aquí ingresar la expresión de tiempo para revisión, en este caso quedaría de la siguiente manera (H/5 * * * * *). Esto significa que cada 5 minutos el proyecto se ejecutará con un factor hash, evitando que todos los trabajos (Git, Sonar, Katalon y Reports) se ejecuten al mismo tiempo, ver evidencia en la figura 104.



Figura 104. Disparador de ejecuciones en Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Así como fue configurado el repositorio, se hace la integración con el analizador estático de código SonarQube. Dentro de la sección *Ejecutar*, seleccionar *Añadir un nuevo paso* y seleccionar *Ejecutar SonarQube Scanner*, como se muestra en la figura 105.



Figura 105. Añadir un nuevo pasó en Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Agregar el JDK previamente configurado y en la sección de *Analysis properties* insertar las siguientes líneas:

```
# Url del servidor donde se aloja sonar
sonar.host.url=http://192.168.0.22:9000

# Credenciales de acceso a la BD de SonarQube
sonar.jdbc.username=postgres
sonar.jdbc.password=admin

# Url de acceso a la BD
sonar.jdbc.url=jdbc:postgresql://localhost/sonar

# Credenciales de acceso al proyecto de SonarQube donde se muestran los resultados
```

```
del análisis
# ID único del proyecto
sonar.projectKey=proyecto
# Nombre del proyecto
sonar.projectName=proyecto

# Versión
sonar.projectVersion=1.0

# Dirección de la carpeta creada en Jenkins para el proyecto
sonar.projectBaseDir=/var/lib/Jenkins/workspace/prueba\ java

# Ubicación de la carpeta que contiene el código a analizar
sonar.sources=src

# Esta variable le indica a sonar que lenguaje será analizado, en caso de no especificar
alguno, se realizará el análisis con los lenguajes que tenga instalado
sonar.language=php

# Formato de codificación
sonar.sourceEncoding=UTF-8
```

La configuración deberá quedar como indica la figura 106:



Figura 106. Código de ejecución Sonar 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Se aplican y se guardan los cambios, para seguir con la instalación de Katalon y su configuración dentro de Jenkins.

Katalon

Antes de iniciar la instalación, es indispensable crear una cuenta en la página oficial (<https://www.katalon.com>) para poder utilizar la herramienta. Así mismo se recomienda revisar los requisitos del sistema y compatibilidad con los navegadores según (Katalon Studio: Entornos Soportados, s.f.), que se muestran en las Figuras 107 y 108, respectivamente.

Requisito	
SO	Windows 7, Windows 8, Windows 10, macOS 10.11+, Linux (basado en Ubuntu)
Procesador	Mínimo : 2 GHz o más rápido Procesador de 32 bits (x86) o 64 bits (x64)
Memoria	<p>Katalon Studio</p> <p>Mínimo : 2 GB de RAM (32 bits) o 4 GB de RAM (64 bits)</p> <p>Recomendado : 4 GB de RAM (32 bits) u 8 GB de RAM (64 bits).</p>
Motor de ejecución	<p>Katalon Runtime Engine</p> <p>Mínimo : 2 GB de RAM (32 bits) o 4 GB de RAM (64 bits)</p> <p>Recomendación para ejecuciones concurrentes (y ejecución con Docker) : el número de sesiones concurrentes x 2GB.</p> <p>Ejemplo : la RAM recomendada para 3 sesiones de ejecución simultánea es de 6 GB (3 x 2 GB).</p>
Almacenamiento	Al menos 1 GB de espacio disponible en el disco duro. Se requiere espacio en disco adicional si se ejecutan pruebas concurrentes.

Figura 107. Requisitos del sistema para Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Navegadores			
Navegadores de escritorio	Versión en Windows	Versión en macOS	Nota
explorador de Internet	9, 10, 11	N / A	Configuraciones de IE requeridas: Configuraciones de Internet Explorer.
Microsoft Edge	18 años	N / A	
Microsoft Edge (chrome)	80+	80+	Compatible con Katalon Studio versión 7.3+
Firefox	56+	56+	Para usar Firefox 57 con Katalon Studio, use Katalon Studio v5.1 +
Google Chrome	58+	58+	
Ópera	N / A	N / A	
Safari	N / A	12+	

Figura 108. Navegadores compatibles con Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Instalaciones adicionales para la correcta ejecución de las pruebas de Katalon

Como el servidor de Smart Campus utiliza el sistema operativo CentOS, que no posee interface gráfica y Katalon es una aplicación de entorno gráfico, es necesario instalar Chrome (o el navegador preferido permitido por Katalon, ver

figura 108) e instalar el complemento Xvfb. Este complemento es un servicio de visualización que realiza operaciones gráficas dentro de la memoria virtual sin necesidad de tener un entorno gráfico en el sistema operativo. Para obtener los paquetes de instalación de Google Chrome se usa el siguiente comando, dando como resultado lo que se evidencia en la figura 109:

```
wget -O google-chrome-stable_current_x86_64.rpm https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm
```

```
leiton@leiton:~$ wget -O google-chrome-stable_current_x86_64.rpm https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm
2020-06-07 17:27:56 -- https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm
Resolviendo dl.google.com (dl.google.com)... 172.217.173.46, 2090:3f0:4005:405:2090
Conectado con dl.google.com (dl.google.com) [172.217.173.46]:443... conectado.
Estado HTTP: 200, esperado respuesta... 200 OK
Resolviendo google-chrome-stable_current_x86_64.rpm
Guardado a: "google-chrome-stable_current_x86_64.rpm"
100%[=====>] 60.476.500 1,209B/s en 57s
2020-06-07 17:28:52 (3.22 MB/s) - "google-chrome-stable_current_x86_64.rpm" guardado [60476500-6047881]
leiton@leiton:~$
```

Figura 109. Instalación de paquetes de Chrome.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Luego se debe instalar los paquetes obtenidos de Google Chrome con el siguiente comando: **sudo yum install ./google-chrome-stable_current_x86_64.rpm**

Para la instalación del complemento Xvfb, es posible buscar paquetes dentro del servidor con el comando: **sudo yum search xvfb**. En caso de que ya existan, sólo será necesario instalarlo. La figura 110 muestra el resultado de la ejecución de la instrucción mencionada.

```
leiton@leiton:~$ sudo yum search xvfb
complementos cargados:fastestmirror
loading mirror speeds from cached hostfile
 * base: mirror.cedia.org.ec
 * epel: mirror.compu0.com
 * extras: mirror.cedia.org.ec
 * updates: mirror.cedia.org.ec
N/S switched: xvfb
python-xvfbwrapper search : run headless display inside X virtual framebuffer (Xvfb)
xorg-x11-server-Xvfb.x86_64 : 0 X Window System virtual framebuffer X server.
Nombre y resumen que coinciden con y sólo , use "buscar todo" para todo.
```

Figura 110. Búsqueda de paquetes dentro del servidor.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

De acuerdo al resultado arrojado, se debe instalar el paquete: **xorg-x11-server-Xvfb.x86_64**, utilizando la instrucción: **sudo yum install xorg-x11-server-Xvfb.x86_64**, obteniendo como resultado lo que se aprecia en la figura 111.

```

Instalado:
  sory-x11-server-Xvfb.x86_64 0:1.20.4-10.e17

Dependencias instaladas:
  libXau.x86_64 0:1.1.2-6.e17          libXfont2.x86_64 0:2.0.3-1.e17
  libXdmcp.x86_64 0:1.1.2-2.e17       libXt.x86_64 0:1.1.5-3.e17
  libXtst.x86_64 0:1.0.9-3.e17        sory-x11-server-common.x86_64 0:1.20.4-10.e17
  sory-x11-south.x86_64 1:1.0.9-1.e17  sory-x11-xkb-qt11c.x86_64 0:7.7-14.e17
  x11proto
  
```

Figura 111. Instalación de paquete xorg-x11-server-Xvfb.x86_64.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Instalación en Windows

Se usa la version de Katalon en Windows ya que, como se mencionó anteriormente, no se puede ejecutar la interfaz gráfica en el servidor. La versión en Windows se usa para crear las pruebas que posteriormente se configurarán dentro de Jenkins y finalmente son ejecutadas en el servidor CentOS 7. Además, según Katalon (s.f.), es importante tener en cuenta lo siguiente si se usa una versión inferior a la 7:

- No se podrán activar versiones inferiores a 7.
- No se actualizarán controladores.
- No habrá parches de seguridad y de rendimiento.
- No habrá mantenimiento futuro para la documentación, soporte o corrección de errores.

Por lo anterior, de la página oficial <https://www.katalon.com/download/> se realiza la descarga de una versión 7 o posterior (ver figura 112).



Figura 112. Instalación Katalon en Windows 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Esto baja un archivo comprimido, que debe ser extraído en una carpeta dentro del disco local C, donde se requiere ejecutar el archivo Katalon que se encuentra dentro de la carpeta. El paso a paso puede verse en las figuras 113 y 114, respectivamente.

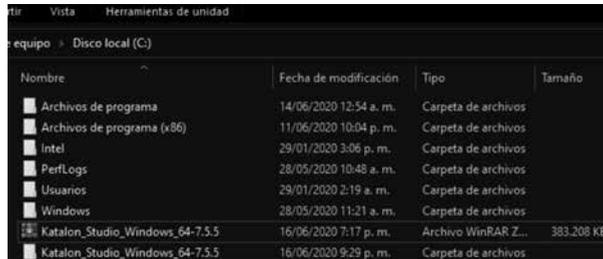


Figura 113. Instalación Katalon en Windows 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

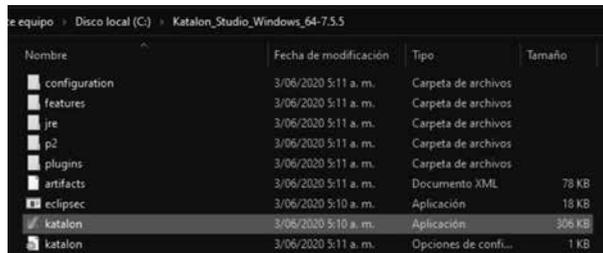


Figura 114. Instalación Katalon en Windows 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Creación de pruebas a modo de ejemplo

Esta sección busca explicar cómo unir Katalon y Git, ya que es una configuración general y puede variar muy poco, siendo útil para cualquier situación.

Al ejecutar por primera vez Katalon, saldrá una interfaz como la que se muestra en la figura 115, donde se debe ingresar el correo y contraseña que se registraron en la página oficial de Katalon (<https://www.katalon.com>), de esta forma se activará la aplicación para su uso.

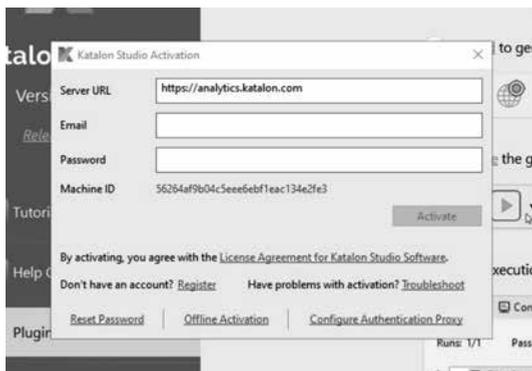


Figura 115. Login de Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Después de ejecutar Katalon, en el menú superior izquierdo se debe seleccionar la opción File y después, dentro del menú que despliega, dar clic en Project, como se muestra en la figura 116.

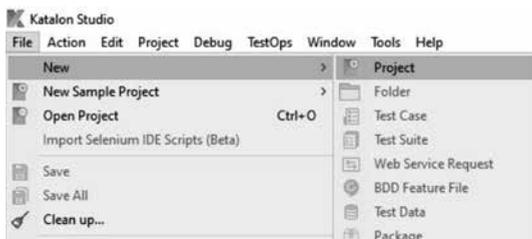


Figura 116. Nuevo proyecto en Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Aparecerá una ventana como la que se ilustra en la figura 117, donde se ingresa un nombre, tipo de proyecto, la plantilla que se usará, la URL del repositorio, la ubicación donde se guardará el proyecto y una descripción. El tipo de proyecto puede ser sólo Web o en combinación con una API, móvil o genérico. Para este ejemplo se selecciona el tipo Web y una plantilla en blanco (blank).

