

Implementación de la metodología BPM y herramientas software para ejercicios de autoevaluación institucional y de programas académicos, como estrategia para el fortalecimiento de procesos de aseguramiento de la calidad en la Universidad Nacional Abierta y a Distancia – UNAD

Arellys de Jesús Correa Rodríguez

Universidad Nacional Abierta Y A Distancia – Unad
Escuela De Ciencias Básicas Tecnologías E Ingeniería
Maestría En Gestión De Tecnologías De La Información
Cartagena, D.T. y C
2021

Implementación de la metodología BPM y herramientas software para ejercicios de autoevaluación institucional y de programas académicos, como estrategia para el fortalecimiento de procesos de aseguramiento de la calidad en la Universidad Nacional Abierta y a Distancia – UNAD

Ing. Arellys de Jesús Correa Rodríguez

Director: Ing., Mg. Claudio Camilo González Clavijo

Universidad Nacional Abierta y a Distancia – UNAD
Escuela de Ciencias Básicas Tecnologías e Ingeniería
Maestría en Gestión de la Información
Cartagena, D.T. y C.

2021

Dedicatoria

El presente documento como resultado de trabajo de grado se lo dedico en primer lugar a Dios, pues fue gracias a Él que pude desarrollar mis estudios de maestría, realizar un posgrado en mi área de formación era un deseo que albergaba hace mucho tiempo en mi corazón, pero siendo madre cabeza de familia mis recursos son limitado. Al salir la segunda convocatoria de MINTIC en el 2018, pude aplicar y a pesar de varios inconvenientes del momento fui aceptada dentro de los elegidos.

Adicionalmente, es un logro que también dedico a mis padres e hijas quienes siempre me apoyaron en mi proceso académico, porque estudiar un posgrado demanda mucho esfuerzo, tiempo, dedicación y sobre todo sacrificio, pero su amor, comprensión y fortaleza siempre me brindaron

Agradecimientos

Agradezco en primer lugar a Dios, por brindarme la oportunidad de seguir de salir seleccionada entre los beneficiarios de la Convocatoria de educación formal nivel posgrado FONTIC-ICETEX y así seguir capacitándome.

A mis padres e hijas por su gran motivación a seguir adelante y no desfallecer, su apoyo incondicional en todo este tiempo fue valioso.

A mi directo de proyecto Ingeniero Claudio Camilo González Clavijo, por compartir sus conocimientos, orientaciones y darme la oportunidad de aprender a investigar, su rigurosidad lo hace un mentor excepcional.

A la Dra. María Nancy Garzón Líder Aseguramiento de la calidad UNAD por su disposición y amabilidad, de igual manera por darme la oportunidad de trabajar en esta iniciativa tan importante y necesaria para la institución.

A mis compañeros de maestría Shirley Quintana, Ulises Díaz, Nathalia Cuervo, Sandra Benavides, Helber Vergara y Fredy Castellanos por los todos los momentos compartidos y el trabajo en equipo.

A mis profesores, por ser guías y luz durante mi proceso de aprendizaje.

Al Ministerio de Tecnologías de Información y Comunicación – MINTIC por su Financiamiento en la Segunda Convocatoria de Gestión TI y Seguridad de la Información.

Resumen

La autoevaluación de programas académicos y de Instituciones de Educación Superior es una actividad que consiste en una revisión sistemática interna de los procesos y sus resultados, mediante la apreciación de sus actores y la verificación de las evidencias documentales, con el fin de mejorar la calidad educativa y de servicios a sus grupos de interés. Se utiliza en procesos de aseguramiento de la calidad para el reconocimiento de acreditación de alta calidad, renovación del registro calificado y certificaciones internacionales, bajo el concepto de Assessment. La presente investigación, describe la implementación de la metodología BPM para el análisis en detalle del proceso de autoevaluación de programas académicos en la Universidad Nacional Abierta y a Distancia, a través del estado actual (AS-IS) y, en consecuencia, establecer su rediseño modelando su Estado Futuro (TO-BE). Finalmente, su automatización se propone por medio de una herramienta de software cuyo propósito es superar las brechas en la ejecución del proceso.

Palabras claves: Autoevaluación de Programas, Aseguramiento de Calidad, metodología BPM, Arquitectura de Solución, Desarrollo de Software.

Abstract

The self-evaluation of academic programs and Higher Education Institutions is an activity that consists of an internal look at the processes and their results, through the appreciation of their actors and the verification of documentary evidence, in order to improve educational quality and services for stakeholders; This is applied in quality assurance processes for the recognition of high quality accreditation, renewal of the qualified registry and international certifications, under the concept of Assessment. This research describes the implementation of the BPM methodology for the detailed analysis of the self-evaluation process of academic programs at the Universidad Nacional Abierta y a Distancia, through the current state (AS-IS) and then establish its redesign modeling its Future State (TO-BE). Finally, its automation is proposed by means of a software tool whose purpose is to overcome the gaps in the execution of the process.

Keywords: Program Self-Assessment, Quality Assurance, BPM methodology, Solution Architecture, Software Development.

Tabla de Contenido

Lista de Tablas	12
Tabla de Figuras.....	16
Introducción	20
Generalidades de la investigación.....	23
Planteamiento del problema.....	23
Formulación del problema	27
Objetivos	27
Objetivo General	27
Objetivos Específicos.....	27
Alcance y Delimitación.....	28
Justificación	28
Marco Referencial.....	31
Revisión de literatura	31
Marco Conceptual y Teórico	37
Metodología.....	40
Fase 1 Análisis del proceso mediante la metodología de Gestión de Procesos de Negocios BPM	41
Identificación del Proceso.....	42
Descubrimiento del Proceso.	42

Análisis del Proceso.....	42
Rediseño del Proceso.	42
Fase 2 Diseño de la arquitectura de software para el sistema de autoevaluación.....	42
Fase 3 Desarrollo del sistema mediante la metodología del proceso de desarrollo ágil Scrum	43
Análisis del proceso mediante la metodología de Gestión de Procesos BPM.....	44
Identificación del procedimiento Gestión de la autoevaluación de programas	44
Actividades del procedimiento en su estado actual	44
Descubrimiento del proceso.....	45
Gestión de los interesados.....	45
Caracterización y descripción de los subprocesos del proceso de autoevaluación.....	53
Análisis del proceso de autoevaluación estado actual AS-IS	56
Diagrama 1 Modelo del Proceso de Autoevaluación AS-IS.....	56
Análisis de los subprocesos del proceso de autoevaluación estado actual AS-IS	59
Diagrama 2 Modelo Preparación de insumos.	59
Diagrama 3 Subproceso Diseño y aplicación de instrumentos	61
Diagrama 4 Subproceso Recopilación documental y verificación de cumplimiento .	63
Diagrama 5 Informe de autoevaluación y plan de mejoramiento	65
Rediseño del proceso de Autoevaluación Propuesta Estado Futuro (TO-BE)	68
Definición de las Necesidades	69

Diagrama 6 Estado Futuro del Proceso de Autoevaluación TO-BE.....	71
Rediseño de los subprocesos de proceso de autoevaluación estado futuro (TO-BE)	75
Diagrama 7 Subproceso Preparación de Insumos TO-BE.....	75
Diagrama 8 Subproceso Diseño, Aplicación y Tabulación de Instrumentos TO-BE.....	77
Diagrama 9 Recopilación Documental y Verificación de cumplimiento TO-BE	79
Diagrama 10 Subproceso Informe de autoevaluación y plan de mejoramiento TO-BE	81
Arquitectura de la solución para el proceso de Autoevaluación.....	83
Selección de la Arquitectura de Software	83
Diseño de la arquitectura lógica de la aplicación.....	85
Selección de aplicación a desarrollar	86
Selección de Puntos de Vista para la Arquitectura y Diseño de Artefactos	86
Vista Funcional del Sistema.....	89
Vista de Información Sistema de Autoevaluación.....	90
Vista de Desarrollo Sistema de Autoevaluación.....	91
Vista de Implementación Física Sistema de Autoevaluación	93
Desarrollo e Implementación del Prototipo de Software Para el Proceso de Autoevaluación	94
Identificación de requerimientos por historias de Usuario	94

	10
Desarrollo del Back-End.....	103
Tecnologías Empleadas para el back-end	106
Creación del proyecto y configuración en el repositorio remoto GitLab	106
Planificación y desarrollo de los Sprints en el repositorio remoto GitLab	107
Diseño de los modelos de las tablas maestras del negocio y el repositorio de archivos.	109
Implementación del servidor web	110
Desarrollo del Front-End.	111
Tecnologías Empleadas para el front-end.....	111
Diseño de wireframes	113
Configuración del Frontend	117
Implementación del cliente web	118
Elaboración de pruebas de funcionalidad con usuarios	121
Resultados	124
Conclusiones	126
Recomendaciones	128
Referencias.....	129
Anexos	136
Anexo 1 Ficha Ponderación de Características.....	136
Anexo 2 Ficha de Muestreo	139

Anexo 3 Ficha de Seguimiento Aplicación de Instrumentos.....	140
Anexo 4 Matriz Apreciación de Estamentos	141
Anexo 5 Ficha Análisis Documental	144
Anexo 6 Código Fuente	145
Anexo 7 Formato de validación de pruebas –Usuario VISAE-Calidad	156
Anexo 8 Cesión de derechos patrimoniales	170

Lista de Tablas

Tabla 1.	26
Objetos del Modelo de Autoevaluación UNAD	26
Tabla 2.	37
Marco Conceptual	37
Tabla 3.	46
Matriz de los Interesados del Proyecto de Autoevaluación	46
Tabla 4.	53
Clasificación de los Interesados del Proyecto de Autoevaluación	53
Tabla 5.	54
Subproceso 1. Preparación de insumos	54
Tabla 6.	54
Subproceso Diseño y aplicación de instrumentos	54
Tabla 7.	55
Subproceso Recopilación documental y verificación de cumplimiento	55
Tabla 8.	56
Informe de autoevaluación y plan de mejoramiento	56
Tabla 9.	70
Definición de Necesidades del estado Futuro	70
Tabla 10.	84

Estructura de la arquitectura de tres (3) capas	84
Tabla 11.	85
Capas Patrón MVC	85
Tabla 12.	87
Puntos de Vistas Arquitecturales Propuestos por Rozanski & Woods, 2012.....	87
Tabla 13.	95
Historia de Usuario HU001	95
Tabla 14.	95
Historia de Usuario HU002.....	95
Tabla 15.	96
Historia de Usuario HU004.....	96
Tabla 16.	96
Historia de Usuario HU005.....	96
Tabla 17.	97
Historia de Usuario HU006.....	97
Tabla 18.	97
Historia de Usuario HU007.....	97
Tabla 19.	98
Historia de Usuario HU008.....	98
Tabla 20.	98

Historia de Usuario HU009	98
Tabla 21.	99
Historia de Usuario HU010	99
Tabla 22.	99
Historia de Usuario HU011	99
Tabla 23.	100
Historia de Usuario HU012	100
Tabla 24.	100
Historia de Usuario HU013	100
Tabla 25.	101
Historia de Usuario HU014	101
Tabla 26.	101
Historia de Usuario HU015	101
Tabla 27.	102
Historia de Usuario HU016	102
Tabla 28.	105
Estructura del Modelo MVT)	105
Tabla 29.	112
Tecnologías Empleadas diseño de Front-End	112
Tabla 30.	123

Formato Elaboración de pruebas Funcionales123

Tabla de Figuras

Figura 1.	32
<i>Algoritmo RSL basado en la metodología de Kitchenham</i>	32
Figura 2.	41
<i>Fases del proyecto</i>	41
Figura 3.	44
<i>Identificación procedimiento Gestión de la autoevaluación de programas. Fuente:</i> <i>(UNAD, 2021)</i>	44
Figura 4.	58
<i>Diagrama AS-IS del proceso de Autoevaluación</i>	58
Figura 5.	60
<i>Diagrama AS-IS Preparación de Insumos</i>	60
Figura 6.	62
<i>Diagrama AS-IS Diseño Aplicación y Tabulación de Instrumentos</i>	62
Figura 7.	64
<i>Diagrama AS-IS Recopilación Documental y verificación de Cumplimiento</i>	64
Figura 8.	66
<i>Diagrama AS-IS Informe de Autoevaluación</i>	66
Figura 9.	74

<i>Estado Futuro del Proceso de Autoevaluación TO-BE</i>	74
Figura 10.	76
<i>Subproceso Preparación de Insumos TO-BE</i>	76
Figura 11.	78
<i>Subproceso Diseño, Aplicación y Tabulación de Instrumentos TO-BE</i>	78
Figura 12.	80
<i>Recopilación Documental y Verificación de cumplimiento TO-BE</i>	80
Figura 13.	82
<i>Subproceso Informe de autoevaluación y plan de mejoramiento TO-BE</i>	82
Figura 14.	90
<i>Modelo de Componentes Sistema de Autoevaluación</i>	90
Figura 15.	91
<i>Modelo de Datos Encuesta del proceso de Autoevaluación</i>	91
Figura 16.	92
<i>Modelo de Estructura de módulos Sistema de Autoevaluación UNAD</i>	92
Figura 17.	93
<i>Modelo de Despliegue Sistema de Autoevaluación UNAD</i>	93
Figura 18.	103
<i>Arquitectura de la aplicación</i>	103
Figura 19.	105

<i>Modelo Vista Plantilla -MVT-</i>	105
Figura 20.	107
<i>Configuración del back-End</i>	107
Figura 21.	109
<i>Planificación de los sprint en el repositorio</i>	109
Figura 22.	110
<i>Modelos del proyecto autoevaluación</i>	110
Figura 23.	114
<i>Wireframe Iniciar Sesión</i>	114
Figura 24.	114
<i>Wireframe crear Dimensión</i>	114
Figura 25.	115
<i>Wireframe Listar Dimensión</i>	115
Figura 26.	115
<i>Wireframe crear factor</i>	115
Figura 27.	116
<i>Wireframe crear características</i>	116
Figura 28.	116
<i>Wireframe crear indicador</i>	116
Figura 29.	117

<i>Wireframe CRUD indicador</i>	117
Figura 30.	118
<i>Configuración del front-end</i>	118
Figura 31.	119
<i>Vista Crear Dimensión</i>	119
Figura 32.	120
<i>Vista Crear Factor</i>	120
Figura 33.	121
<i>Vista crear característica</i>	121

Introducción

Las instituciones de educación superior, en busca de la calidad de sus programas académicos, se ven en la necesidad de implementar acciones estratégicas basadas en referentes válidos, asociados a procesos de aseguramiento de calidad, como los proporcionados por el Ministerio de Educación Nacional, el Consejo Nacional de Educación Superior -CESU, la Comisión Nacional Intersectorial de Aseguramiento de la Calidad de la Educación Superior - CONACES y el Consejo Nacional de Acreditación -CNA, los cuales involucran a todos los actores del sistema educativo en actividades de autoevaluación con el fin de valorar la organización y sus programas, por actores internos como externos, para así conocer desde diferentes criterios la percepción en cuanto a la calidad del servicio educativo y garantizar su mejoramiento continuo.

En consecuencia, La Universidad Nacional Abierta y a Distancia, en su Sistema de Aseguramiento de la Calidad Integral de los Programas y Servicios de la UNAD (UNAD, 2020) ha definido un modelo propio para ejercicios de autoevaluación, con base en los lineamientos del Conceso Nacional de Acreditación (Concejo Nacional de Acreditación CNA, 2013), denominado: Modelo de Autoevaluación Unadista, en el cual mediante el procedimiento Gestión de la Autoevaluación de Programas, se estable la ruta para aplicar los lineamientos y criterios para la autoevaluación de programas y servicios ofrecidos por la institución (UNAD, 2021).

No obstante, a causa de la carencia de herramientas tecnológicas apropiadas, junto con los altos volúmenes y diversidad de información, resulta compleja su ejecución en lo concerniente con la aplicación de instrumentos y la revisión y calificación de documentos. Si bien, aunque puedan existir en el mercado diferentes soluciones para gestionar tales procesos, para el caso de la UNAD, no es su mejor opción considerando que no se ajustan a sus necesidades, por su modelo de autoevaluación propio, por su modelo de operación propio, y una alta complejidad

desde el punto de vista de la cobertura de sus programas académicos y, en general de sus servicios, por tanto esas soluciones estándar no son viables para la institución; igualmente, la incorporación de una solución del mercado demandaría un ejercicio de desarrollo a la medida. Por el contrario, desde el punto de vista de las iniciativas y las estrategias institucionales para su innovación tecnológica, con sus capacidades instaladas, tanto de docentes investigadores como de estudiantes, es viable generar aportes significativos para el desarrollo de las tecnologías de la información y la comunicación de la institución. Para el caso particular que se involucra en este proyecto, es articulado por la Escuela de Ciencias Básicas, Tecnología e Ingeniería a una meta institucional para incorporar nuevas herramientas tecnológicas basadas en software para la gestión de ejercicios de autoevaluación (UNAD, 2019a).

Por ello, la presente investigación tiene como propósito implementar la metodología BPM, y herramientas de software para la gestión del proceso de autoevaluación institucional y de programas académicos, como fortalecimiento de los procesos de aseguramiento de la calidad en la Universidad Nacional Abierta y a Distancia – UNAD; en tal sentido, se ha organizado el documento de la siguiente manera:

En el capítulo 1 se presentan las generalidades haciendo una descripción detallada del problema para formularlo con la pregunta de investigación, establecer los objetivos y el alcance y de la investigación, así como su justificación.

El capítulo 2 presenta el marco referencial, el cual se compone de la revisión de la literatura y el marco conceptual y teórico que incluye referencias, conceptos y teorías relevantes asociados al proyecto.

El capítulo 3 presenta el tipo de investigación y la metodología utilizada para alcanzar el propósito establecido en ella.

El capítulo 4 presenta la aplicación de la metodología BPM para el análisis y rediseño del proceso de autoevaluación de la UNAD.

El capítulo 5 presenta la estructuración del sistema a través de la Arquitectura de software para el proceso de Autoevaluación de la UNAD, bajo el enfoque de puntos de vista.

El capítulo 6 presenta el desarrollo del prototipo de software para el proceso de autoevaluación Unadista teniendo en cuenta la metodología de desarrollo ágil SCRUM.

Generalidades de la investigación

Planteamiento del problema

Al hablar de calidad educativa se encuentran variadas definiciones; así, el Ministerio de Educación Nacional, define la calidad como “el conjunto de atributos articulados, interdependientes, dinámicos, contruidos por la comunidad académica como referentes y que responden a las demandas sociales, culturales y ambientales(Ministerio de Educación Nacional - MEN-, 2019b). Por otra parte, Cañón Rodríguez & Salazar Contreras (2011) abordan el concepto a partir de tres criterios que comprenden, calidad fundamentada en el desempeño relacionado con un modelo idealizado, calidad orientada a las acciones misionales y proyecto educativo de la institución y calidad asentada en la apreciación de los actores del proceso en función del impacto de su acción educativa.

En orden a lo anterior, las Instituciones de Educación Superior (en adelante IES) son conscientes que la búsqueda de la calidad en la educación es una necesidad en la sociedad del conocimiento, y está enmarcada en procesos de aseguramientos de la calidad y ligada a la acreditación institucional y de programas académicos, generando así una cultura de auto regulación, garantizando el mejoramiento continuo, por consiguiente, la autoevaluación hace parte fundamental de estos procesos, y el Ministerio de Educación Nacional (MEN) la define de la siguiente manera:

La autoevaluación es el proceso de reflexión interna que llevan a cabo las instituciones de Educación Superior, en el marco de su autonomía, que les permite identificar sus avances, retos y aspectos por mejorar de sus procesos académicos, administrativos, logísticos y de proyección con la comunidad. La autoevaluación es en últimas la forma objetiva como la IES manifiestan su compromiso constante con el

mejoramiento de la Calidad de la Educación Superior del país y por tanto, con una oferta educativa pertinente con los entornos sociales y naturales y coherente con sus finalidades y sus proyectos educativos.(Ministerio de Educación Nacional - MEN-, 2010).

Para procesos de acreditación de alta calidad, el Consejo Nacional de Acreditación (CNA) establece dentro de sus fases la autoevaluación y la describe como una valoración interna que radica en la revisión a fondo y sistemática de los procesos y resultados de la oferta educativa, individual o institucional, y de los procesos de apoyo con el propósito de conocer la percepción de los diversos actores del proceso educativo sobre la calidad de los productos y servicios que se generan, al igual que sobre la pertinencia y suficiencia del conjunto de evidencias que soportan la prestación del servicio. Esta actividad debe realizarse de manera constante y es liderada por la institución con base un modelo de aseguramiento de calidad propio, el propuesto por el CNA u otro modelo de referencia, según sea el caso (Concejo Nacional de Acrecitación CNA, 2013).

El MEN determinó que las instituciones deben contar con un sistema interno de aseguramiento de la calidad que contemple la gestión de la información, dispositivos que evidencien la evolución del cumplimiento de condiciones de calidad, dispositivos para garantizar la apreciación de la comunidad académica, articulación de los planes de mejoramiento con la planeación y el presupuesto institucional, dispositivos que permitan procesos continuos de autoevaluación y autorregulación, evidenciados a través informes periódicos(Ministerio de Educación Nacional -MEN-, 2019a).

La Universidad Nacional Abierta y a Distancia (UNAD), para dar cumplimiento a lo dispuesto por el MEN, cuenta con el Sistema de Aseguramiento de la Calidad Integral de los Programas y Servicios de la UNAD, adscrito a la Vicerrectoría de Servicios a Aspirantes, Estudiantes y Egresados (VISAE) y entre sus componentes, se encuentra:

El componente de evaluación y seguimiento, que involucra a toda la comunidad académica en torno a políticas, normas, herramientas, organismos, personal, procesos y procedimientos que, mutuamente relacionados, tienden a garantizar la calidad integral y la excelencia institucional. El eje articulador es el Modelo de Autoevaluación Unadista, soportado en el cumplimiento efectivo de las responsabilidades sustantivas de la Universidad (UNAD, 2020).