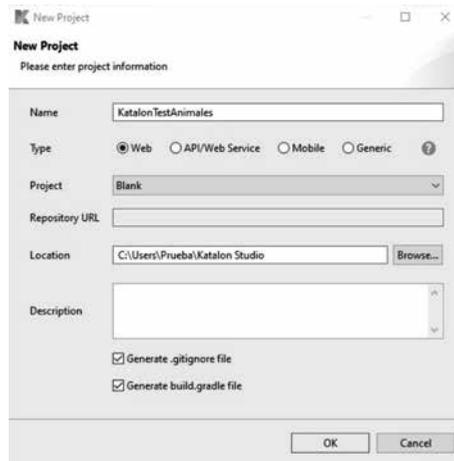


Figura 117. Configuración de nuevo proyecto en Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019)

En este caso se configura a Chrome como navegador por defecto para ejecutar las pruebas. Situarse en el menú superior izquierdo y desplegar las opciones de Project y dar clic en Settings; los pasos anteriores se muestran en la figura 118.

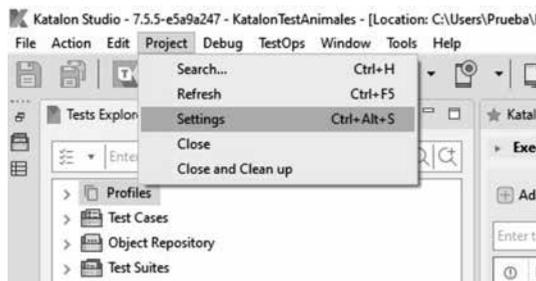


Figura 118. Navegador predeterminado Chrome en Katalon 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019)

Para configurar el navegador es necesario en *Execution* del menú que se encuentra al lado izquierdo, cambiar las opciones dentro de *Default execution*, donde se muestran los navegadores. Ahí se puede seleccionar Chrome o el que se desee. Aplicar los cambios dando clic en el OK. Ver figura 119.

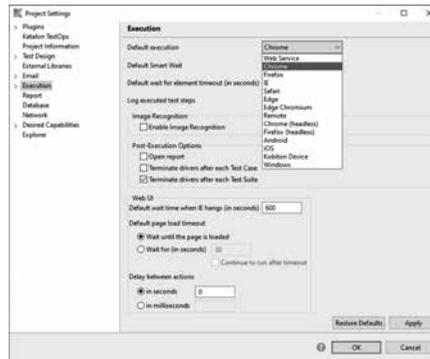


Figura 119. Navegador predeterminado Chrome en Katalon 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para este ejemplo, se crearon tres Test Case y un Test Suite. Un *Test Case* o, por su traducción al español, *caso de prueba*, consiste en la grabación del paso a paso para cumplir una función específica, por ejemplo, el registro de un nuevo usuario. Esto creará automáticamente un script completo del proceso, donde se graba el paso a paso requerido para la transacción, diligenciando cada campo y presionando la opción de envío. En cambio, el *Test Suite* o, por su traducción al español, *conjunto de pruebas*, se basa en la ejecución de varios Test Case previamente configurados, que cumplen con varias funcionalidades asociadas a un proceso y que pueden tener o no dependencia entre sus resultados, por lo que se requiere que sean ejecutados en un orden.

Para comenzar con la creación del *Test Case*, dirigirse al menú izquierdo y desplegar las opciones de **Test Case**, después en **New** y seleccionar **Test Case**, como se indica en la figura 120.

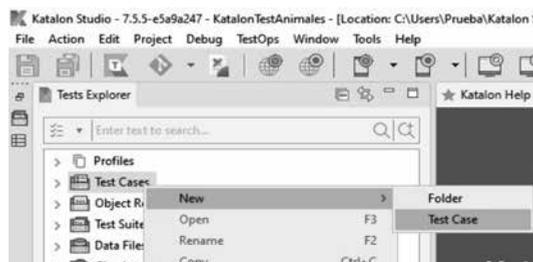


Figura 120. Configuración de Test Case en Katalon 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Esto desplegará una pantalla donde se ingresará el nombre y la descripción del caso de prueba, como muestra la figura 121. Se finaliza al dar clic en **OK**.

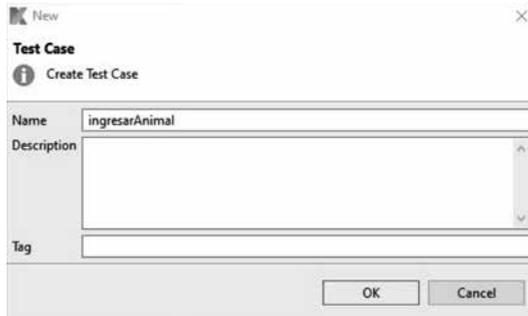


Figura 121. Configuración de Test Case en Katalon 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Hay tres formas de grabar los pasos. Forma manual (Manual Mode), Script (Script Mode) o por medio de la grabación y reproducción (Record & Play). El modo manual permite a los usuarios crear pruebas de automatización fácilmente con pocas habilidades de programación requeridas. El modo Script permite a los usuarios expertos escribir pruebas de automatización mediante programación en la vista Script de casos de prueba. El modo de grabación y reproducción consiste en grabar los pasos de una acción que se ejecuta en el navegador, después se creará un Script automáticamente para ser usado individualmente después o en un Test Case, haciéndolo más simple, para familiarizarse con la herramienta. En este caso se utiliza el modo de grabación y reproducción.

Para realizar una grabación, presionar el ícono en forma circular de color verde con un punto rojo (Record Web). Se recomienda ver la figura 122.

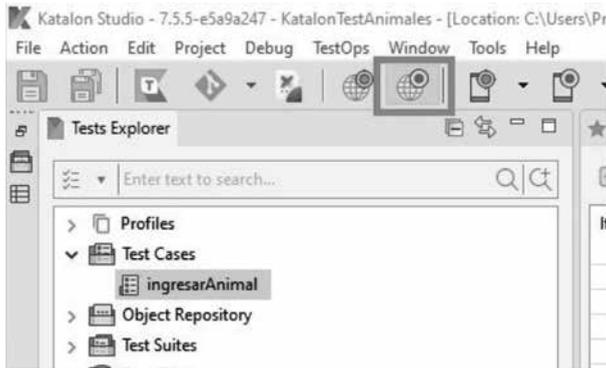


Figura 122. Configuración de Test Case en Katalon 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

A continuación, se desplegará una pantalla como en la figura 123, donde se ingresa la URL del aplicativo que se desea grabar y también se selecciona el navegador en el que se ejecutará el procedimiento.

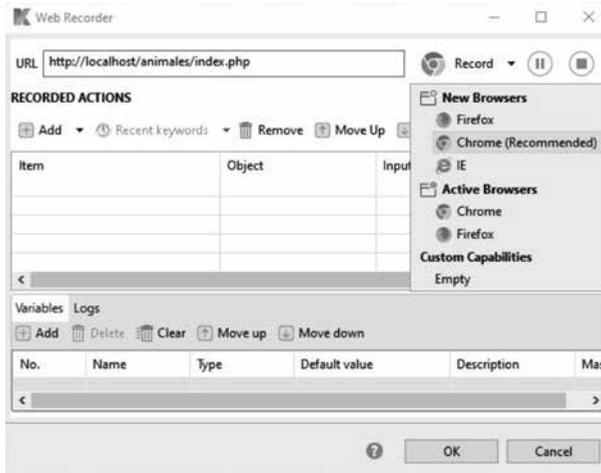


Figura 123. Configuración de Test Case en Katalon 4.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para empezar la ejecución debe de dar clic en Record (figura 124), que ejecutará automáticamente el navegador, abriendo la aplicación a probar.

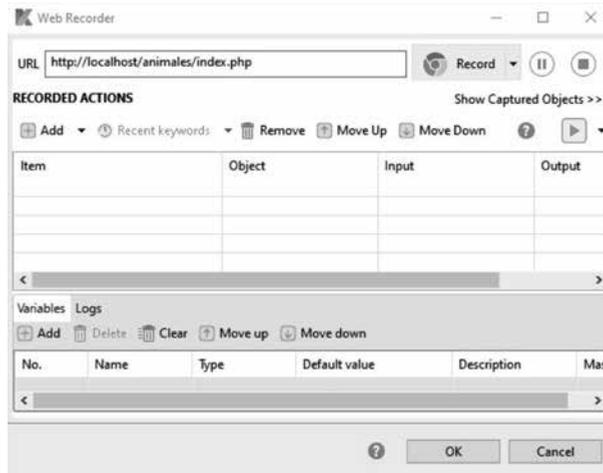


Figura 124. Configuración de Test Case en Katalon 5.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En este caso se trata de un proyecto simple, que fue codificado con PHP puro y se grabará la acción de registrar un ítem (Nombre Animal), como se observa en la figura 125.

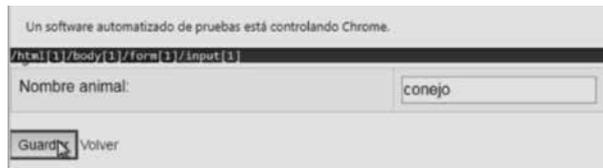


Figura 125. Configuración de Test Case en Katalon 7.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para terminar la grabación de las acciones se cierra el navegador y automáticamente se abrirá una ventana. Ver figura 126 en la que se mostrará las acciones grabadas.



Figura 126. Acciones guardadas del Test Case en Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Una vez presionado **OK**, se desplegará la opción **Add Element to Object Repository** que se ilustra en la figura 127. Aquí se mostrarán todas las acciones grabadas y que quedarán almacenadas en una carpeta llamada **Object Repository**. Es posible crear una subcarpeta para especificar que las acciones realizadas son del módulo de ingresar un nuevo animal si uno lo requiere.

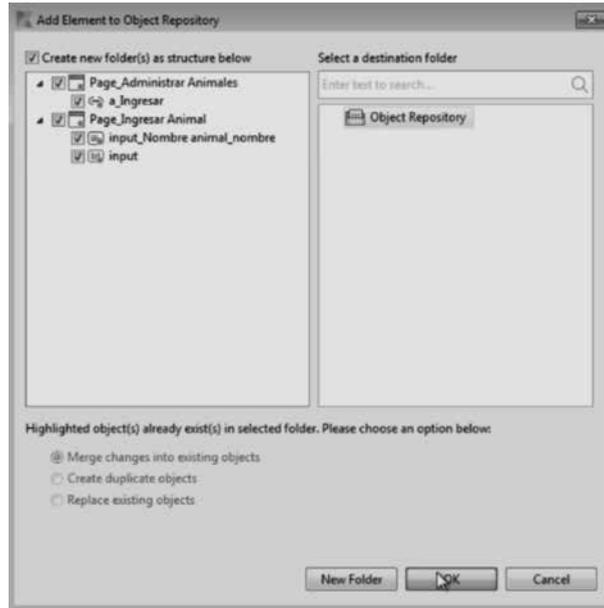


Figura 127. Repositorios de objetos en Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Al de dar clic en OK se mostrará el paso a paso grabado, como se ve en la figura 128.

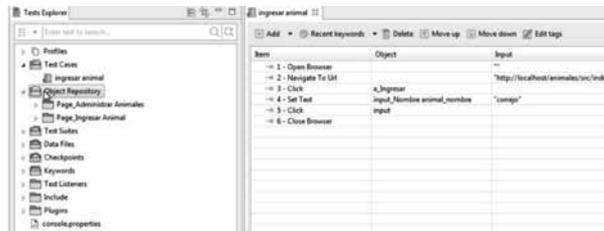


Figura 128. Interfaz principal de Katalon despues de añadir un Test Case.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para continuar con el ejemplo, dentro del Test Suite, se adicionarán tres Test Case. En cada uno se grabará las acciones de Actualizar Animal y Eliminar Animal, como se hizo con Ingresar Animal. El resultado final del Test Suite organizado quedará como se muestra en la figura 129.

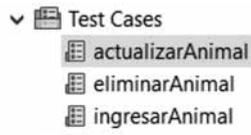


Figura 129. Creación de dos Test Case adicionales.

Fuente: (Leiton Muñoz y Moya Loaliza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para la configuración, en el menú izquierdo dar clic derecho sobre Test Suites para desplegar las opciones. Elegir **New** y, por último, **Test Suite**. Los pasos anteriores se ilustran en la figura 130

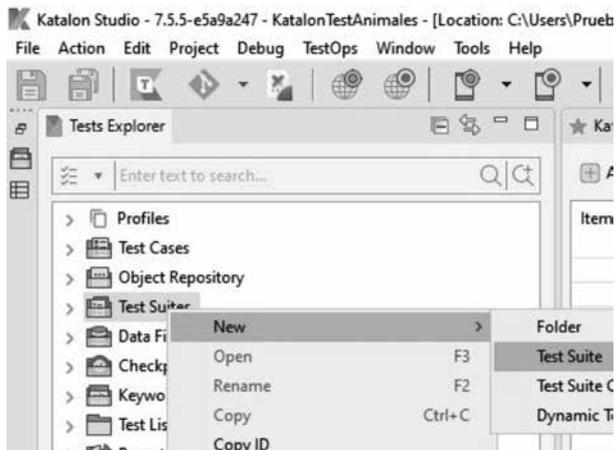


Figura 130. Creación de Test Suite en Katalon 1.

Fuente: (Leiton Muñoz y Moya Loaliza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Después aparecerá una interfaz similar a la que se muestra en la figura 131, donde se debe ingresar un nombre y una descripción.

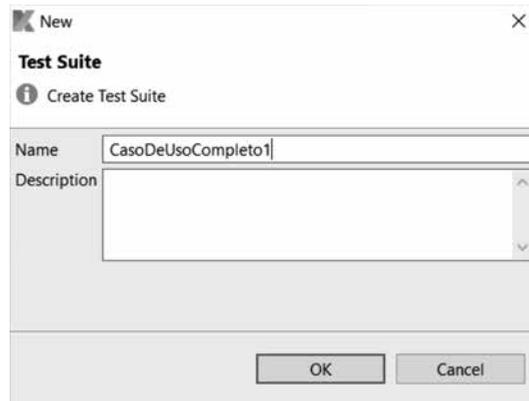


Figura 131. Creación de Test Suite en Katalon 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Lo anterior creará el Test Suite y se abrirá automáticamente una pestaña al lado derecho, en la cual se puede añadir los Test Cases creados anteriormente mediante la opción **Add**, ver figura 132.

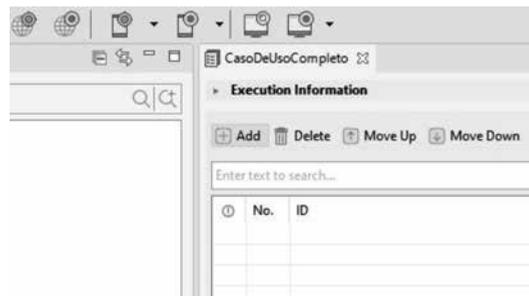


Figura 132. Creación de Test Suite en Katalon 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Es importante mencionar que al momento de añadir los Test Cases, se debe realizar en el orden en que se desea que se ejecuten. Para el ejemplo de guía, queda de la forma en que se muestra en la figura 133.

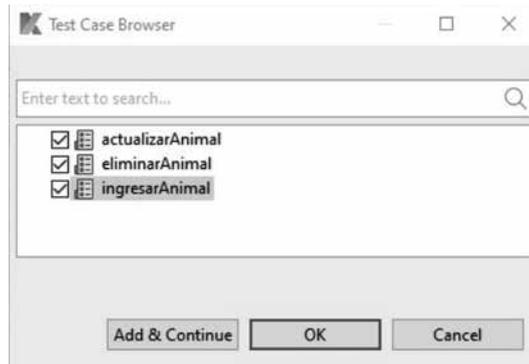


Figura 133. Creación de Test Suite en Katalon 4.

Fuente: (Leiton Muñoz Brayan Alexis Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Para ejecutar el Test Suite, se debe estar con el conjunto de pruebas abierto y seleccionado y dirigirse al menú de la parte superior central en la que se deben desplegar las opciones que se encuentran al lado derecho del logo de Play. Seleccionar el navegador en el que quiere ejecutar. Guiarse con la figura 134.



Figura 134. Ejecución de Test Suite en Katalon 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Automáticamente se abrirá una ventana del navegador seleccionado en donde se ejecutarán todas las acciones guardadas en el Test Suite, es decir, los Test Cases. El proceso de carga se muestra en la figura 135

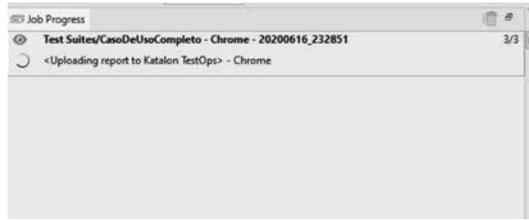


Figura 135. Ejecución de Test Suite en Katalon 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Los resultados se muestran mediante el log de ejecución, como indica la figura 136. Aquí se mostrarán cuáles Test Cases fueron ejecutados, el tiempo que tomó cada uno en ejecutarse y los resultados obtenidos.



Figura 136. Log de ejecución de Test Suite en Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Luego comprobada su ejecución de manera correcta, se procede a *exportar como comando CMD* para poder implementarlo en Jenkins. Para ello, se debe ubicar el menú superior central y dar clic en el logo que contiene lo siguiente: los caracteres **>_ (Build CMD)**, resaltado en un recuadro rojo de la figura 137 para hacer más comprensible la explicación.

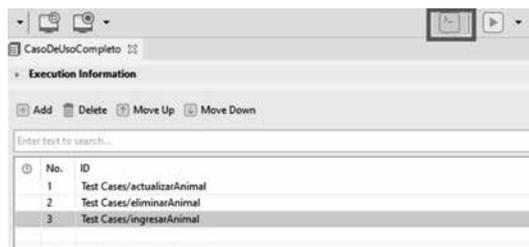


Figura 137. Comando CMD de Test Suite en Katalon 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Antes de continuar, para exportar el comando CMD es necesario generar una API Key de Katalon, que se podrá encontrar dentro del link <https://analytics.katalon.com/user/profile>, en donde se inicia sesión y después se ingresa a la pestaña de API Key para copiar el serial que se encuentra en el recuadro rojo, como se muestra en la figura 138.

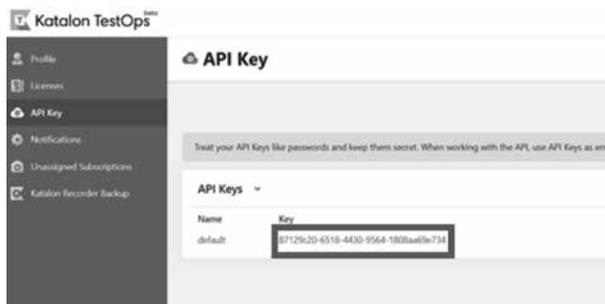


Figura 138. API Key de Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Dentro de la herramienta, se pega el **API Key** en el campo **Katalon API Key**, como se muestra en la figura 139.

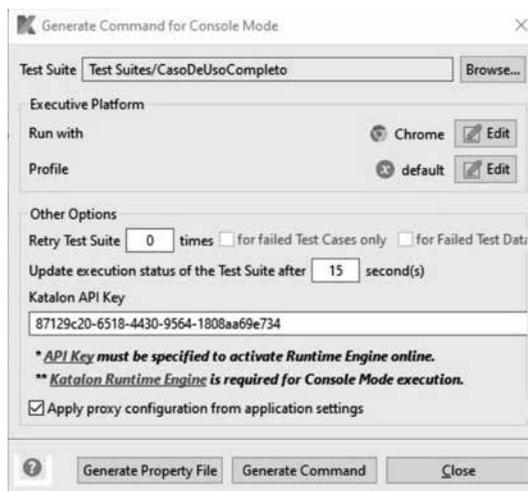


Figura 139. Comando CMD de Test Suite en Katalon 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Luego se debe seleccionar el Test Suite creado, por lo que dentro de la opción Browser se busca el que se ha creado anteriormente (ver figura 140).

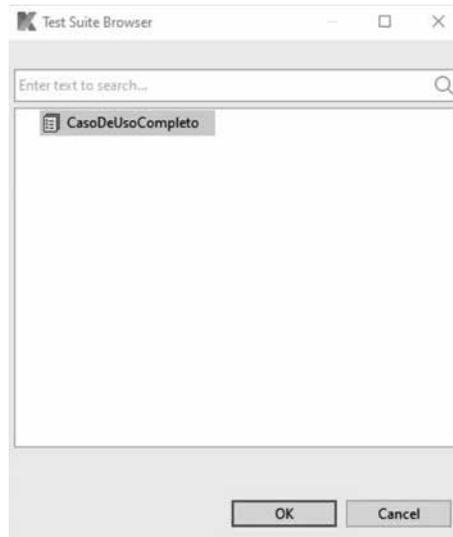


Figura 140. Comando CMD de Test Suite en Katalon 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Por último, dar clic en **Generate Command**, como muestra en la figura 141. Este comando deberá copiarse para ser implementado dentro de la configuración del Proyecto en Jenkins.

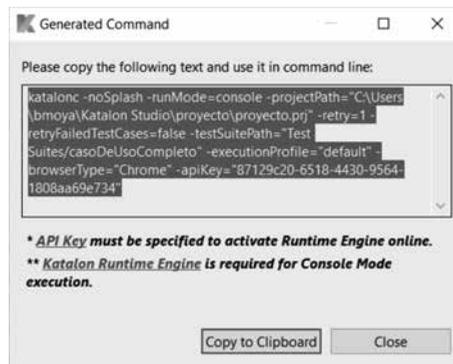


Figura 141. Comando CMD de Test Suite en Katalon 4.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Antes de comenzar con el proceso de las pruebas, es indispensable configurar Git para Katalon. Esto ayuda a versionar el conjunto de pruebas del proyecto, manteniendo el orden en todo momento. Por tanto, situarse en el menú superior izquierdo y dar clic en *Window* para desplegar las opciones (ver figura 142),

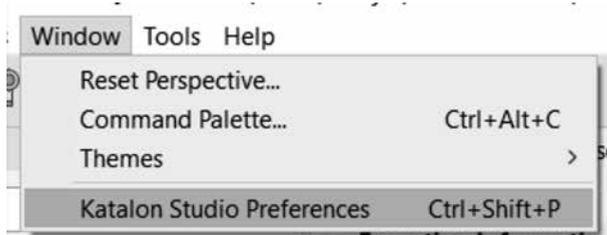


Figura 142. Configuración de Git en Katalon 1.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Después, se debe seleccionar *Katalon Studio Preferences* para abrir la interfaz que se muestra en la figura 143. Ahí seleccionar la opción *Git*, luego la opción *Enable Git Integration* y finalizar la configuración dando clic en *OK*.

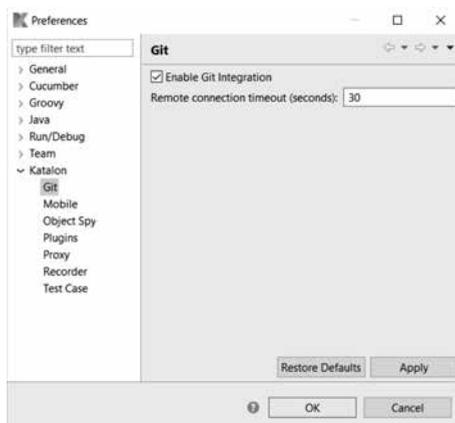


Figura 143. Configuración de Git en Katalon 2.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Al quedar configurado lo anterior, la herramienta habilitará un botón rojo con el logo de Git en la interfaz principal, ver figura 144. Este servirá para completar el resto de la configuración de Git.

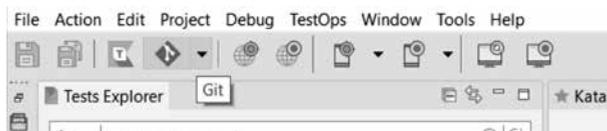


Figura 144. Configuración de Git en Katalon 3.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Desplegar las opciones con la fecha que se encuentra al lado derecho y dar clic en Share Project para iniciar Git, como se ilustra en la figura 145.

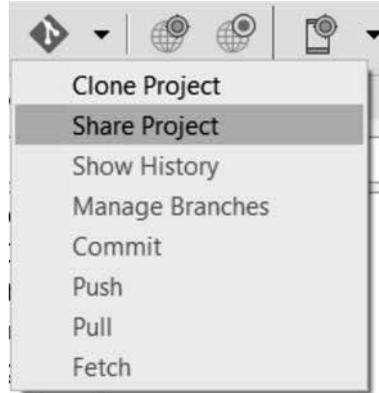


Figura 145. Configuración de Git en Katalon 4.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Si el resultado es exitoso saldrá un mensaje como el de la figura 146.