Este Modelo de Autoevaluación Unadista, a través del procedimiento Gestión de la Autoevaluación de Programas, establece la ruta para aplicar los lineamientos y criterios para la autoevaluación de programas y servicios ofrecidos por la UNAD, donde la consolidación de resultados y construcción del informe de autoevaluación tiene una frecuencia específica y, para los programas formales de educación superior, se deben realizar, a menos, dos (2) ejercicios de autoevaluación en la vigencia del registro calificado: el primero una vez haya completado dos (2) años de oferta y el segundo en el cuarto año de haber obtenido el registro calificado. (UNAD, 2021).

Actualmente, el Modelo de Autoevaluación Unadista, de acuerdo con las responsabilidades sustantivas de la institución, está organizado en términos de seis dimensiones, a saber: Formación, Investigación, Proyección Social, Inclusión, Internacionalización e Innovación, y fundado en los lineamientos para la acreditación institucional y de programas, establecidos por el CNA. Dicho modelo se ha ajustado autónomamente en factores, características, indicadores y evidencias de cumplimiento particulares para el desarrollo institucional y de programas académicos en la modalidad en educación a distancia y virtual. Al mismo tiempo, está estructurado en tres (3) componentes, según el objeto de acreditación: componente de acreditación institucional, componente de acreditación de programas de grado y

especializaciones y componente de acreditación de programas de maestrías y doctorados. Estos componentes varían en el número de sus características, indicadores y evidencias de cumplimiento (UNAD, 2011).

En la siguiente tabla se observa la cantidad de factores, características, indicadores y evidencias de cumplimiento definidos para cada objeto del modelo de Autoevaluación Unadista.

Tabla 1.

Objetos del Modelo de Autoevaluación UNAD

Objeto de Acreditación	Factores	Características	indicadores	evidencias de cumplimiento
Autoevaluación institucional	15	45	128	409
Autoevaluación de programas de grado y especialización	15	47	66	331
Autoevaluación de programas de maestría	15	44	62	260

Fuente: Autoría propia basado en el modelo de acreditación Unadista

Se puede observar que el modelo de autoevaluación Unadista está muy bien organizado y fundamentado en un marco conceptual, legal e institucional; sin embargo, su operación es compleja considerando las características, contextos de la universidad y las herramientas utilizadas para ejecutarlo.

En la actualidad, para la ejecución de ejercicios de autoevaluación se utilizan algunas aplicaciones de la suite de oficina de Google, como la herramienta Google Form para obtener la apreciación por parte de los grupos de interés; la tabulación es consolidada en hojas de Excel (Ver anexos 1 al 5), al igual que los resultados obtenidos en las diferentes fases del proceso, lo cual requiere de una minuciosa revisión y, finalmente, el almacenamiento de toda la información resultante del proceso se hace en la nube por medio de Google Drive. En consecuencia, este tipo de recursos no garantiza la integridad y seguridad de la información y genera dependencia de la cuenta de usuario que administra la información.

Dado lo anterior surge la siguiente pregunta de investigación:

Formulación del problema

¿Cómo Implementar la metodología BPM y herramientas software para ejercicios de autoevaluación institucional y de programas académicos como estrategia para el fortalecimiento de procesos de aseguramiento de la calidad en la Universidad Nacional Abierta y a Distancia - UNAD?

Objetivos

Objetivo General

Implementar la metodología BPM y una herramienta de software para la gestión del proceso de autoevaluación institucional y de programas académicos, como fortalecimiento de los procesos de aseguramiento de la calidad en la Universidad Nacional Abierta y a Distancia – UNAD.

Objetivos Específicos

Analizar el proceso de autoevaluación mediante la metodología de Gestión de Procesos de Negocios BPM.

Diseñar la Arquitectura de software para el prototipo de software del proceso de autoevaluación de la UNAD.

Desarrollar e implementar el prototipo de software como soporte del proceso de autoevaluación.

Alcance y Delimitación

Este proyecto se planteó con el fin de incorporar métodos y herramientas para automatizar y optimizar el proceso de Autoevaluación para el mejoramiento continuo de la calidad en la UNAD; por consiguiente, aplicando la metodología BPM, se espera analizar tanto en el estado actual como futuro y, a partir de este, desarrollar e implementar un prototipo de software que soporte la gestión del proceso de Autoevaluación, específicamente para la apreciación de los grupos de interés.

En este sentido, el proyecto consta de dos partes:

Implementación de la metodología BPM, a través de la cual se realizará un diagnóstico, que servirá como base para analizar el proceso de autoevaluación y establecer su estado futuro, teniendo en cuenta la incorporación de un sistema de información y aplicando la política de gobierno digital.

Desarrollo de un prototipo de software para ejercicios de autoevaluación como estrategia para el fortalecimiento de procesos de aseguramiento de la calidad.

Justificación

El artículo 67 de la Constitución Política de Colombia (CPC) establece que: “corresponde al Estado regular y ejercer la suprema inspección y vigilancia de la educación con el fin de velar por su calidad, por el cumplimiento de sus fines y por la mejor formación moral, intelectual y física de los educandos” (Const., 1991). Así mismo, buscar la calidad de las IES es una necesidad en la

sociedad del conocimiento, y es el fundamento de los procesos de acreditación de programas e institucional.

Es así como el MEN, en su decreto 1330 de 2019, establece las condiciones de calidad, dentro de las cuales se encuentra la cultura de autoevaluación, como el conjunto de mecanismos que las instituciones tienen para el seguimiento sistemático del cumplimiento de sus objetivos misionales, el análisis de las condiciones que afectan su desarrollo y las medidas para el mejoramiento continuo (Ministerio de Educación Nacional -MEN-, 2019a).

En concordancia con (CPC, 1991) y (MEN, 2019), la universidad, dentro del Modelo de Autoevaluación Unadista, mediante el procedimiento Gestión de la Autoevaluación de Programas, establece la ruta para aplicar los lineamientos y criterios para la autoevaluación de programas y servicios ofrecidos por la UNAD, dando cumplimiento a marcos normativos nacional e institucional, pero como se describió anteriormente, la operación del proceso de autoevaluación no es la más óptima, por tanto, es sustancial analizar y aplicar métodos, herramientas y tecnologías que permitan alcanzar los objetivos misionales de forma ágil.

Para subsanar la actual situación, en el plan de desarrollo institucional 2019-2023, en su proyecto No. 14 denominado Aseguramiento Interno de la Calidad Académica, tiene como objetivo: “Desarrollar estrategias para la gestión, acompañamiento, seguimiento y monitoreo de las condiciones de calidad de programas y servicios de la oferta educativa de la Universidad, en el contexto de la cultura de la autoevaluación y autorregulación”, y entre sus metas se establece realizar una aplicación para la gestión de ejercicios de autoevaluación (UNAD, 2019b).

Por tanto, incorporar el uso de métodos, modelos y tecnologías de vanguardia para la optimización de la ejecución del Modelo de Autoevaluación Unadista se considera una oportunidad para el mejoramiento organizacional.

Por otro lado, el Ministerio de las Tecnologías de la Información y comunicación (MINTIC), en su política de gobierno digital tiene por objetivo “Promover el uso y aprovechamiento de las tecnologías de la información y las comunicaciones para consolidar un Estado y ciudadanos competitivos, proactivos, e innovadores, que generen valor público en un entorno de confianza digital (MINTIC-, 2018).

Este proyecto busca optimizar el proceso de autoevaluación en la UNAD, la cual es una institución de educación superior pública y autónoma, por consiguiente, aplicando la metodología de gestión de procesos como BPM, y el aprovechamiento de las TIC, es posible modelar una solución que se logrará incorporar como acción de mejora para superar brechas en la ejecución del proceso. A su vez, con el desarrollo de un prototipo de software se da cumplimiento al plan de desarrollo institucional de la UNAD 2019-2023 y a lineamientos de estado, ya que se enmarca en la Política de Gobierno Digital del MINTIC.

De acuerdo con lo mencionado anteriormente, es pertinente el desarrollo del proyecto ya que permitirá dar solución a una necesidad de carácter institucional, el cual estará alineado con su declaración misional y, al mismo tiempo, dará cumplimiento a lo estipulado por el gobierno nacional colombiano a través del Ministerio de Educación Nacional (MEN) y del Ministerio de las Tecnologías de la Información y las Comunicaciones (MINTIC).

Marco Referencial

Este capítulo consta de dos partes: la primera trata sobre la revisión de literatura relacionada con la metodología BPM aplicada a procesos académicos y administrativos, así como a procesos de aseguramiento de calidad; la segunda corresponde al marco conceptual y teórico donde se incluyen los conceptos más relevantes asociados al desarrollo de software, BPM y calidad, entre otros inherentes al proyecto.

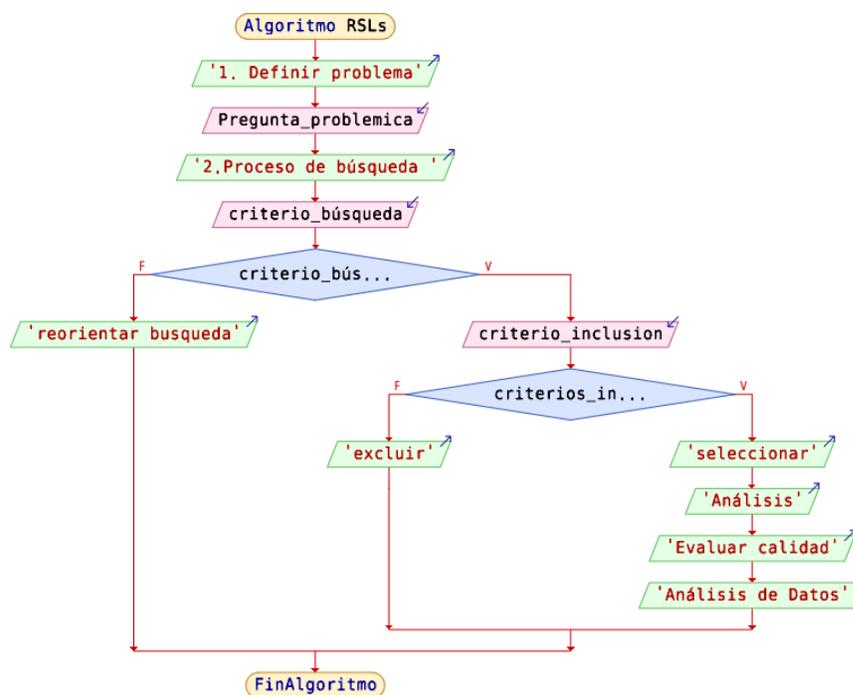
Revisión de literatura

En este apartado, inicialmente se realizó una búsqueda general sobre la implementación de BPM a procesos académicos y administrativos y para ello se empleó el diseño documental, el cual es un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos obtenidos y registrados por otros investigadores en fuentes documentales impresas, audiovisuales o electrónicas (Arias Rondon, 2012), para referentes no científicos. Luego se hizo una búsqueda más específica a través del método de revisión sistemática propuesto por (Kitchenham, 2004) con el objetivo de buscar aquellos estudios rigurosos donde se ha implementado la metodología BPM a procesos de aseguramiento de la calidad en IES. En la figura 1 se presenta el algoritmo que muestra las etapas del proceso de revisión sistemática:

- ✓ *Definición del problema*
- ✓ *Proceso de búsqueda = criterios de búsqueda, criterios de inclusión, criterios de exclusión*
- ✓ *Evaluación de calidad = Definición de categorías*
- ✓ *Análisis e interpretación*
- ✓ *Selección de artículos*

Figura 1.