Figura 146. Configuración de Git en Katalon 5.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Se recomienda verificar que se haya creado la carpeta `.git` con el nombre del proyecto (`animalesKatalon`) en la carpeta Katalon Studio, que se encuentra en la ruta: `C:\Users\"usuario"\Katalon Studio`, como se ilustra en la figura 147 a modo de garantizar la correcta creación y vinculación del repositorio.

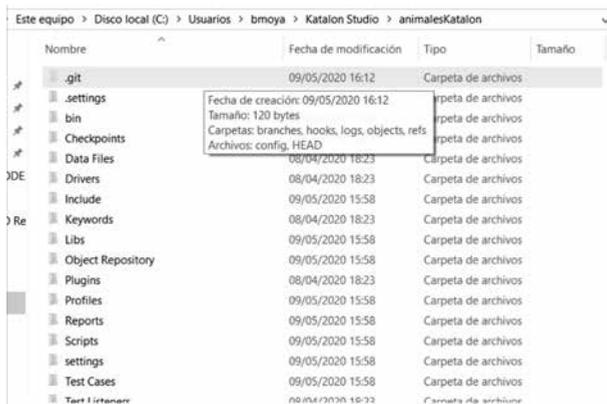


Figura 147. Configuración de Git en Katalon 6.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Una vez verificada la creación de la carpeta .git, se pasa a crear el repositorio directamente en Git (github.com), donde al final de la creación se proporciona una ruta de identificación de este repositorio creado, copiar ruta (ver figura 148) y tener en cuenta para el próximo paso.

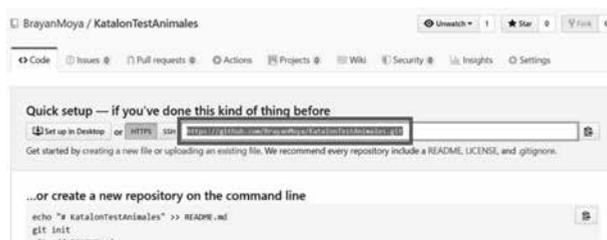


Figura 148. Configuración de Git en Katalon 7.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Luego, se es necesario hacer un Commit al Proyecto de Katalon para guardarlo en el repositorio creado en Git. Para tener más información acerca de las configuraciones en un proyecto, visitar este link oficial de Katalon: <https://docs.katalon.com/katalon-studio/docs/execution-settings.html#execution-settings>.

El proyecto creado anteriormente, como se encuentra en un repositorio de Git (ver figuras 149 y 150), se debe clonar dentro del servidor, para accederlo por parte de Jenkins. En este caso, se clonó dentro de la carpeta Workspace de Jenkins. Para hacerlo, se ingresa a la ruta donde se desea clonar y ejecutar

el comando *git clone* seguido de la ruta del repositorio, en este caso quedaría como se muestra en la figura 149:

```
leiton@leiton:~/Documents$ git clone https://github.com/leiton/katalon-test-framework.git
```

Figura 149. Ruta del repositorio.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019)

Los archivos descargados se muestran en la figura 150:

```
leiton@leiton:~/Documents$ ls -la
total 12
drwxr-xr-x  7 leiton leiton  78 Jun 14 12:49 bin
-rw-r--r--  1 ??? root  426 Jun 7 16:58 build.gradle
drwxr-xr-x  2 leiton leiton   6 Jun 7 19:41 Checkstyle
-rw-r--r--  1 ??? root  134 Jun 7 16:58 compile.properties
drwxr-xr-x  2 leiton leiton   6 Jun 7 19:41 data files
drwxr-xr-x  2 leiton leiton   6 Jun 7 19:41 database
-rw-r--r--  1 ??? root   51 Jun 7 19:41 README
drwxr-xr-x  2 leiton leiton   6 Jun 7 19:41 resources
drwxr-xr-x  3 leiton leiton  51 Jun 14 12:54 lib
-rw-r--r--  1 ??? root  89 Jun 11 21:17 Object Repository
drwxr-xr-x  2 leiton leiton   6 Jun 7 19:41 TestRunner
-rw-r--r--  1 ??? root   26 Jun 7 16:58 testfiles
drwxr-xr-x  1 ??? root 1171 Jun 7 28:16 proyecto.prj
drwxr-xr-x  7 leiton leiton 167 Jun 14 12:49 src
-rw-r--r--  1 ??? root   69 Jun 11 21:17 testplan
drwxr-xr-x  4 ??? root   38 Jun 7 16:58 settings
-rw-r--r--  1 ??? root  181 Jun 11 21:17 test cases
drwxr-xr-x  2 leiton leiton   6 Jun 7 19:41 TestRunner
-rw-r--r--  1 ??? root  186 Jun 11 21:17 test suite
```

Figura 150. Ruta y archivos de proyecto prueba.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019)

Instalación y configuración en Jenkins

Katalon es la herramienta seleccionada para la automatización de pruebas del ambiente de calidad preventiva. Se podrá utilizar por medio de la inclusión de *Katalon Studio Plugin* dentro de *Jenkins*. Así, cuando un cambio en el código fuente es realizado a nivel del repositorio, la herramienta Jenkins detecta este cambio y dispara todo un ciclo de calidad a través de las herramientas que tiene integradas en el proyecto a revisar. En este caso, dispararía IC por medio de Jenkins; revisaría las pruebas unitarias del aplicativo, ejecutaría una revisión estática de código por medio de SonarQube y luego ejecutaría las pruebas funcionales automatizadas mediante Katalon. Cabe resaltar que esta forma de instalación se tiene que hacer sólo una vez y estará disponible de manera global para cualquier proyecto.

Para realizar la configuración de Katalon dentro de Jenkins, se debe ingresar a la interfaz de *Administrar Plugins* dentro de *Jenkins*. Buscar *Katalon Studio Plugin*, seleccionar para instalar (ver figura 151) y con esto quedará habilitado.



Figura 151. Instalación de plugin Katalon en Jenkins.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

En caso de no contar con el plugin, seleccionar la opción *Todos los plugins*, ingresar el nombre, seleccionar e instalar.

Una vez realizado esto, se vuelve a la opción de *Configuración* del proyecto y se debe ubicar la sección de *Ejecutar*. En esta sección dar clic en *Añadir un nuevo paso*, seleccionar *Execute Katalon Studio Test*, como se ve en la figura 152. Esto servirá para poder correr el script de pruebas Katalon generado anteriormente:

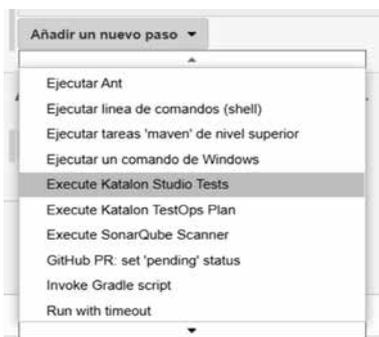


Figura 152. Paso nuevo para ejecutar pruebas de Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019

Posterior a esto, se debe seleccionar una versión de Katalon para instalar en el servidor CentOS en el campo *Download Katalon Studio Version* y, a continuación, pegar el comando generado de Katalon (CMD) (ver figura 163), pero ya configurado para CentOS. La variable *ProjectPath* que da el comando generado de Katalon es la que indica dónde está ubicado el archivo .prj a ejecutar, la ruta se cambiará y se colocará la que apunta a este mismo archivo del proyecto de Katalon, anteriormente clonado del repositorio. El código completo deberá quedar de la siguiente manera:

```
-noSplash --config -webui.autoUpdateDrivers=true  
-sendMail="bmoya17@hotmail.com" -projectPath="/var/lib/  
jenkins/workspace/prueba java/KatalonTestAnimales/proyecto.
```

```
prj" -retry=0 -retryFailedTestCases=false -testSuitePath="Test Suites/  
katalon" -executionProfile="default" -browserType="Chrome"  
-apiKey="87129c20-6518-4430-9564-1808aa69e734"
```

Se pegará en el campo Command arguments⁷, como se detalla en la figura 153.



Figura 153. Código Test de Katalon.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020) some recent research related to the topics covered in this work is exposed, as well as some tools that comply with the good development practices proposed by (Pastrana et al., 2019)

La ruta donde queda instalado Katalon en el servidor CentOS 7 es la siguiente: `/var/lib/jenkins/.katalon/6.3.3/Katalon_Studio_Engine_Linux_64-6.3.3`

Luego, en la sección *Acciones para ejecutar*, se añade la acción de *Publicar los resultados de Tests JUnit*. En el campo *Ficheros XML con los informes de tests*, se ingresa la ruta en donde se encuentran los archivos .xml, que están ubicados en este caso en la siguiente dirección: `KatalonTestAnimales/Reports/casoDeUsoCompleto/*/*.xml`⁸. Quedaría de la siguiente manera, ver figura 154:

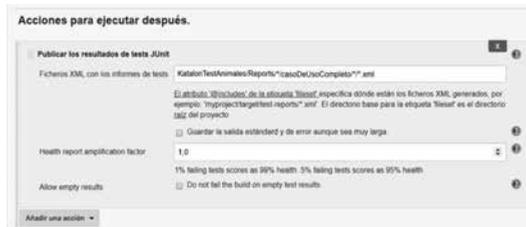


Figura 154. Configuraciones JUnit.

Fuente: (Leiton Muñoz y Moya Loaiza, 2020)

⁷ Se recomienda que la versión a instalar sea inferior a la 7.0, ya que las versiones superiores presentan fallos de activación al ejecutar las pruebas en Jenkins, caso que no sucede en versiones inferiores.
⁸ Con los * se indica que ingrese a todos los archivos de la carpeta padre, con esto se verán todos los reportes que deje la ejecución del proyecto.

Proyectos realizados por Smart Campus

Para la fase I del proyecto Ecosistema Smart Campus se realizaron satisfactoriamente un total de 9 trabajos de grado, de los cuales el 56% (5 trabajos) se enfocaron en el eje de Smart Government y el restante 44% (4 trabajos) fueron de Smart People, como muestra la figura 155.

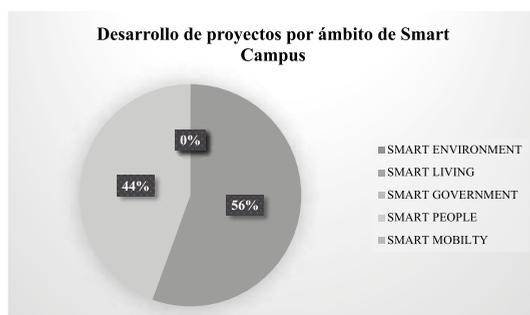


Figura 155. Porcentaje de proyectos ejecutados por eje Fase I.

Autora: Ana Milena Rojas Calero. Seguimiento de proyectos Smart Campus fase I

Así mismo, en la figura 156 se destaca la producción realizada por los investigadores del proyecto en dicha fase. Teniendo presente que el profesor Manuel Alejandro Pastrana participa con una codirección con el profesor Leandro Flórez Aristizábal.

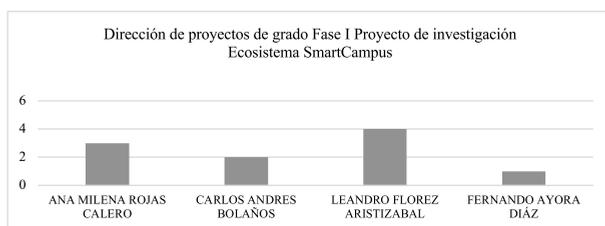


Figura 156. Proyectos dirigidos por investigador.

Autora: Ana Milena Rojas Calero. Seguimiento de proyectos Smart Campus fase I

A continuación, se da un detalle puntual sobre los proyectos realizados.

FASE I

Smart People - Ciudadanos

Trabajo de grado: Sistema de Información para plataforma IOS de eventos académicos y financieros de la Institución Universitaria Antonio José Camacho.

Estudiante(s): Cristian Fabián Barco Volveras

Director: Leandro Flórez Aristizábal

Datos de contacto director: learistizabal@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2017

Programa: Ingeniería de Sistemas (Diurno)

Trabajo de grado: Sistema de Información para la integración de componentes de software que permita la implementación de un módulo que integre la funcionalidad de control de seguridad y consulta de notas para su acceso desde un aplicativo móvil dirigido a los estudiantes de la Institución Universitaria Antonio José Camacho

Estudiante(s): Fabián Esteban Capote Casas

Director: Leandro Flórez Aristizábal

Datos de contacto director: learistizabal@admon.uniajc.edu.co

Periodo finalización: Primer semestre del 2018

Programa: Ingeniería de Sistemas (Nocturno)

Trabajo de grado: Laboratorio Remoto de Redes

Estudiante(s): Lizeth Valentina Burbano Rodríguez-Jessica Yuliana Moreno Guerrero

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Primer semestre del 2018

Programa: Ingeniería de Sistemas (Diurno)

Distinción: Trabajo de grado meritório

Trabajo de grado: Sistema para la Gestión de Comunicación de Carácter Académico entre los Estudiantes y Docentes en la Institución Universitaria Antonio José Camacho

Estudiante(s): Fernando Andrés Cifuentes Calderón - Nasly Katerine Escobar

Director: Leandro Flórez Aristizabal – Manuel Alejandro Pastrana Pardo

Datos de contacto director: learistizabal@admon.uniajc.edu.co

Periodo finalización: Primer semestre del 2018

Programa: Ingeniería de Sistemas (Diurno)

Distinción: Trabajo de grado meritório

Smart Government - Administración y gobierno

Trabajo de grado: Implementación de un sistema de Gestión Documental electrónico de archivo en la Institución Universitaria Antonio José Camacho

Estudiante(s): Paula Andrea Cañas Josa

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2018

Programa: Ingeniería de Sistemas (Diurno)

Distinción: Trabajo de grado meritório

Trabajo de grado: Desarrollo de Plataforma Web para la creación de formularios de encuestas para la Institución Universitaria Antonio José Camacho

Estudiante(s): Vanessa Rodríguez Galvis - Diego Armando Cortés Valencia

Director: Fernando Ayora Díaz

Datos de contacto director: fayora@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2018

Programa: Tecnología en Sistemas (Nocturno)

Trabajo de grado: Sistema de información web para la gestión de contratos en la UNIAJC - Fase I

Estudiante(s): Juan Pablo Dorado Mesa - Alexander Castaño Ramírez

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2018

Programa: Ingeniería de Sistemas (Diurno)

Trabajo de grado: Software para la reserva de salas y equipos de cómputo para la Institución Universitaria Antonio José Camacho Fase I

Estudiante(s): Kevin Fabián Rodríguez Collazos - Héctor Fabio Rodríguez Gaviria

Director: Carlos Andrés Bolaños Ceballos

Datos de contacto director: cabo@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2018

Programa: Ingeniería de Sistemas (Diurno)

Trabajo de grado: Sistema para la gestión de proyectos Smart Campus

Estudiante(s): Rubén Díaz Pulli - Manuela Cardona

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2018

Programa: Ingeniería de Sistemas (Diurno)

Distinción: Trabajo de grado meritório

FASE II

Para la fase II del proyecto Ecosistema Smart Campus se realizaron satisfactoriamente un total de 21 trabajos de grado, de los cuales el 43% (9 trabajos) se enfocaron en el eje de Smart Government, el 23% (5 trabajos) a Smart Environment, el 19% (4 trabajos) fueron de Smart People, el 9% a Smart Living (2 trabajos) y el restante 5% (1 trabajo) a Smart Mobility, evidenciando un crecimiento de alcance en los ejes trabajados dentro del macro proyecto, como muestra la figura 157.

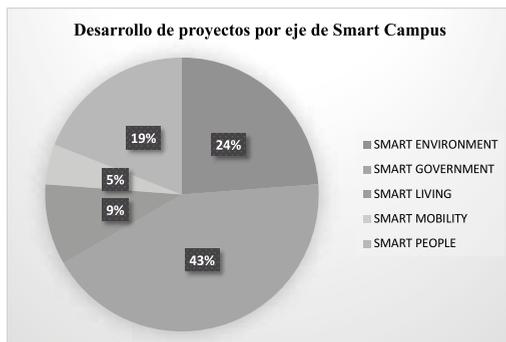


Figura 157. Porcentaje de proyectos ejecutados por eje Fase I.

Autora: Ana Milena Rojas Calero. Seguimiento de proyectos Smart Campus fase II

Así mismo, en la figura 158 se destaca la producción realizada por los investigadores del proyecto en dicha fase.

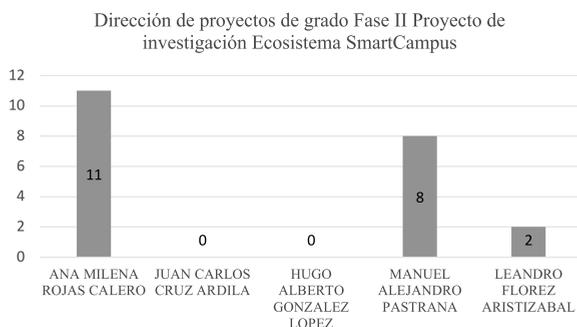


Figura 158. Proyectos dirigidos por investigador.

Autora: Ana Milena Rojas Calero. Seguimiento de proyectos Smart Campus fase II

A continuación, se da un detalle puntual sobre los proyectos realizados.

Smart People - Ciudadanos

Trabajo de grado: Aplicación que permite consultar notas de materias para estudiantes de la UNIAJC. (Actualización Android)

Estudiante(s): Fabio Capote

Director: Leandro Flórez Aristizabal

Datos de contacto director: learistizabal@admon.uniajc.edu.co

Periodo finalización: Programa: Ingeniería de Sistemas (Nocturno)

Trabajo de grado: Diseño y desarrollo de un sistema de información sobre dispositivos móviles para diagnósticos y quices en clases

Estudiante(s): Jorge Alexander Ortiz Cortés

Director: Leandro Flórez Aristizábal

Datos de contacto director: learistizabal@admon.uniajc.edu.co

Periodo finalización: Programa: Ingeniería de Sistemas (Nocturno)

Trabajo de grado: Sistema para la Gestión de Eventos Institucionales

Estudiante(s): Jaime Esteban Marulanda Vélez - Sebastián Ramírez Grisales

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: Primer semestre del 2020

Programa: Ingeniería de Sistemas (Nocturno)

Distinción: Trabajo de grado meritorio

Trabajo de grado: Actualización del software para la expedición de certificados y constancias dentro de la Institución Universitaria Antonio José Camacho

Estudiante(s): Raúl Andrés Hernández Ocampo - Stiven Castro Arias

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2020

Programa: Ingeniería de Sistemas (Nocturno)

Smart Government – Administración y gobierno

Trabajo de grado: Desarrollo de aplicación móvil de consulta y módulo de integración con el sistema „Academusoft“ para el software de reserva de salas y equipos para la „UNIAJC“ Fase II

Estudiante(s): Kevin Fabián Rodríguez Collazos - Héctor Fabio González Gaviria

Directores: Carlos Andrés Bolaños Ceballos- Ana Milena Rojas Calero

Datos de contacto directores: cabo@admon.uniajc.edu.co amrojas@admon.uniajc.edu.co

Periodo finalización: Programa: Ingeniería de Sistemas (Nocturno)

Trabajo de grado: Implementación de un dispositivo de acceso inteligente mediante tarjetas RFID y sensor biométrico para huella dactilar en la Institución Universitaria Antonio José Camacho

Estudiante(s): Luis Eduardo Hurtado Grijalba

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Programa: Ingeniería Electrónica (Nocturno)

Distinción: Trabajo de grado meritorio

Trabajo de grado: Aplicación web para gestión del ascenso del escalafón de Docentes de la UNIAJC

Estudiante(s): Cristian Gómez - Erick Pulido Neira

Director: Carlos Andrés Bolaños Ceballos- Ana Milena Rojas Calero

Datos de contacto director: cabo@admon.uniajc.edu.co amrojas@admon.uniajc.edu.co

Periodo finalización: Programa: Ingeniería de Sistemas (Nocturno)

Trabajo de grado: Diseño e implementación de solución analítica B.I. para la Gestión de Indicadores Académicos de la Institución Universitaria Antonio José Camacho.

Estudiante(s): Jhoan Sebastián Giraldo Mosquera - Juan David Ortiz Guevara

Director: Ana Milena Rojas Calero - Carlos Andrés Bolaños Ceballos

Datos de contacto director: amrojas@admon.uniajc.edu.co cabo@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2020

Programa: Ingeniería de Sistemas (Nocturno)

Distinción: Trabajo de grado meritorio

Trabajo de grado: Sistema para la gestión de solicitudes a la Facultad de Ingenierías de la Institución Universitaria Antonio José Camacho

Estudiante(s): Diego Andrés Gutiérrez Díaz - Gilberto Delgado Velasco

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: Primer Semestre 2020

Programa: Ingeniería de Sistemas (Nocturno)

Distinción: Trabajo de grado meritorio

Trabajo de grado: Sistema de gestión del índice de deserción en la UNIAJC.

Estudiante(s): María Camila Silva Moreno

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: pendiente por entregar

Programa: Ingeniería de Sistemas (Nocturno)

Trabajo de grado: Adaptación de herramienta para visualizar el control y monitoreo de los recursos de la red de la UNIAJC

Estudiante(s): Gustavo Borja

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Pendiente por entregar

Programa: Ingeniería de Sistemas (Diurno)

Smart Enviroment - Entorno y eficiencia

Trabajo de grado: Implementación de un dispositivo de monitoreo para un sistema inteligente con sensores distribuidos para las variables de temperatura y humedad el NOC de la UNIAJC

Estudiante(s): Joseph Caicedo Ambuila - Andrés Mauricio Rodríguez Bermúdez - Juan Felipe Collazos Prieto

Director: Ana Milena Rojas - Mario German Domínguez

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Pendiente por entregar

Programa: Tecnología en Electrónica Industrial

Trabajo de grado: Desarrollo de la aplicación móvil para la Unidad Municipal de Asistencia Técnica Agropecuaria (UMATA) en el Municipio de Versalles-Valle

Estudiante(s): Carlos Alberto Valencia Cocomá - Ronal Andrés Tamayo Zapata

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: Primer semestre del 2020

Programa: Ingeniería de Sistemas (Nocturno)

Trabajo de grado: Documentación e implementación de buenas prácticas DevOps con herramientas de automatización de pruebas y de despliegue a través del refinamiento de software de Smart Campus

Estudiante(s): William Andrés Leiton Muñoz - Brayan Alexis Moya Loaiza

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: Segundo semestre del 2017

Programa: Ingeniería de Sistemas (Diurno)

Distinción: Trabajo de grado meritorio

Trabajo de grado: Diseño de un dispositivo de medición inteligente para la monitorización del consumo energético en una sala de sistema de la UNIAJC

Estudiante(s): Diego Alejandro López García - Jhonatan Jair Mina Carabalí - Sergio Manolo Ordóñez

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización: Pendiente por entregar

Programa: Ingeniería Electrónica (Diurno)

Trabajo de grado: Prototipo de un punto ecológico inteligente para fomentar la cultura ambiental en las instalaciones de la UNIAJC

Estudiante(s): Andrés Camilo Angulo Correa

Director: Ana Milena Rojas Calero – Erika Sarria

Datos de contacto director: amrojas@admon.uniajc.edu.co esarria@admon.uniajc.edu.co

Periodo finalización: Segundo semestre 2008

Programa: Ingeniería Electrónica (Diurno)

Smart Mobility - Movilidad

Trabajo de grado: Diseño e implementación de la APP Green Car UNIAJC, como estrategia de movilidad inteligente dirigida a la comunidad académica de la UNIAJC

Estudiante(s): Luis Alberto Mora Montoya - José Roberto Granada Chacón

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: Para sustentar 2021 primer semestre

Programa: Ingeniería de Sistemas (Nocturno)

Distinción:

Smart Living - Estilo de vida

Trabajo de grado: Aplicación Móvil Cafetería Virtual UNIAJC

Estudiante(s): Nicolás Ortiz - Nicolás Bermúdez

Director: Manuel Alejandro Pastrana Pardo

Datos de contacto director: mapastrana@admon.uniajc.edu.co

Periodo finalización: Primer periodo 2020

Programa: Ingeniería de Sistemas (Nocturno)

Distinción: Ninguna

Trabajo de grado: Desarrollo de un sistema de información web para la gestión de historias clínicas realizadas por el programa de salud de Bienestar Universitario de la UNIAJC

Estudiante(s): Lina Vanessa López Castañeda

Director: Ana Milena Rojas Calero

Datos de contacto director: amrojas@admon.uniajc.edu.co

Periodo finalización:

Programa: Ingeniería de Sistemas (Nocturno)

Distinción: Ninguna

Publicaciones en las que participó el equipo de investigadores

A continuación, se hace mención a las publicaciones obtenidas por los investigadores del proyecto:

Producción bibliográfica - Artículo - Publicado en revista especializada

Autores: Fabián Esteban Capote, Leandro Floréz Aristizábal, Ana Milena Rojas Calero, Carlos Andrés Bolaños Ceballos, Sandra Cano, César A. Collazos.