Algoritmo RSL basado en la metodología de Kitchenham



Fuente: Propia

A continuación, se presentan resultados del ejercicio de revisión bibliográfica:

(Fajardo Rozo & Izquierdo Portela, 2019) diseñaron una propuesta de aplicación del enfoque de Gestión por Procesos BPM para la gestión de seguimiento a prácticas de programas de teleinformática del CEET-SENA, cuya finalidad fue detectar puntos de mejora que permitan optimizar el proceso, de tal forma que se contribuya al rendimiento, calidad e incremento de la satisfacción de los clientes. El desarrollo del proyecto estuvo basado en las primeras fases del ciclo de vida BPM: diseño y modelado, y generó grandes beneficios en el proceso de Etapa Productiva rediseñado, ya que en este se pudieron gestionar y controlar de forma eficiente las diferentes actividades que se debieron cumplir dentro del flujo de trabajo; mejorando tiempos de

respuesta en atención a los usuarios (aprendices, empresas) y entrega de informes a coordinación, de tal forma que se cumplan las metas de certificación de aprendices.

Pimenta & Abrantes (2018), utilizando la notación de modelado de procesos de negocios, BPMN, como herramienta de gestión de procesos, desarrollaron un modelo de flujo de información para el proceso de internacionalización, como herramienta de gestión para asistir el trabajo de la recepción y programa de bienvenida a los estudiantes internacionales en la Universidad Federal de Minas Gerais (UFMG), permitiendo así que ellos pudieran conocer el procedimiento de intercambio. De esta manera, a través de la modelización de la información, se logró contribuir al proceso de internacionalización.

(Cevallos Tito et al., 2018) mediante el enfoque BPM analizaron los procesos académicos y administrativos del Instituto de Postgrado de la Universidad Técnica del Norte, que no se encontraban dentro del sistema informático, con el objetivo de desarrollar una herramienta informática utilizando SCRUM y RUP como metodologías de desarrollo. En los resultados obtenidos se observó que la integración de la librería BPMN-JS en el sistema informático integrado de la UTN permitió identificar, modelar y determinar las actividades y procedimientos dentro de un modelo de gestión de procesos y se dio a conocer al personal administrativo, docente y estudiantes del Instituto de Posgrado de una manera ágil.

(Peña Hidalgo & López Ortiz, 2018) analizan el proceso de gestión de la oferta educativa en sus procedimientos: nuevo registro calificado y renovación de registro (p-8-1 y p-8-3), para iniciar lo que posteriormente fue un diseño arquitectónico que cubrió todas las instancias y requerimientos de la Escuela de Ciencias Básicas Tecnología e Ingeniería, ECBTI, utilizando el framework de trabajo TOGAF, abordando las fases: preliminar, visión de arquitectura, arquitectura de negocios, arquitecturas de sistemas de información y arquitectura de tecnología,

para adecuar una estrategia tecnológica de información que permitiera fortalecer el área de tecnología de información y que sea un activo de la Escuela, capaz de responder a una iniciativa de negocio competitiva y exitosa y a su vez tener un adecuado control, seguridad y disponibilidad permanente de la información

(Ferruzola Gómez et al., 2017) analizaron el beneficio de describir y modelar procesos administrativos que se presentan en la educación superior, específicamente el proceso de prácticas profesionales (PPP) utilizando la metodología de gestión de proceso BPM, y la notación gráfica BPMN para modelar procesos y apoyados en la herramienta WebRatio, se concluyó que los estudiantes pierden gran cantidad de tiempo para desarrollar el PPP, por lo cual es fundamental el desarrollo de un software que permita acelerar esta fase por medio del cual se logrará integrar las actividades por medio de una página web, lo que facilitará el desarrollo de los procedimientos para cumplir el requisito de forma íntegra para el PPP.

FLEACĂ, presentó un estudio basado en métodos útiles de gestión de calidad y reingeniería de procesos de negocios, en el que propuso un modelo organizativo coherente para la estrategia de una institución de educación superior, incorporando áreas clave como movilidad internacional para estudiantes y personal, internacionalización y mejora de planes de estudio y nuevos métodos de aprendizaje electrónico, y Cooperación estratégica, asociaciones y desarrollo de capacidades con actores internacionales. Los hallazgos destacan los beneficios de un enfoque holístico al garantizar la gobernanza de la estrategia de internacionalización compuesta por tres grupos de procesos centrales (académico, de investigación y relaciones comerciales más amplias) a fin de proporcionar que la arquitectura del proceso pueda ofrecer una oferta de educación superior internacionalizada de valor agregado (Fleacă, 2017).

(Congacha Aushay et al., 2017) desarrollaron una plataforma informática que da soporte a la gestión de servicios académicos (GSA), como el de las Prácticas Pre-profesionales (PPP) de la Carrera de Sistemas y Computación, pero para lograr dicho objetivo sistematizaron el proceso de selección de la herramienta informática requerida, modelaron el proceso de servicio, validaron el desempeño del modelo y automatizaron el proceso de gestión de las PPP. Se identificó a Bizagi Studio como la tecnología apropiada por su funcionalidad, escalabilidad y capacidad de integración. Los resultados muestran que la metodología propuesta permitió desarrollar una plataforma eficiente y eficaz para la gestión de las PPP.

Si bien los anteriores estudios no están orientados a procesos de aseguramiento de la calidad en instituciones de educación superior, si a procesos académicos y administrativos de ellas, donde se aplica o implementa la metodología BPM, de la mano de la tecnología, por tanto, permiten inferir que su implementación es un medio para alcanzar los objetivos esperados.

Los siguientes estudios que se presentan ya están orientados a la implementación de BPM a procesos de aseguramiento de la calidad en IES.

(Duro & Gilart, 2016), basados en otros estudios, sugieren que para proporcionar una educación de calidad es necesario centrarse en la cadena de valor de servicio de la educación superior; esto permite identificar cualquier vacío y cuello de botella para ser eliminados, de igual manera el aprovechamiento óptimo de sus recursos. Destaca también que la tecnología juega un papel importante, puesto que ayuda a conocer todos los factores que proporcionan valor con fines de lucro autofinanciados o con financiación externa, pero para llegar a ese objetivo es necesario centrarse en los procesos; por tanto, analiza diferentes filosofías de gestión, pero destacando que BPM, que es muy utilizada actualmente por IES en diferentes procesos, entre ellos los de calidad mediante el uso de herramientas software, como es el caso de la Universidad de Deusto en

España que usa una plataforma de gestión de procesos y permitió la implementación de un sistema de garantía interna de calidad (SGIC) que es de obligatorio cumplimiento en Europa.

Estos dos autores, destacan claramente que la calidad en la educación se logra cuando la institución se centra en la cadena de valor, es decir, se divide en sus actividades estratégicas más relevantes, y entre esas actividades se encuentra el proceso de gestión de calidad. Aunque hay diversas metodologías, demuestran que existen varios estudios que han implementado la gestión de procesos BPM junto con el apoyo de la tecnología a procesos de calidad en la educación.

(Flores et al., 2014) también recomiendan el enfoque BPM junto con las Tecnologías de la Información y Comunicación (TIC). Ellos presentan un estudio realizado en la Universidad Técnica de Ambato – UTA en Ecuador para mejorar su modelo de gestión, a través de la automatización del proceso de recolección de datos y evidencias para la evaluación de carreras, determinadas por el Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEEACES), para mejorar los niveles de calidad y reducir los tiempos del proceso. Finalmente demostraron que la adopción de BPM y los sistemas BPM como soporte tecnológico facilitaron la automatización de los procesos, permitiendo integrar las políticas institucionales, el personal administrativo y la infraestructura tecnológica disponible.

(Calle, Mayorga, Flores, & Lavín, 2014) establecen que para obtener el logro de los objetivos esperados al aplicar BPM en los procesos, es necesario una adecuada metodología que conlleve a la simplificación de los procesos, y no a automatizar con deficiencias, por tanto, proponen la metodología BPM RAD para el proceso de recolección de evidencias para la evaluación y acreditación de las carreras en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial en la UTA. Adicionalmente, presentan una herramienta software donde tuvieron en cuenta las fases del ciclo de vida BPM como son: diseñar, modelar, ejecutar, monitorear y

optimizar; para las dos primeras fases se utilizó BPM-RAD junto con la herramienta Bonita BPM, que permitió continuar con las siguientes fases.

En esta segunda parte de referentes, se observa que en las instituciones mencionadas dentro de sus procesos de aseguramientos de calidad como es la recolección de datos y evidencias para la evaluación de programas, fueron automatizados mediante la combinación de metodología BPM y herramientas de software, con el fin de optimizarlos y llegar al logro de sus objetivos conforme a su cadena de valor, por lo cual es una estrategia enfocada a la calidad.

Marco Conceptual y Teórico

Este punto busca conocer el conjunto de conceptos más relevantes que enmarcan esta investigación, así como las teorías que lo sustentan, para comprender rigurosamente las bases teóricas que soportan los procesos y los métodos, modelos y herramientas a usar para su optimización, por lo cual se abordaran conceptos como autoevaluación, gestión de procesos de negocio BPM, calidad, aseguramiento de la calidad, desarrollo de software, otros.

Tabla 2.

Marco Conceptual

Concepto	Autor	Análisis
Arquitectura de Software	(Blancarte Iturralde, 2020)	El diseño de más alto nivel de la estructura de un sistema, el cual consiste en un conjunto de patrones y abstracciones que proporcionan un marco claro para la implementación del sistema.

Business Process Management (BPM)	(Garimella et al., 2011) (Camargo et al., 2010)	Los autores presentan tres conceptos de BPM de los cuales se puede establecer que mediante la metodología BPM es posible gestionar los procesos de una empresa permitiendo así el mejoramiento continuo de la organización y para cumplir eficientemente sus metas y objetivos ya que ofrece los siguientes beneficios.
Beneficios de BPM	(Cetina Riaño, 2016) citando a (Muñoz, (s.f.).	Los beneficios del enfoque BPM están orientados a aumentar la productividad, rapidez en la comercialización, optimización de las operaciones, cumplir con normativas y al mismo tiempo reducir costes, hacer que la innovación sea parte del proceso.
Calidad	(CNA,2013)	Hace referencia a la síntesis de características que permiten reconocer un programa académico específico o una institución de determinado tipo y hacer un juicio sobre la distancia relativa entre el modo como en esa institución o en ese programa académico se presta dicho servicio y el óptimo que corresponde a su naturaleza.
Sistema de Aseguramiento de la Calidad de	(Ministerio de Educación Nacional - MEN-, 2019b)	Sus principales objetivos van orientados a que las instituciones de educación superior rindan cuentas ante la sociedad y el Estado sobre el

la Educación**Superior**

servicio educativo que prestan, provean información confiable a los usuarios del servicio educativo y se propicie el auto examen permanente de instituciones y programas académicos en el contexto de una cultura de la evaluación. El Sistema de Aseguramiento de la Calidad de la Educación Superior está conformado por tres componentes relacionados entre sí: información, evaluación y fomento.