Título: "Development of a SOA Platform Support the Integration of Software Components Based On Mobile Devices for a Smart Campus".

En: Cali, Colombia.

Advances in Computing. CCC 2017. Communications in Computer and Information Science, vol. 735. Springer, Cham. https://doi.org/10.1007/978-3-319-66562-7_48

Palabras clave: Smart Campus Service-Oriented, Architecture Android, Web service Database Academusoft

Producción bibliográfica - Artículo - Publicado en revista especializada

Autores: Hugo Ordonez Eraso, Manuel Alejandro Pastrana Pardo, Carlos Alberto Cobos Lozada.

Título: "ISO 29110 en Colombia: de la teoría a la práctica".

En: Colombia

Revista: Guillermo De Ockham ISSN: 1794-192X ed: UNIVERSIDAD DE SAN BUENAVENTURA v.18 fasc.1 p.1 - 15, 2020, DOI: 10.21500/22563202.4299

Palabras clave: ISO 29110, buenas prácticas, Desarrollo de software,

Producción bibliográfica - Artículo - Publicado en revista especializada

Autores: Manuel Alejandro Pastrana Pardo, Hugo Ordonez Eraso, Ana Milena Rojas Calero.

Título: "Ensuring Compliance with Sprint Requirements in SCRUM".

En: Singapur Advances in Intelligent Systems and Computing ISSN: 2194-5357
ed: Springer Science + Business Media v.924 fasc. p.33 - 45, 2019, DOI:
10.1007/978-981-13-6861-5_3

Palabras clave: Scrum, Procesos Para Desarrollo de Software, Software Quality Assurance-SQA.

Producción bibliográfica - Artículo - Publicado en revista especializada

Autores: Manuel Alejandro Pastrana Pardo, Hugo Ordonez Eraso, Fernando Andrés Cifuentes Calderón.

Título: Arquitectura de Integración para Servicios y Soluciones Smart Campus

Revista: INGCUC ISSN: 0122-6517 ed: IngCUC 16. fasc.2, 2020, DOI:
10.17981/ingecuc.16.2.2020.21

Palabras clave: ESB, Arquitectura de Integración, Arquitectura orientada a servicios, SOA.

Producción bibliográfica - Artículo - Publicado en revista especializada

Autores: Andrés Camilo Angulo Correa, Erika Sarria Navarro

Tipo: Capítulo de libro.

Título: Prototipo de un punto ecológico inteligente para fomentar la cultura ambiental en las instalaciones de la UNIAJC Cali, Colombia, septiembre, 2018.

En: La Investigación Formativa, ISBN:978-958-8292-84-7, pp. 222 - 226, 2020

Palabras clave: Reciclaje Inteligente, Reconocimiento Automático de Voz

Capítulos de libro

Producción bibliográfica - tipo capítulo de libro

Autores: Manuel Alejandro Pastrana Pardo, Hugo Ordonez Eraso.

Tipo: Capítulo de libro.

Título: "A model approach to ISO-29110 for very small enterprises".

Proceedings of the XXII Iberoamerican Conference on Software Engineering, ClbSE 2019, La Habana, Cuba, April 22-26, 2019. Curran Associates 2019.

En: Colombia. ISBN: 978-1-5108-8795-4 ed: Ibero-American Conference on Software Engineering, v, pp. 517-530.

Producción bibliográfica - tipo capítulo de libro

Autores: Manuel Alejandro Pastrana Pardo, Hugo Ordonez Eraso.

Tipo: Capítulo de libro.

Título: "La Elicitación de Requisitos mediante la técnica Inception Deck en las

empresas de Santiago de Cali”.

Proceedings of XXIV Congreso Colombiano de Computación, 14CCC, Pasto, Colombia, septiembre 25-27, 2019.

En: Colombia. ISBN: pendiente. Ed.: UNICESMAG, 2020

Producción bibliográfica - tipo capítulo de libro

Autores: Manuel Alejandro Pastrana Pardo, Hugo Ordonez Erasó.

Tipo: Capítulo de libro.

Título: “Inception Deck un camino hacia el desarrollo de Sof Skills requeridas por el sector TI en Colombia”.

Proceedings of XXIV Congreso Colombiano de Computación, 14CCC, Pasto, Colombia, septiembre 25-27, 2019.

En: Colombia ISBN: pendiente. Ed.: UNICESMAG, 2020

Producción bibliográfica - Otro artículo publicado - Revista de divulgación

Autores: Ana Milena Rojas Calero, Manuel Alejandro Pastrana Pardo.

Título: “Desarrollo de un sistema de información para la comercialización de cupones de alta segmentación en facturas de servicios públicos”.

En: Colombia. 2018. Actitud. ISSN: 1909-8510 pp.5-14 v.15

Palabras clave: Desarrollo de software, Arquitectura de software,

Áreas: Ingeniería y Tecnología, Ingenierías Eléctrica, Electrónica e Informática, Ingeniería de Sistemas y Comunicaciones.

Producción bibliográfica - Otro artículo publicado - Periódico de noticias

Autores: Ana Milena Rojas Calero, Manuel Alejandro Pastrana Pardo.

Título: “Patrones de diseño de software para modelos arquitecturales en smart campus basado en inception deck e ingeniería kaisen”.

En: Colombia. 2018. Sapiencia. ISSN: 0286-2204 pp.52-64 v.12

Palabras clave: Arquitectura de software, Desarrollo de software, smart campus, ingeniería kaisen,

Áreas: Ingeniería y Tecnología, Ingenierías Eléctrica, Electrónica e Informática, Ingeniería de Sistemas y Comunicaciones.

Producción bibliográfica - Otro artículo publicado - Periódico de noticias

Autores: Manuel Alejandro Pastrana Pardo, Ana Milena Rojas Calero.

Título: “Smart Living: cafetería virtual para la institución universitaria Antonio José Camacho”.

En: Colombia. 2018. Sapienía. ISSN: 0286-2204, pp.52-64 v.2

Palabras: Smart Campus, Smart Living, Desarrollo web, arquitectura de software

Áreas: Ingeniería y Tecnología, Ingenierías Eléctrica, Electrónica e Informática, Ingeniería de Sistemas y Comunicaciones.

Eventos científicos en los que participo el equipo de investigadores Registros de software

Nombre del evento: 12 Congreso Colombiano de Computación

Tipo de evento: Congreso

Ámbito: Internacional

Realizado el: 2017-09-20, 2019-09-22 en CALI - Universidad Autónoma de Colombia.

Productos asociados

- **Nombre del producto:** Development of a SOA Platform Support the Integration of Software Components Based On Mobile Devices for a Smart Campus
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** FABIÁN ESTEBAN CAPOTE CASAS
- **Rol en el evento:** Ponente

Nombre del evento: 14 Congreso Colombiano de Computación

Tipo de evento: Congreso

Ámbito: Internacional

Realizado el: 2019-09-25, 2019-09-27 en PASTO - Universidad Mariana

Productos asociados

- **Nombre del producto:** Inception Deck un camino hacia el desarrollo de Softskills para el sector TI Colombia
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia
- **Nombre del producto:** La Elicitación de Requisitos mediante la técnica Inception Deck en las empresas de Santiago de Cali

- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** MANUEL ALEJANDRO PASTRANA PARDO
- **Rol en el evento:** Ponente

Nombre del evento: IX ENCUENTRO SOFTHARD 2019

Tipo de evento: Encuentro

Ámbito: Nacional

Realizado el: 2019-05-02, 2019-05-02 en CALI - Institución Universitaria Antonio José Camacho

Participantes

- **Nombre:** MANUEL ALEJANDRO PASTRANA PARDO
- **Rol en el evento:** Ponente

Nombre del evento: IX TECNOTIC 2019

Tipo de evento: Encuentro

Ámbito: Nacional **Realizado el:** 2019-09-26, 2019-09-26 en CALI - Institución Universitaria Antonio José Camacho

Productos asociados

- **Nombre del producto:** Desarrollo de una aplicación (UMATA) multiplataforma en Ionic basado en microservicios.
- **Tipo de producto:** Producción técnica - Presentación de trabajo – Ponencia

Participantes

- **Nombre:** MANUEL ALEJANDRO PASTRANA PARDO, **Rol en el evento:** Ponente
- **Nombre:** RONAL ANDRES TAMAYO ZAPATA,
- **Rol en el evento:** Ponente
- **Nombre:** CARLOS ALBERTO VALENCIA COCOMA
- **Rol en el evento:** Ponente

Nombre del evento: IC4S2018- INTERNATIONAL CONFERENCE ON COMPUTER, COMMUNICATION AND COMPUTATIONAL SCIENCES

Tipo de evento: Otro

Ámbito: Internacional

Realizado el: 2018-10-20, 2018-10-21 en Bangkok - MANDARIN HOTEL BANGKOK, BANGKOK, THAILAND

Productos asociados

- **Nombre del producto:** Ensuring compliance with sprint requirements in SCRUM Preventive quality assurance in SCRUM
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** MANUEL ALEJANDRO PASTRANA PARDO Rol en el evento: Ponente

Nombre del evento: ENCUENTRO CIENTÍFICO Y EDUCATIVO DE EXPERIENCIAS INVESTIGATIVAS

Tipo de evento: Encuentro

Ámbito: Nacional

Realizado el: 2016-09-29, 2016-10-01 en CALL - Institución Universitaria Antonio José Camacho

Productos asociados

- **Nombre del producto:** Ecosistema Smart Campus UNIAJC.
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** ANA MILENA ROJAS CALERO
- **Rol en el evento:** Ponente

Nombre del evento: 2do ENCUENTRO CIENTÍFICO Y EDUCATIVO DE EXPERIENCIAS INVESTIGATIVAS

Tipo de evento: Encuentro

Ámbito: Nacional

Realizado el: 2017-10-07, 2017-10-08 en CALI - Institución Universitaria Antonio José Camacho

Productos asociados

- **Nombre del producto:** Prototipo de Laboratorio Remoto de Redes.
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** YESICA YULIANA MORENO GUERRERO
- **Rol en el evento:** Ponente
- **Nombre:** LISETH VALENTINA BURBANO RODRÍGUEZ
- **Rol en el evento:** Ponente
- **Nombre:** ANA MILENA ROJAS CALERO
- **Rol en el evento:** Ponente

Nombre del evento: 2do ENCUENTRO CIENTÍFICO Y EDUCATIVO DE EXPERIENCIAS INVESTIGATIVAS

Tipo de evento: Encuentro

Ámbito: Nacional

Realizado el: 2017-10-07, 2017-10-08 en CALI - Institución Universitaria Antonio José Camacho

Productos asociados

- **Nombre del producto:** Prototipo de Laboratorio Remoto de Redes.
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** YESICA YULIANA MORENO GUERRERO
- **Rol en el evento:** Ponente

- **Nombre:** LISETH VALENTINA BURBANO RODRÍGUEZ

- **Rol en el evento:** Ponente

- **Nombre:** ANA MILENA ROJAS CALERO

- **Rol en el evento:** Ponente

Nombre del evento: ece2i ENCUENTRO CIENTÍFICO Y EDUCATIVO DE EXPERIENCIAS INVESTIGATIVAS

Tipo de evento: Encuentro

Ámbito: Nacional

Realizado el: 01-08-09-16, en CALL - Institución Universitaria Antonio José Camacho.

Productos asociados

- **Nombre del producto:** Punto Ecológico Inteligente para fomentar la cultura ambiental en las instalaciones de la Institución Universitaria Antonio José Camacho

- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** ANDRÉS CAMILO ANGULO CORREA.