Metodología

El presente proyecto corresponde a una investigación de tipo aplicada; la Organización para la Cooperación y el Desarrollo Económicos (OCDE) la define de la siguiente manera:

Se emprende para determinar los posibles usos de los resultados de la investigación básica, o para determinar nuevos métodos o formas de alcanzar objetivos específicos predeterminados. Este tipo de investigación implica la consideración de todos los conocimientos existentes y su profundización, en un intento de solucionar problemas específicos. (OCDE, 2013, p.90).

Teniendo en cuenta el alcance del proyecto, para llegar al objetivo general y lograr la optimización del proceso de autoevaluación, este se desarrolló haciendo uso de múltiples metodologías según el objetivo a alcanzar en cada fase, es así como se implementa BPM, para el análisis y modelización del proceso, Arquitectura de software bajo del enfoque de puntos de vista para la estructura del sistema y Scrum, como marco de desarrollo para el prototipo de software. A continuación, en la figura 2 se pueden apreciar estas fases con cada una de sus etapas y seguidamente se encontrará la descripción de cada una de ellas:

Figura 2.*Fases del proyecto***Fase 1 Análisis del proceso mediante la metodología de Gestión de Procesos de Negocios****BPM**

Para analizar el proceso de Autoevaluación se usará BPM como Metodología empresarial, cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua.

Para esta fase del proyecto se utilizaron las etapas del ciclo de vida del BPM descritas por Rodríguez, Gutierrez y Lasso (2020), como son: identificación del proceso, descubrimiento del proceso, análisis del estado actual del proceso, rediseño del proceso, implementación del proceso, monitoreo y control del proceso. Sin embargo, como se mencionó anteriormente solo se tomarán las correspondientes para el análisis del proceso las cuales se describen a continuación:

Identificación del Proceso.

Se hace el reconocimiento del proceso crítico en la institución, que para este caso es el procedimiento de autoevaluación de programas académicos e institucional, mostrando como resultado una visión general.

Descubrimiento del Proceso.

Se conoce de manera más detallada la funcionalidad del proceso, sus actores, flujo de actividades, indicadores, normas y reglas del negocio.

Análisis del Proceso.

Busca entender y evaluar el proceso tal cual como se lleva a cabo en la actualidad o AS-IS, en notación BPMN, incluyendo un análisis cualitativo y cuantitativo, que evidencia la situación problema del proceso.

Rediseño del Proceso.

Se realiza luego de que el proceso esté documentado y analizado, para plantear opciones de oportunidades y mejoras que permitan impactar positivamente en los resultados.

Fase 2 Diseño de la arquitectura de software para el sistema de autoevaluación

Para el diseño del prototipo de software, que permite la automatización del proceso de autoevaluación de la UNAD, se analizó el tipo de aplicación más conveniente y determinó la arquitectura correspondiente, entre varias candidatas, para luego diseñar la arquitectura de solución bajo la perspectiva de puntos de vista planteado por (Rozanski, Woods Marks, & Plc, 2011), para mostrar propiedades específicas, ya que cada uno representa un comportamiento en particular. Esta fase contó con las siguientes etapas:

- ✓ *Selección del tipo de aplicación a desarrollar*
- ✓ *Revisión de arquitecturas candidatas*
- ✓ *Selección de la arquitectura del sistema*

- ✓ *Selección de los puntos de vista y diseño de los artefactos*

Fase 3 Desarrollo del sistema mediante la metodología del proceso de desarrollo ágil Scrum

La metodología de desarrollo utilizada en el proceso de construcción de software para el proceso de autoevaluación Unadista fue Scrum; marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente (Schwaber & Sutherland, 2020).

Esta fase se conformó de las siguientes etapas:

- ✓ Determinar los requerimientos del sistema e historias de usuario
- ✓ Preparación de ambiente de desarrollo
- ✓ Ejecución de las iteraciones (Sprints, Proceso cíclico)
- ✓ Despliegue en ambiente de desarrollo en la nube
- ✓ Pruebas del prototipo de Software
- ✓ Entrega del prototipo de software

Análisis del proceso mediante la metodología de Gestión de Procesos BPM

En este capítulo se describe la aplicación de la metodología BPM al proceso Gestión de la autoevaluación de programas; para ello primero se necesitó, conocer en forma general del proceso, seguidamente pasando de lo general a lo específico se lleva a un nivel de detalle, y así identificar las actividades que lo componen, los actores que intervienen en ellas, y todo lo relacionado con su funcionalidad; posteriormente se modeló su estado actual, a través del modelo AS-IS y finalmente se estableció una propuesta de mejora a través del modelo TO-BE.

Identificación del procedimiento Gestión de la autoevaluación de programas

En la figura 3 se puede observar una descripción general del proceso de Gestión e la autoevaluación de programas:

Figura 3.

Identificación procedimiento Gestión de la autoevaluación de programas. Fuente: (UNAD, 2021)

1. Información General del Procedimiento	
1.1. Unidad Responsable:	VISAE
1.2. Objetivo:	Establecer la ruta para aplicar los lineamientos y criterios para la autoevaluación de programas y servicios ofrecidos por la UNAD en sus sistemas de oferta educativa, en coherencia con lo dispuesto en el Proyecto Académico Pedagógico Solidario (PAPS), utilizando los resultados como fuente de información para la toma de decisiones, la autogestión y la autorregulación, en torno a la cultura de la calidad.
1.3. Alcance:	Aplica a toda la oferta educativa vigente del meta sistema Unadista; involucra el análisis de la información de percepción de los grupos de interés, la valoración de las evidencias de cumplimiento del modelo de autoevaluación unadista, y la formulación y ejecución de acciones de mejora; en el marco del modelo de autoevaluación Unadista (antes modelo de acreditación Unadista).

Actividades del procedimiento en su estado actual

A continuación, se presentan las actividades del proceso en su estado actual:

1. *Programar la aplicación de los instrumentos de recolección de información correspondientes*
2. *Consolidar los resultados de la aplicación de los ejercicios de consulta en el marco de lo establecido en el modelo de autoevaluación.*
3. *Consolidar evidencias de cumplimiento*
4. *Analizar la información*
5. *Elaborar el informe de autoevaluación*
6. *Revisar el informe de autoevaluación por parte del comité respectivo del sistema al que pertenece el programa*
7. *Revisar el informe de autoevaluación por parte de VISAE VIACI, y la GCMO*
8. *Registrar el Plan de mejoramiento en el SSAM*
9. *Articular el plan de mejoramiento con el plan operativo de la unidad, para el siguiente año*
10. *Verificar el cierre de acciones del plan de mejoramiento del programa para desarrollar un nuevo ejercicio de autoevaluación*

Descubrimiento del proceso

Después de identificar el proceso Gestión de la Autoevaluación de Programas, a continuación, se procede a identificar sus actores y analizar en detalle el proceso.

Gestión de los interesados

En esta sección, se logra identificar a las personas, grupos, y entidades que de forma interna o externa son afectados o afectan directa o indirectamente el sistema de aseguramiento de la calidad de la educación superior, en ese sentido el Modelo de Autoevaluación Unadista Lo que a su vez involucra el proyecto para gestionar procesos de autoevaluación en la UNAD.

Tabla 3.*Matriz de los Interesados del Proyecto de Autoevaluación*

Nombre interesado	Rol dentro del proceso	Interno	Externo
Vicerrectoría de Servicios Aspirantes, Estudiantes y Egresados VISAE	<p>La Vicerrectoría de Servicios Aspirantes, Estudiantes y Egresados es un Componente estructural del meta sistema UNAD, de carácter misional, que gestiona y coadyuva al aseguramiento de la calidad del servicio Unadista a aspirantes, estudiantes y egresados, a través de la valoración del impacto, oportunidad, pertinencia y calidad de los programas y servicios derivados de las ofertas de las unidades misionales, mediante procesos de autoevaluación permanente, autorregulación y mejoramiento continuo, que promuevan la cultura del buen servicio y el bienestar integral, para afianzar la retención, permanencia y satisfacción, tanto estudiantil como de sus egresados, a través de la interacción permanente con las redes estamentarias correspondientes.</p>	Interno	

Comité de Acreditación y Calidad del Programa, CACP	Es la instancia responsable del desarrollo del proceso de autoevaluación del Programa.	Interno
Líder de Aseguramiento de la Calidad	Cómo líder de calidad y representante del CACP es la persona responsable de los procesos de autoevaluación institucional y de los programas.	Interno
Líder de programa	Responsable de seleccionar y convocar a los integrantes del Comité Curricular del Programa (CCP) y de gestionar la autoevaluación del programa a su cargo, enviar instrumentos a los diferentes estamentos y gestionar la evaluación documental del programa.	Interno
Comité curricular de programa	Son docentes del programa y tienen la responsabilidad de ponderar características y evaluar la evidencia documental del programa.	Interno

Estudiantes	A través del diligenciamiento de instrumento de levantamiento de información plantean su opinión sobre la calidad de un programa académico o la institución.	Interno
Docentes	A través del diligenciamiento de instrumento de levantamiento de información plantean su opinión sobre la calidad de un programa académico o la institución.	
Egresados	A través del diligenciamiento de instrumento de levantamiento de información plantean su opinión sobre la calidad de un programa académico o la institución.	Externo
Administrativos	A través del diligenciamiento de instrumento de levantamiento de información plantean su opinión sobre la calidad de un programa académico o la institución.	Interno
Empleadores	A través del diligenciamiento de instrumento de levantamiento de información plantean su opinión sobre la calidad de un programa académico o la institución.	Externo

Decanos de escuela	Encargado de liderar la renovación del registro calificado de los programas, creación de nuevos programas, hacen Seguimiento al proceso de autoevaluación del programa y ejecución del plan de mejoramiento.	Interno
Pares Académicos	Son el soporte fundamental para el proceso de acreditación. Par significa igual o semejante totalmente. En sentido estricto, el par es semejante por cuanto puede ser reconocido por los miembros de la comunidad como uno de los suyos.	Externo
Consejo Nacional de Acreditación (CNA)	Emitir concepto técnico sobre la calidad de los programas académicos. Con respeto a la autoevaluación no intervienen en el proceso solo se observa el resultado de la autoevaluación.	Externo
Consejo Nacional de Educación Superior -CESU	organismo con funciones de planificación, asesoría, coordinación y recomendación en el nivel de educación superior que apoya al Ministerio de Educación Nacional en la consecución de los fines y propósitos del Sistema de Aseguramiento de la Calidad.	Externo

Comisión Nacional Intersectorial de Aseguramiento de la Calidad de la Educación Superior - CONACES	Órgano que se encarga de la evaluación del cumplimiento de los requisitos para la creación de instituciones de educación superior	Externo
Ministerio de Ciencia, Tecnología e Innovación -CTel	Formula y articula la política pública para la generación de conocimiento, la innovación, la apropiación social y la competitividad.	Externo
Instituto Colombiano para la Evaluación de la Educación -ICFES	Entidad vinculada al Ministerio de Educación Nacional y ofrece servicios de evaluación de la educación en todos sus niveles (Exámenes de Estado), adelanta investigación sobre los factores que inciden en la calidad de la educación. brinda información que contribuye al mejoramiento y la toma de decisiones en la calidad de la educación	Externo