- **Rol en el evento:** Ponente

- **Nombre:** ERIKA SARRIA

- **Rol en el evento:** Ponente

Nombre del evento: V ENCUENTRO REGIONAL DE SEMILLEROS DE INVESTIGACIÓN IES TÉCNICAS Y TECNOLÓGICA DEL VALLE DEL CAUCA

II ENCUENTRO INTERNACIONAL DE GRUPOS Y SEMILLEROS DE INVESTIGACIÓN TÉCNICA PROFESIONAL Y TECNOLÓGICA

Tipo de evento: Encuentro

Ámbito: Nacional

Realizado el: 2017-04-26, 2017-04-27 en SANTANDER DE QUILICHAO CAUCA - Corporación Universitaria COMFACAUCA.

Productos asociados

- **Nombre del producto:** Sistema de información para la integración de componentes de software que permita la implementación de un módulo que integre la funcionalidad de control de seguridad y consulta de notas para su acceso desde un aplicativo móvil dirigido a los estudiantes de la Institución Universitaria Antonio José Camacho.
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** FABIÁN ESTEBAN CAPOTE CASAS
- **Rol en el evento:** Ponente
- **Nombre:** LEANDRO FLÓREZ ARISTIZÁBAL
- **Rol en el evento:** Ponente

Nombre del evento: VI ENCUENTRO REGIONAL DE SEMILLEROS DE INVESTIGACIÓN IES TÉCNICAS Y TECNOLÓGICA DEL VALLE DEL CAUCA
II ENCUENTRO INTERNACIONAL DE GRUPOS Y SEMILLEROS DE INVESTIGACIÓN TÉCNICA PROFESIONAL Y TECNOLÓGICA

Tipo de evento: Encuentro

Ámbito: Nacional

Realizado el: 018-04-25, 2017-04-26 en CALI - UAOTEC, Sede San Fernando

Productos asociados

- **Nombre del producto:** Prototipo de Laboratorio Remoto de Redes
- **Tipo de producto:** Producción técnica - Presentación de trabajo - Ponencia

Participantes

- **Nombre:** LISETH VALENTINA BURBANO RODRÍGUEZ
- **Rol en el evento:** Ponente
- **Nombre:** YESICA YULIANA MORENO GUERRERO
- **Rol en el evento:** Ponente

- Nombre: ANA MILENA ROJAS CALERO
- Rol en el evento: Ponente

Registros de software

Producción técnica - Softwares - Computacional

Manuel Alejandro Pastrana Pardo. Sistema de registro y reportes de las acciones de proyección social de la Institución Universitaria Antonio José Camacho de la ciudad de Cali.

Nombre comercial: Sistema de registro y reportes de las acciones de proyección social de la Institución Universitaria Antonio José Camacho de la ciudad de Cali.

contrato/registro: En: Colombia, 2018

Plataforma: El usuario puede tener acceso al aplicativo mediante un dispositivo que tenga un web browser y esté conectado a una red o intranet, por medio del puerto 8080 (por defecto) usando el protocolo TCP/IP a través de HTTP. Se establece un puente de conexión entre el usuario y el dispositivo, y el servidor de aplicaciones glassfish que contiene la aplicación Java desplegada. El servidor de aplicaciones, mediante el uso de un pool de conexiones y sesiones, se conecta a un servidor de base de datos Oracle a través del puerto 1521 (por defecto) usando el protocolo TCP/IP con un usuario y contraseña de conexión, de donde extrae toda la data permitida y solicitada por el usuario mediante el web browser.

Palabras clave: Arquitectura de software, Procesos Para Desarrollo de Software, Scrum, Spring Framework.

Áreas: Ingeniería y Tecnología, Ingenierías Eléctrica, Electrónica e Informática, Ingeniería de Sistemas y Comunicaciones.

Producción técnica - Softwares - Computacional

Manuel Alejandro Pastrana Pardo, Leandro Flórez Aristizábal. Sistema para la Gestión de Comunicación de Carácter Académico entre los Estudiantes y Docentes en la Institución Universitaria Antonio José Camacho.

Nombre comercial: Sistema para la Gestión de Comunicación de Carácter Académico entre los Estudiantes y Docentes en la Institución Universitaria Antonio José Camacho.

Contrato/registro: En: 2018.

Plataforma: El aplicativo funciona para las plataformas Android e iOS.

Palabras: Arquitectura de software, Desarrollo de software, Procesos Para Desarrollo de Software, Scrum, smart campus.

Áreas: Ingeniería y Tecnología, Ingenierías Eléctrica, Electrónica e Informática, Ingeniería de Sistemas y Comunicaciones.

Producción técnica - Softwares - Computacional

Ana Milena Rojas Calero, Juan Pablo Dorado Mesa. Desarrollo de una plataforma Web para la Gestión de Contratos de Mínima Cuantía y Contratación Directa en la Institución Universitaria Antonio José Camacho.

Nombre comercial: Desarrollo de una plataforma Web para la Gestión de Contratos de Mínima Cuantía y Contratación Directa en la Institución Universitaria Antonio José Camacho.

Contrato/registro: En: 2018.

Plataforma: Unos de los aspectos más relevantes en el cumplimiento de la función pública es el manejo de la gestión contractual, por la incidencia directa que esta tiene sobre el cumplimiento de las funciones constitucionales, que son asignadas en el cumplimiento de las funciones que les son asignadas en la inversión y en el manejo del presupuesto público.

El aplicativo está en desarrollado en PHP, la plataforma web permite elaborar los estudios previos para la contratación directa y de mínima cuantía; también permite llevar un control documental de las actas o documentos que son elaboradas por el interventor del contrato, posterior a la firma de la suscripción del mismo.

Palabras clave: Contratación pública, estudio previo, Smart Government, procesos de desarrollo de software, Scrum, smart campus.

Áreas: Ingeniería y Tecnología, Ingenierías Eléctrica, Electrónica e Informática, Ingeniería de Sistemas y Comunicaciones.

Producción técnica - Softwares - Computacional

Ana Milena Rojas Calero, Manuela Cardona Arias, Rubens Diez Puli. Desarrollo de un Sistema de Información para la gestión de proyectos de Smart Campus

Nombre comercial: Sistema de Información para la gestión de proyectos de Smart Campus

Contrato/registro: En: 2019.

Plataforma: El sistema de Información para la gestión de proyectos de Smart Campus, permite ver en tiempo real la visibilidad del flujo de trabajo de los equipos de desarrollo que se encuentran vinculados al proyecto de investigación. El tablero consta de cuatro columnas (Por hacer, Haciendo, Pruebas, Terminado). Este proyecto busca contribuir significativamente a mejorar el control y seguimiento de los proyectos de grado en modalidad asistida en el grupo de investigación GRINTIC, facilitando al director de proyecto y estudiantes, la comunicación, el seguimiento de tareas, la gestión documental y finalmente la gestión del conocimiento de cada proyecto. Es importante resaltar que el marco de trabajo con el cual se realiza el proceso de elaboración del proyecto está basado en el framework Scrum, su desarrollo se centra en la aplicación de tableros utilizando la metodología Kanban.

Palabras clave: Smart Campus, Gestión de proyectos, Kanban, tablero, gestión de conocimiento, gestión de documentos

Áreas: Ingeniería y Tecnología, Ingenierías Eléctrica, Electrónica e Informática, Ingeniería de Sistemas y Comunicaciones.

Impacto en la comunidad

Los impactos alcanzados en la fase I y fase II del proyecto de investigación Ecosistema Smart Campus UNIAJC serán abordados desde dos perspectivas: funciones sustantivas y funciones adjetivas de las Instituciones de Educación Superior (IES).

La primera tiene que ver con las funciones misionales de la educación, enunciadas en el marco de la Ley General de Educación 115 del año 1994, que son: Docencia, Investigación y Extensión. Esta última entendida en un sentido mucho más amplio en las IES como la proyección social. No obstante, en el mundo hay una fuerte tendencia que ha sido acogida en el Plan Estratégico de Desarrollo de la Institución Universitaria Antonio José Camacho 2020-2030, para que el Bienestar Universitario se reconozca también como una función sustantiva y, en tal sentido, se hará el análisis de los impactos misionales para estos cuatro ejes de desarrollo.

De otro lado, la segunda perspectiva desde la cual se analizarán los impactos es desde el punto de vista de las funciones adjetivas, que son funciones que no hacen parte del núcleo misional del desarrollo, pero que comprenden las acciones y programas asociados a mejorar la calidad y eficiencia de las funciones misionales. En tal sentido, son todas las áreas de la universidad que trabajan para que las funciones misionales puedan desarrollarse de la forma más adecuada. Hacen parte de las funciones adjetivas: las funciones administrativas, el mercadeo, la gestión tecnológica y la gestión logística. Es por esa razón que también se analizarán los impactos que ha tenido el proyecto en estas áreas de desempeño de la universidad.

Si bien, históricamente los pilares fundamentales del desarrollo de las funciones misionales en el mundo han sido la docencia y la investigación, en la universidad latinoamericana, en términos generales, estos dos pilares no han sido fáciles de conciliar, dado que son producto de modelos universitarios primigenios clásicos, con un fuerte arraigo euro centrista, en los cuales se privilegiaban con diferentes pesos cada una de estas funciones misionales.

Este fenómeno ha hecho que en las universidades se genere una tensión entre las funciones de docencia y las funciones de investigación. En este escenario, los profesores reclaman más tiempo para poder desarrollar la investigación, lo que termina generando una lucha entre el tiempo del docente para ejercer las funciones de investigación y el tiempo destinado a las funciones de docencia.

Uno de los principales impactos que ha tenido el proyecto ha sido propiciar un escenario de diálogo entre estas dos funciones misionales. En consecuencia, el desarrollo de esta iniciativa ha permitido la articulación curricular de las estructuras investigativas con el desarrollo de la oferta académica, que se realiza a través de los docentes investigadores. Esta estrategia nutre el proceso de construcción de conocimiento a través de la vinculación de estudiantes de los diferentes cursos en los procesos de investigación, permitiendo que los investigadores del grupo GRINTIC puedan gestionar el talento humano en las asignaturas que estratégicamente se han escogido para apoyar el desarrollo del proyecto de investigación. Estas asignaturas son: programación, ingeniería de software, programación de dispositivos móviles, temática de grado, proyecto integrador. De esta forma, los profesores investigadores han tenido la oportunidad de poder identificar los estudiantes que harán parte del colectivo de investigación, los cuales podrán vincularse al proyecto mediante el ingreso al semillero de investigación ITMedia y a través de su trabajo en

estos colectivos; estos estudiantes pueden iniciar el desarrollo de proyectos de grado en modalidad asistida en el grupo de investigación.

Al igual que la estructura curricular ha provisto talento humano para el desarrollo del proyecto de investigación, de forma recíproca, la estructura investigativa ha contribuido con el proceso de actualización curricular mediante el desarrollo de contenidos pertinentes y acordes a las necesidades de la industria del software, experiencias educativas basadas en el aprendizaje significativo, traídas de los resultados de investigación y plataformas tecnológicas modernas para apoyar el desarrollo de los procesos de formación.

Otro impacto en la función misional de la investigación está relacionado con el fortalecimiento de los indicadores y características de la acreditación de alta calidad. En particular, con los referidos a la investigación, innovación, creación artística y cultural, así como también al de visibilidad nacional e internacional. Es importante mencionar que, a través de los diferentes resultados de investigación, se aportaron productos para la categorización del grupo de investigación GRINTIC en las convocatorias de medición de grupos, desarrollo tecnológico o de innovación y el reconocimiento de investigadores del Sistema Nacional de Ciencia y Tecnología e Innovación (SNCTEI), definidos por Colciencias. Aquí, efectivamente resalta como aporte significativo el desarrollo de productos de nuevo conocimiento, productos de apropiación social de conocimiento y productos de actividades relacionadas con la formación del talento humano, entre los que podemos mencionar: la publicación de artículos en revistas indexadas, la participación en encuentros científicos a nivel nacional e internacional, encuentros de semilleros, consecución de registros de software, desarrollo de talleres, dirección de trabajos de grado, los cuales se encuentran expuestos en el capítulo de resultados de este libro.

Estos resultados han favorecido no sólo los procesos de medición del grupo, sino que también han permitido a los docentes avanzar en los procesos de desarrollo y cualificación profesoral propios de la institución, participar en eventos nacionales e internacionales de divulgación de conocimiento, publicación de resultados de investigación, registros de software y demás actividades derivadas de los procesos de investigación, financiadas por la Institución y que contribuyen al mejoramiento del escalafón de los docentes y al ascenso del perfil como investigador en Colciencias.

Los estudiantes investigadores también son beneficiarios de algunas de estas medidas de fomento de la investigación, mediante la financiación de la movilidad a eventos de divulgación, encuentros de semilleros y publicación de sus resultados en revistas científicas o de divulgación.

Desde el punto de vista de la función de proyección social también se ha conseguido un impacto muy relevante, a través del desarrollo del proyecto integrado con las funciones de investigación por medio del desarrollo de proyectos de grado que están orientados a la solución de problemas organizacionales del sector educativo desde un enfoque metodológico de marcos de trabajo ágil como: SCRUM, prácticas DevOps para el aseguramiento de la calidad del software y la adopción de marcos internacionales para la gestión de proyectos, lo que ha permitido a nuestros estudiantes hacer el cierre de brechas entre los conocimientos académicos de los procesos de formación universitaria y las necesidades reales de la industria del software.

La atención de estas necesidades a través de procesos empresariales ha permitido que el estudiante se acerque a una experiencia significativa en términos de su desarrollo profesional, potenciando sus oportunidades de inserción laboral, de mejoramiento de su calidad de vida y de su desarrollo profesional, convirtiendo al semillero de investigación en un referente para los centros tecnológicos de mayor importancia ubicados en el Valle del Cauca, Medellín y Bogotá, a los cuales han llegado varios de nuestros estudiantes.

También en el eje de proyección social y con la colaboración de profesores del equipo de investigadores, se realizó una consultoría a la fundación FUNIAJC en la creación del Centro de Formación de Capacidades Laborales y Empresariales, el diseño y actualización de ambientes de aprendizaje y el desarrollo curricular de una oferta académica especializada para el sector TIC.

Desde el punto de vista de la función misional del bienestar universitario, se han producido impactos positivos en varios niveles de la organización, el primero de ellos directamente en el equipo de jóvenes investigadores del semillero ITMedia, que por su participación en esta iniciativa son elegibles para el estímulo denominado becario de investigación, a través del cual el estudiante recibe un apoyo financiero equivalente al 80% de los derechos de matrícula y un auxilio de transporte que busca mitigar las condiciones de vulnerabilidad de los estudiantes de la Institución Universitaria Antonio José Camacho, los cuales en un 97% pertenecen a los estratos socioeconómicos 1, 2 y 3.

A nivel organizacional, también se tienen impactos significativos desde el eje misional del bienestar en la gestión de citas e historias clínicas para la orientación psicosocial, gestión de cafeterías en el campus universitario y en la generación de apoyos financieros mediante el programa referidos.

Por otra parte, están los impactos alcanzados desde las funciones adjetivas, en las cuales también hay resultados en diferentes aspectos. El primero de ellos tiene que ver con la gestión financiera, en tanto que se ha logrado el incremento de número de estudiantes resultado de la implementación de los proyectos: la integración del CRM con el sistema ERP empresarial Academusoft, la implementación de un Call Center y el desarrollo de la App de referidos Unidos Unicamacho, lo cual ha permitido generar estrategias de marketing digital y un permanente control de la gestión de cierre de venta de servicios educativos, mediado por el monitoreo del proceso de maduración de los aspirantes preinscritos a estudiantes matriculados, a través de sistemas de inteligencia de negocios. El incremento de estudiantes que contribuye a la ocupación de la capacidad instalada ociosa ha tenido un impacto en el mejoramiento de la eficiencia organizacional y ha favorecido el mejoramiento de la estabilidad y las condiciones de sostenibilidad de la universidad.

Dentro de las funciones adjetivas está también la gestión tecnológica, en la cual se ha avanzado en la estandarización del proceso de desarrollo de software mediante la implementación de buenas prácticas, el aseguramiento de calidad preventiva y el fortalecimiento de infraestructura tecnológica de servidores, consolidando una plataforma de desarrollo que ha hecho que el espacio dedicado para el semillero ITMedia se convierta en un laboratorio de práctica para los jóvenes investigadores que con las soluciones desarrolladas en modalidad de proyectos de grado contribuyen a la transformación digital de la universidad.

En el ámbito académico-administrativo, los desarrollos que han tenido mayor impacto han sido el desarrollo de la App de consulta de notas, la App de certificados académicos, la App para la comunicación docente-estudiante, la App calendario académico, la implementación de la ventanilla única, la App para la elaboración de diagnósticos académicos y quices, el diseño e implementación de solución analítica B.I para la Gestión de Indicadores Académicos, la implementación del flujo trabajo y trazabilidad para el sistema de información web para la gestión de contratos, el Sistema de Información para la Gestión de Proyectos Smart Campus, el diseño e implementación de

un sistema web para categorización docente (CATEDOC). Estas son algunas de las soluciones desarrolladas en el marco del proyecto de investigación.

De forma paralela, con el semillero SELECT se realizaron los siguientes proyectos: implementación de un prototipo para el control de acceso inteligente mediante tarjetas RFID y sensor biométrico para huella dactilar; el diseño de un dispositivo de medición inteligente para la monitorización del consumo energético; la implementación de un dispositivo de monitoreo para un sistema inteligente con sensores distribuidos para las variables de temperatura y humedad en el NOC de la UNIAJC y el desarrollo de un prototipo ecológico inteligente para generar una cultura de cuidado al medio ambiente en la comunidad universitaria. Con la realización de estos proyectos se han integrado estudiantes de programas de ingeniería electrónica y de tecnología en electrónica industrial, los cuales han sido guiados por profesores investigadores del grupo de investigación INTELIGO de la Facultad de Ingenierías.

6 Conclusiones y trabajos futuros

La priorización del proyecto en el marco del Plan Estratégico de Desarrollo de la universidad y la alineación de este con los objetivos estratégicos ha sido fundamental para el desarrollo de un ecosistema en la institución. Esto, por supuesto, ha requerido trabajar coordinadamente en el desarrollo de proyectos que articulan los ámbitos de desarrollo de un *Smart Campus*: Smart Living, Smart Environment, Smart People, Smart Economy, Smart Government y Smart Movility. Todos estos ámbitos han sido abordados desde el desarrollo de iniciativas que le apuntan al desarrollo de las funciones misionales y adjetivas de la universidad.

También, este proceso de transformación ha requerido la reestructuración de políticas al interior de la universidad. De la misma forma, la tecnología ha jugado un papel muy importante para lograr este ambicioso propósito en el que la participación de docentes y jóvenes investigadores ha jugado un papel muy importante. Ellos han sido los artífices de los resultados alcanzados hasta hoy, no sólo con la generación de productos de software, sino también con su participación en congresos mediante ponencias y publicaciones en revistas indexadas, que han permitido darle visibilidad al grupo GRINTIC, tanto en el ámbito nacional como internacional. Estos resultados obtenidos dan cuenta de la importancia de articular la investigación y la docencia como ejes transversales para el desarrollo del proyecto.

Asimismo, la adopción de un modelo de desarrollo basado en los marcos de desarrollo ágil como SCRUM y buenas prácticas de T.I. ha permitido consolidar un semillero de investigación con altas competencias humanas, que a la vez ha facilitado el fortalecimiento de capacidades institucionales para la producción de software, lo cual podría potenciar a futuro el desarrollo, innovación y emprendimiento, mediante la creación de Startup centradas en el desarrollo de productos y servicios empresariales que permitan avanzar de forma más rápida en el proceso de modernización y transformación digital de la universidad.

De la misma forma, el modelo de desarrolló permitió que los productos de software fueran creados con altos estándares de calidad, lo que a futuro facilitará el mantenimiento y la escalabilidad de las soluciones, para un crecimiento evolutivo efectivo y eficiente, dadas las necesidades cambiantes de la institución frente a nuevos requerimientos y mejoras que se puedan dar a futuro.

Los trabajos futuros para desarrollar en la tercera fase del proyecto de investigación estarán encaminados a fortalecer las capacidades institucionales para el desarrollo de proyectos orientados a mitigar la deserción académica de estudiantes, al fortalecimiento de las plataformas de interacción para la atención de servicios universitarios, la integración de servicios a través del desarrollo de una mesa de ayuda y la modernización de la plataforma de comunicaciones empresarial. Adicionalmente, se invertirán recursos para monitorear el aforo de ingreso a la universidad, que permita a futuro analizar variables tales como: ausencia intersemestral, control de asistencia de estudiantes, docentes y personal administrativo.



Referencias

- Beck, K. (1999). Extreme Programming Explained: Embrace Change. In *XP Series*. <https://doi.org/10.1136/adc.2005.076794>
- Cohn, M. (2004). User Stories Applied: For Agile Software Development (Addison Wesley Signature Series). *Writing*, 1. <https://doi.org/10.1017/CBO9781107415324.004>
- Delgado, G., & Gutiérrez, D. (2020). Sistema para la gestión de solicitudes a la facultad de ingenierías de la Institución Universitaria Antonio José Camacho (Trabajo de Grado). Institución Universitaria Antonio José Camacho. Cali, Colombia.
- European Comission. (2010). Europa 2020: Una estrategia para un crecimiento inteligente, sostenible e integrador. In *COM(2010) 2020*. <https://doi.org/10.1007/s13398-014-0173-7.2>
- Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, 9, 28–35. <https://doi.org/10.1177/004057368303900411>
- Institución Universitaria Antonio José Camacho - UNIAJC. (2015). *Anuario Estadístico 2015*. Cali: Institución Universitaria Antonio José Camacho.
- Institución Universitaria Antonio José Camacho - UNIAJC. (2020). *Plan Estratégico de Antonio José Camacho 2020-2030*. Cali: Institución Universitaria Antonio José Camacho.
- Kruchten, P. (2006). Planos Arquitectónicos : El Modelo de " 4 + 1 " Vistas de la La Arquitectura del Software. *IEEE Software*, 12(6), 1–16.
- Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, 157, 110384. <https://doi.org/10.1016/j.jss.2019.07.083>
- Pastrana, M., Ordoñez, H., & Cobos, C. (2019). ISO 29110 en Colombia : de la teoría a la práctica - ISO 29110 in Colombia : from theory to practice. *Guillermo de Ockham: Revista Científica*, 17(2), 71–80.

- Pastrana, M., Ordóñez, H., Ordonez, A., & Merchan, L. (2017). Requirements elicitation based on inception deck and business processes models in scrum. In *Communications in Computer and Information Science*, 735. https://doi.org/10.1007/978-3-319-66562-7_24
- Pastrana, M., Ordóñez, H., Ordonez, A., Thom, L. H., & Merchan, L. (2018). Optimization of the inception deck technique for eliciting requirements in SCRUM through business process models. *Lecture Notes in Business Information Processing*, 308. https://doi.org/10.1007/978-3-319-74030-0_52
- Pastrana, M., Ordoñez, H., Rojas, A., & Ordonez, A. (2019). Ensuring Compliance with Sprint Requirements in SCRUM: Preventive Quality Assurance in SCRUM. *Advances in Intelligent Systems and Computing*, 924, 33–45. https://doi.org/10.1007/978-981-13-6861-5_3
- PMBOK. (2017). PMBOK Guide - 6th Edition. In *Project Management Institute*. <https://doi.org/10.1002/pmj.20125>
- Pressman, R. S., & Maxim, B. R. (2015). Software Engineering : A Practitioner's Approach, Eighth Edition. In McGraw-Hill (Ed.), *ACM SIGSOFT Software Engineering Notes* (McGraw-Hill). Boston, USA: McGraw-Hill.
- Rasmusson, J. (2010). *The Agile Samurai—How Agile Masters Deliver Great Software*. Dallas: Pragmatic Bookshelf.
- Schwaber, K., & Sutherland, J. (2017). *La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*. Retrieved from <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf#zoom=100>
- Venters, C., Capilla, R., Betz, S., Penzenstadler, B., Crick, T., Crouch, S., Nakagawa, E. Y., Becker, C., & Carrillo, C. (2018). Software Sustainability: Research and Practice from a Software Architecture Viewpoint. *Journal of Systems and Software*, 138, 174-188. <https://doi.org/10.1016/j.jss.2017.12.026>
- Leiton Muñoz, W.A. & Moya Loaiza, B.A. (2020). *Documentación e implementación de buenas prácticas DevOps con herramientas de*

automatización de pruebas y de despliegue a través del refinamiento de software de Smart Campus (Trabajo de grado). Institución Universitaria Antonio José Camacho. Cali, Colombia.