Instituto		
Colombiano de Crédito Educativo y Estudios Técnicos en Exterior "Mariano Ospina Pérez" -ICETEX	Entidad del Estado que promueve la Educación Superior a través del otorgamiento de créditos educativos y su recaudo, con recursos propios o de terceros, a la población con menores posibilidades económicas y buen desempeño académico.	Externo
Ministerio de Salud y Protección Social	Entidad pública del nivel central del Gobierno Nacional y cabeza del sector salud, encargada de conocer, dirigir, evaluar y orientar el sistema de seguridad social en salud, mediante la formulación de políticas, planes y programas, la coordinación intersectorial y la articulación de actores de salud con el fin de mejorar la calidad, oportunidad, accesibilidad de los servicios de salud y sostenibilidad del sistema, incrementando los niveles de satisfacción de los pacientes, familias, comunidades y habitantes del territorio nacional.	Externo

Comisión Intersectorial de Humano en Salud - CIHS	Órgano responsable de la toma de decisiones derivadas de las funciones públicas relacionadas con la formación, el ejercicio y el desempeño del talento humano en salud que requieran acciones conjuntas de los Ministerios de la Protección Social y de Educación Nacional.	Externo
--	---	---------

Fuente: Elaboración propia

Tabla 4.*Clasificación de los Interesados del Proyecto de Autoevaluación*

Interesado	Indiferente	Resistente	Neutral	De apoyo	Líder
Vicerrectoría de Servicios Aspirantes, Estudiantes y Egresados VISAE					AD
Comité de Acreditación y Calidad del Programa -CACP					AD
Líder de Aseguramiento de la Calidad					AD
Líder de programa					AD
Comité curricular de programa				A	D
Decanos de escuelas					AD
Estudiantes			A	D	
Docentes				AD	
Egresados			A	D	
Empleadores			A	D	
Consejo Nacional de Acreditación			A		

A: Actual - D: Deseado

Fuente: Elaboración propia

Caracterización y descripción de los subprocesos del proceso de autoevaluación

Dentro del proceso de autoevaluación se identificaron los subprocesos: preparación de insumos, diseño y aplicación de instrumentos, recopilación documental y verificación de cumplimiento e Informe final y plan de mejoramiento, estos se describen a continuación:

Tabla 5.*Subproceso 1. Preparación de insumos*

Subproceso 1. Preparación de insumos	
Actividades	Actores
1.1 Ponderación de características de autoevaluación.	Comité curricular de programa
1.2 Generar históricos de autoevaluación	Unidades de apoyo
1.3 Determinar población y muestra por cada grupo de interés.	Líder nacional de programa $CF = \sum_{i=1}^n (Ci * PC)$
Documentos generados	Acta de comité curricular Ficha de seguimiento y muestreo

Tabla 6.*Subproceso Diseño y aplicación de instrumentos*

Subproceso 2. Diseño y aplicación de instrumentos	
Actividades	Actores
2.1 Diseño de instrumentos	VISAE
2.2 Envío de instrumentos	Red de líderes zonales
2.3 Aplicación de instrumentos	Grupos de interés
2.4 Tabulación de resultados	VISAE

2.5 Apreciación de grupos de interés	Líder nacional de programa
Documentos generados	Tabla de resultados por grupos de interés Matriz apreciación grupos de interés
Fuente: Elaboración propia	

Tabla 7.

Subproceso Recopilación documental y verificación de cumplimiento

Subproceso 3. Recopilación documental y verificación de cumplimiento	
Actividades	Actores
3.1 Recopilación de evidencia documental del programa e institución	Líder Nacional de Programa
3.2 Clasificación de los tipos de documentos	Comité curricular
3.3 Relación del documento con la característica.	Comité curricular
3.4 Calificación del documento	Comité curricular
3.5 Determinación del cumplimiento del documento	Comité curricular
Documentos generados	Matriz de evaluación documental Cuadro de calificación valoración documental
Fuente: Elaboración propia	

Tabla 8.*Informe de autoevaluación y plan de mejoramiento*

Subproceso 4. Informe de autoevaluación y plan de mejoramiento	
Actividades	Actores
4.1 Consolidación de resultados	VISAE
4.2 Elaboración de informe de autoevaluación	Líder de programa
4.3 Formular plan de mejoramiento	Líder de programa
4.4 Formalización del plan de mejoramiento en el S-SAM	Decano
4.5 Ejecutar plan de mejoramiento.	Decano
Documentos generados	Matriz resultado de autoevaluación Informe de autoevaluación Plan de mejoramiento

Análisis del proceso de autoevaluación estado actual AS-IS

Para el modelamiento del proceso de autoevaluación se utilizó la notación de modelo de procesos de negocio BPMN (Business Process Model and Notation,) mediante flujos de trabajo (workflow), coordina la secuencia lógica de los procesos y mensajes que fluyen entre los actores y las actividades de un proceso de negocio. (NEXTECH, 2020). Teniendo en cuenta esta notación a continuación se presentan los diagramas AS-IS:

Diagrama 1 Modelo del Proceso de Autoevaluación AS-IS

Las actividades del procedimiento en su estado actual son:

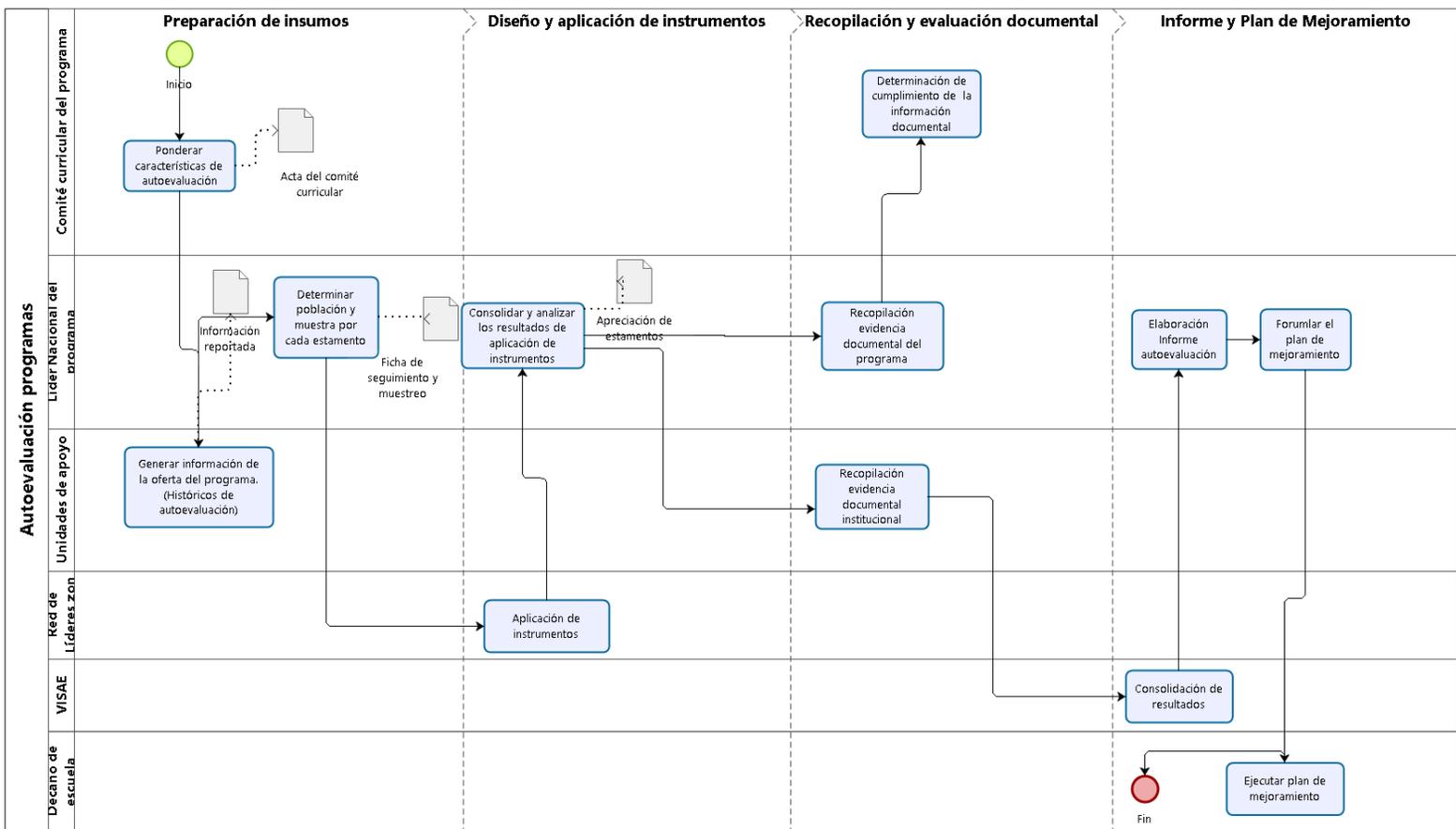
Inicio

- ✓ *El comité curricular pondera las características de autoevaluación y se genera el acta del comité curricular*
- ✓ *El líder nacional del programa determina la población y muestra de cada estamento generando la ficha de seguimiento y muestreo*
- ✓ *El líder zonal envía los instrumentos para ser aplicados por los estamentos*
- ✓ *El líder nacional del programa consolida y analiza los resultados de la aplicación de instrumentos y se genera el documento apreciación de estamentos*
- ✓ *El líder nacional del programa recopila la evidencia documental del programa*
- ✓ *El comité curricular del programa determina el cumplimiento de la evaluación documental*
- ✓ *VISAE consolida los resultados*
- ✓ *El líder nacional del programa elabora el informe de autoevaluación y formula el plan de mejoramiento*
- ✓ *El decano (a) de escuela ejecuta el plan de mejoramiento*

Fin

Figura 4.

Diagrama AS-IS del proceso de Autoevaluación



Análisis de los subprocesos del proceso de autoevaluación estado actual AS-IS

Después de presentar en forma general el diagrama del proceso de autoevaluación, a continuación, se detalla cada subproceso:

Diagrama 2 Modelo Preparación de insumos.

El subproceso de preparación de insumos se compone de las siguientes actividades:

Inicio

- ✓ *El comité curricular pondera las características de autoevaluación y se genera el acta del comité curricular*
- ✓ *Las unidades de apoyo generan información de la oferta de los programas*
- ✓ *Las unidades de apoyo entregan los datos discriminados*
- ✓ *El líder nacional del programa revisa los históricos de autoevaluación*
- ✓ *El líder nacional del programa determina la población y muestra de cada estamento generando la ficha de seguimiento y muestreo*

Fin

Figura 5.

Diagrama AS-IS Preparación de Insumos

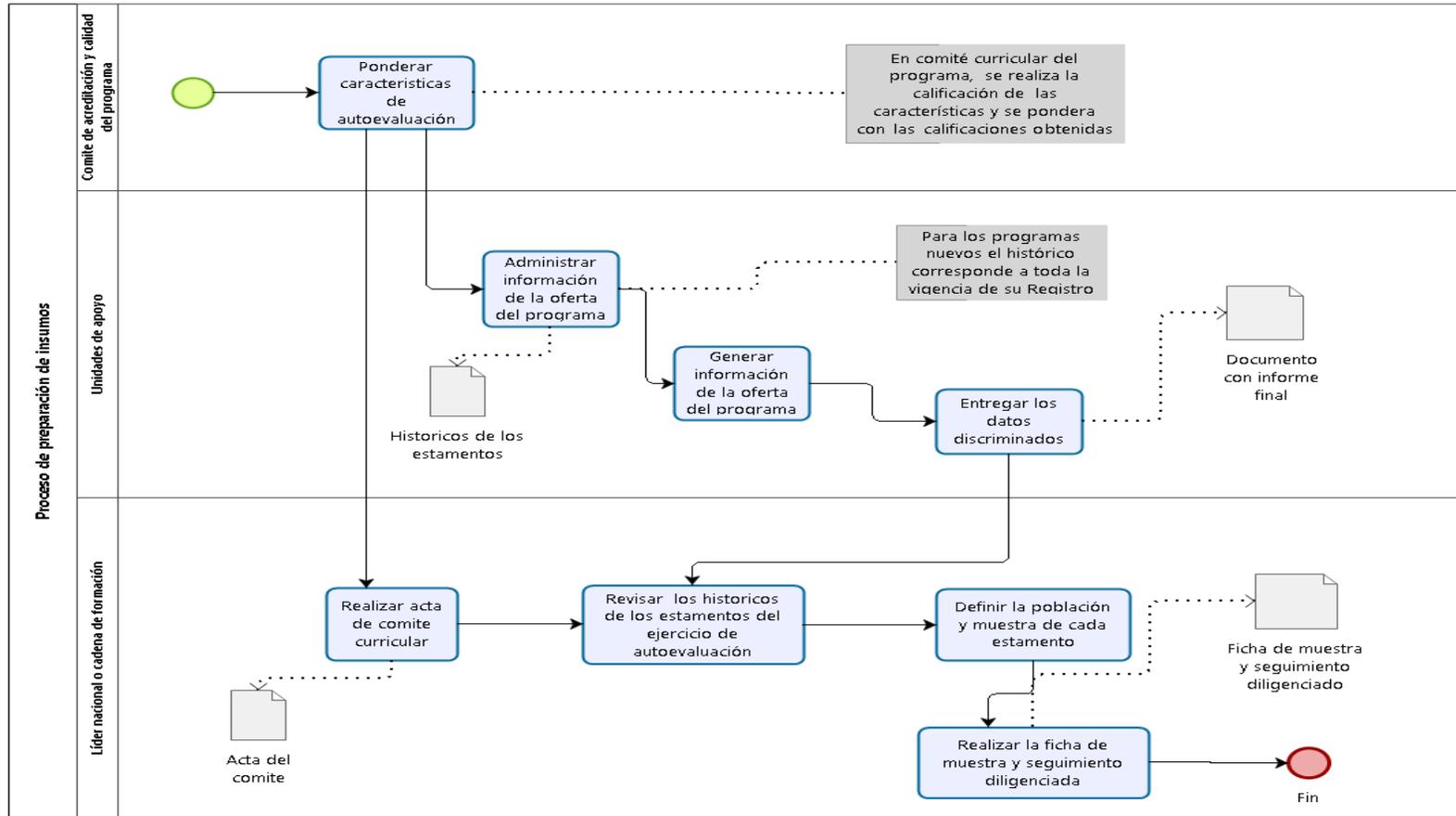


Diagrama 3 Subproceso Diseño y aplicación de instrumentos

El subproceso diseño, aplicación y tabulación de instrumentos se compone de las siguientes actividades:

Inicio

- ✓ *El líder de calidad de VISAE diseña los instrumentos para cada grupo de interés en Google Forms*
- ✓ *El líder de calidad de VISAE envía los enlaces de los instrumentos a al líder nacional de programa*
- ✓ *El líder nacional de programa reenvía los enlaces a los líderes zonales*
- ✓ *Los líderes zonales envían los enlaces de los instrumentos para ser aplicados por los estamentos*
- ✓ *Los estamentos o grupos de interés dar respuesta a los instrumentos*
- ✓ *El líder zonal envía los instrumentos para ser aplicados por los estamentos*
- ✓ *El líder de calidad de VISAE tabula y consolida las respuestas por grupo de interés.*
- ✓ *El líder nacional del programa consolida y analiza los resultados de la aplicación de instrumentos y se genera el documento matriz apreciación de estamentos.*

Fin

Figura 6.

Diagrama AS-IS Diseño Aplicación y Tabulación de Instrumentos

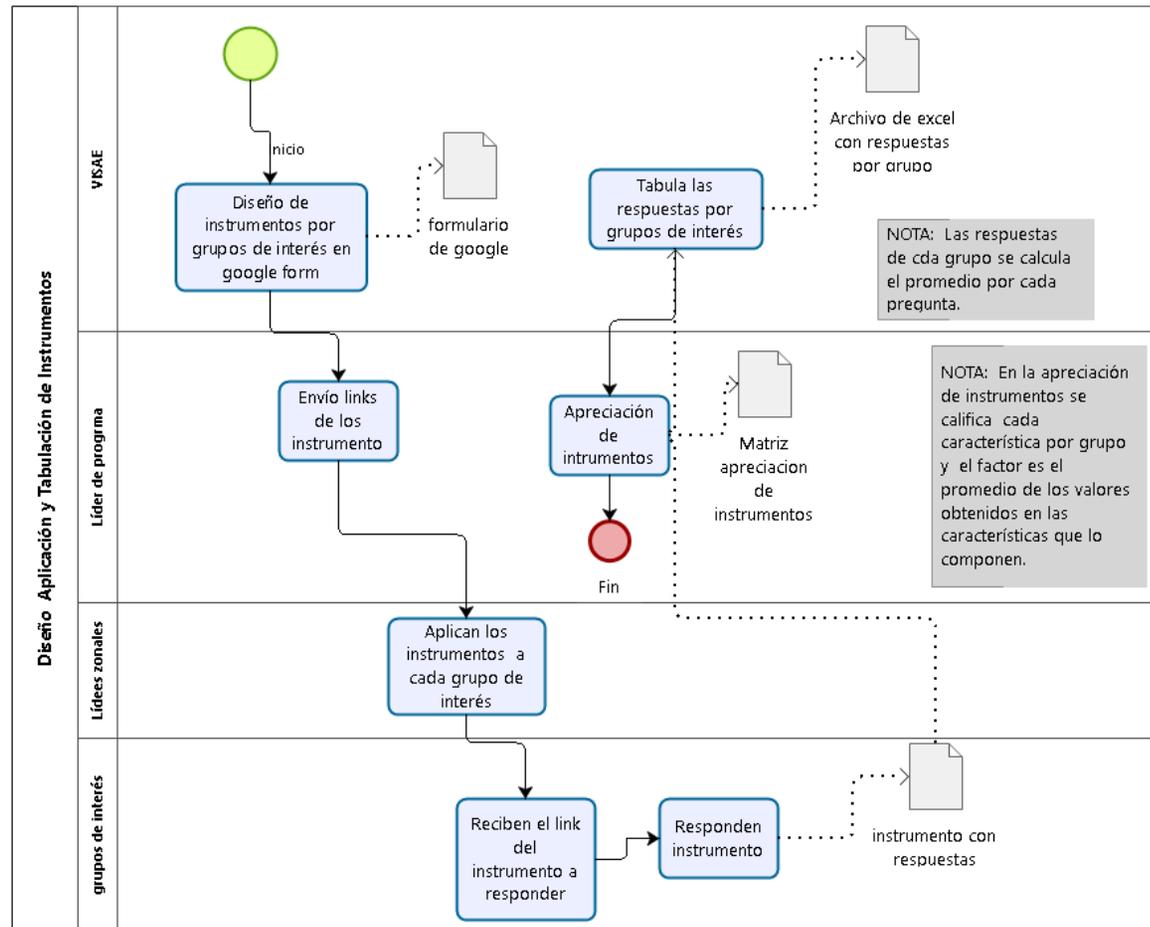


Diagrama 4 Subproceso Recopilación documental y verificación de cumplimiento

El subproceso recopilación documental y verificación de cumplimiento se compone de las siguientes actividades:

Inicio

- ✓ *Las unidades misionales recopilan la evidencia documental de los factores y características institucional*
- ✓ *El líder nacional del programa recopila la evidencia documental de los factores y características*
- ✓ *El comité curricular del programa determina el grado de cumplimiento de la información documental frente a los criterios de autoevaluación y se genera el formato de valoración documental*
- ✓ *VISAE genera la tabla de consolidación de resultados*

Fin

Figura 7.

Diagrama AS-IS Recopilación Documental y verificación de Cumplimiento

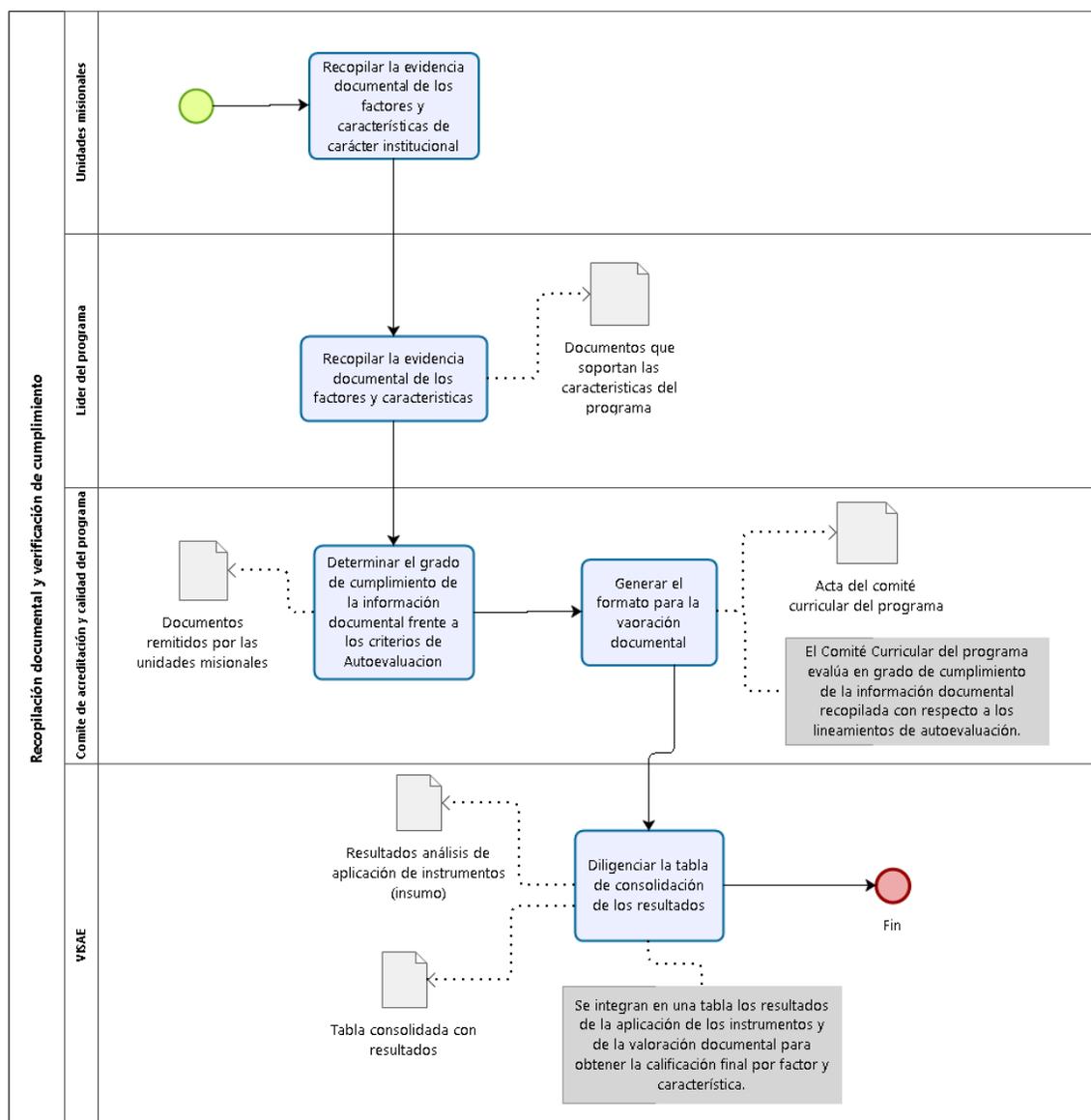


Diagrama 5 Informe de autoevaluación y plan de mejoramiento

El subproceso informe de autoevaluación y plan de mejoramiento se compone de las siguientes actividades:

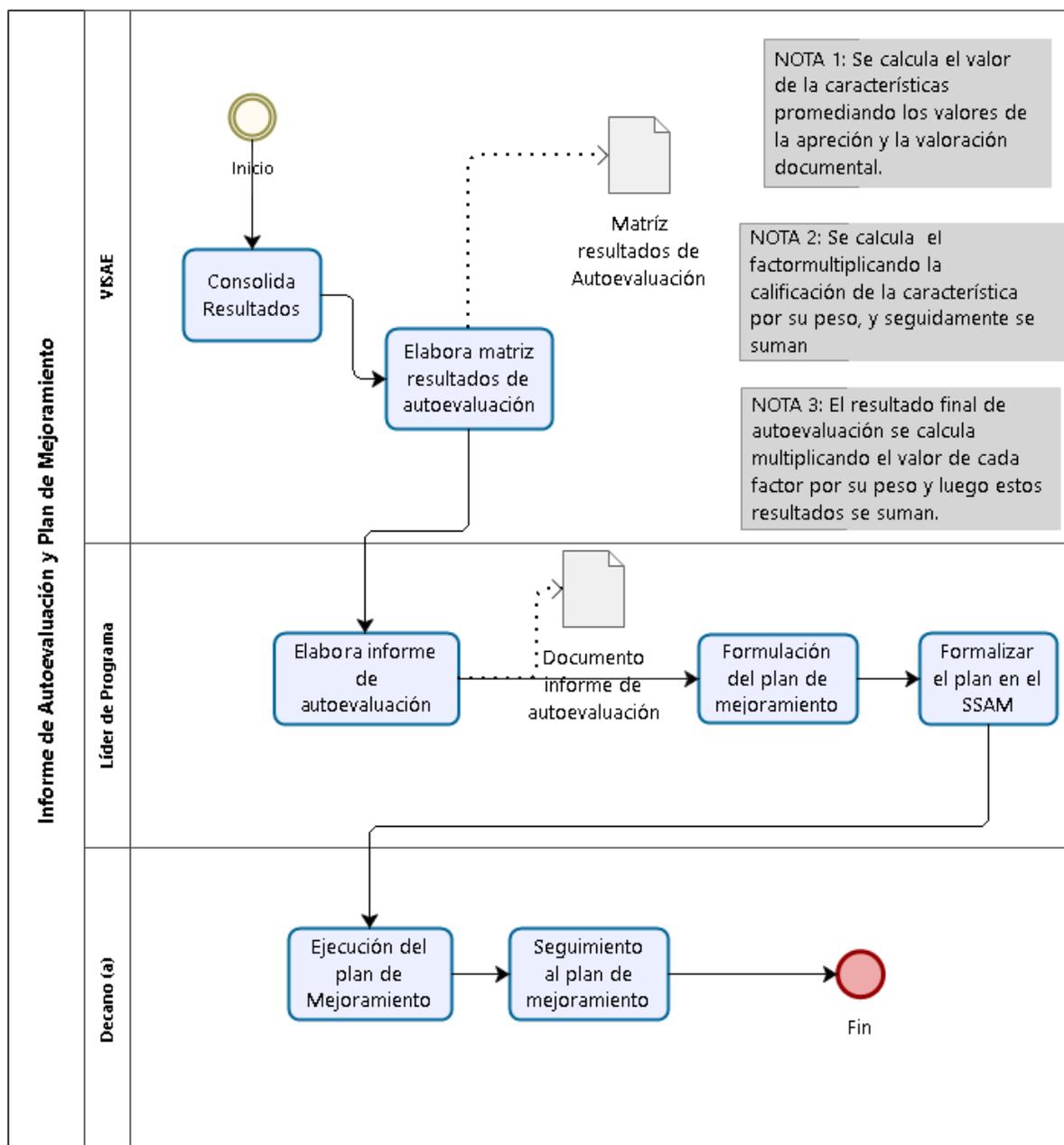
Inicio

- ✓ *VISAE consolida los resultados y elabora la matriz resultados de autoevaluación*
- ✓ *El líder nacional del programa elabora el informe de autoevaluación y formula el plan de mejoramiento*
- ✓ *El líder nacional del programa formaliza en el sistema SSAM el plan de mejoramiento*
- ✓ *El decano (a) de escuela ejecuta el plan de mejoramiento*
- ✓ *El decano (a) de escuela hace seguimiento al plan de mejoramiento*

Fin

Figura 8.

Diagrama AS-IS Informe de Autoevaluación



Al representar gráficamente el estado actual del proceso de autoevaluación se encontraron los siguientes hallazgos que demuestran su complejidad y demora:

- *La ponderación de las características no queda registrada en ningún espacio, este archivo es guardado en una carpeta de drive de una cuenta institucional, se requiere que se realice de forma automática dejando registro de la operación.*
- *En cada proceso de autoevaluación toca solicitar a las unidades de apoyo los históricos de autoevaluaciones anteriores.*
- *El muestreo se hace con una tabla de Excel tomando los datos que proporciona RyC con respecto al tamaño de la población de un determinado programa en la misma tabla se hace el seguimiento.*
- *Los instrumentos se elaboran en formularios de Google desde una cuenta institucional*
- *Los líderes zonales envían los instrumentos a los correos institucionales de los grupos de interés en su zona*
- *Los grupos de interés de la zona dan respuesta al instrumento recibido, pero no hay forma de verificar quien realmente da respuesta.*
- *La tabulación de las respuestas es tediosa, porque se bajan las respuestas luego en otra tabla por pregunta se coloca las respuestas y así obtener un promedio de la respuesta.*
- *En otra hoja de Excel con los resultados tabulados obtiene la apreciación de cada característica por grupo de interés.*
- *Para la recopilación documental no existe un espacio donde se pueda obtener dicha información.*
- *La información institucional tiende a duplicarse porque es la misma para todos los programas.*
- *Aunque existe una clasificación institucional para los documentos hay que revisar constantemente esa información para así clasificar los documentos.*
- *La relación del documento se hace en una basta hoja de Excel y resulta tedioso lento el proceso*
- *La hoja de Excel según la clasificación del documento puede tener dos, tres o cuatro criterios de evaluación, donde se da la calificación por cada criterio y la hoja obtiene el promedio parcial, La calificación documental de la característica se obtiene*

promediando las calificaciones parciales de los documentos que pertenecen a dichas características.

- *En otra hoja de Excel en un cuadro llamado calificación final tomando la evaluación documental de cada característica se obtiene la calificación del factor.*
- *El seguimiento a las autoevaluaciones de los programas hasta ahora es que se está consolidando también en una hoja de Excel.*
- *Al no contar con un sistema de información toda la comunicación relacionada con el proceso se hace a través del correo electrónico.*
- *No es seguro manejar información importante en los drives de las cuentas instituciones, si algún funcionario sale de la institución se lleva la información vinculada a su cuenta.*

Rediseño del proceso de Autoevaluación Propuesta Estado Futuro (TO-BE)

Después de haber analizado el estado actual del proceso de autoevaluación, se observa que existe una necesidad de mejora para optimizar el proceso a través de su sistematización. Si bien el proceso está muy organizado, requiere algunas mejoras, las cuales se detallan a continuación:

- *Registrar los factores, características, indicadores y evidencias de cumplimiento del modelo de autoevaluación Unadista.*
- *Generar los instrumentos para realizar la apreciación de los programas y la institución.*
- *Acceso al sistema de RyC de la universidad específicamente para obtener los datos de los estamentos por programas, zonas y periodo académico.*
- *Calcular la población y muestra de programa en cada estamento por zonas y periodo académico.*
- *Enviar los instrumentos a cada grupo según la muestra obtenida.*
- *Los estamentos puedan dar respuesta al instrumento que le corresponda.*
- *Hacer seguimiento a los instrumentos en la medida que se vayan diligenciando (número de instrumentos diligenciados).*

- *Obtener promedio de las respuestas obtenidas en cada pregunta*
- *Calcular la calificación de cada característica por estamento y total de la característica y por cada calificación mostrar el juicio cualitativo de valor.*
- *Calcular la calificación del factor para la apreciación.*
- *Clasificar y calificar los documentos*
- *Calcular la calificación documental de la característica, aquí se presentan unos cálculos innecesarios dado que está condicionado a la formula en la hoja de cálculo.*
- *Presentar la evaluación documental de la característica y el factor.*
- *Calcular la calificación total de cada característica y el factor y mostrar el juicio cualitativo de valor.*
- *Calcular el resultado final de la autoevaluación del programa y mostrar el juicio cualitativo de valor.*
- *Presentar los resultados obtenidos*
- *Seguimiento a las autoevaluaciones*
- *El sistema debe contar con manuales de usuarios de acuerdo con su rol*

Definición de las Necesidades

De acuerdo con las necesidades encontradas a continuación son clasificadas en la tabla 9, Definición de necesidades del estado futuro donde a través de un sistema de información se espera obtener el estado deseado:

Tabla 9.*Definición de Necesidades del estado Futuro*

Necesidad	Actual	Deseado
Capacidad	La tabulación de los datos es lenta y tediosa y la información es bastante amplia. Desmesurado tiempo para realizar las tareas de elaboración, aplicación, tabulación y análisis del instrumento.	Recuperación rápida de la información Manejo de un volumen creciente de información Aumento de la velocidad de procesamiento de los datos
Control	Cálculos innecesarios La información de cada programa se almacena en distintas carpetas en la nube.	Exactitud y consistencia Información almacenada en un mismo lugar
Comunicación	La comunicación es vía correo electrónico, de igual manera la trazabilidad del proceso, la información está en diferentes lugares. mala comunicación entre los responsables del proceso y los interesados	Aumento en la comunicación Integración de las distintas unidades de la institución, mantener la información en un mismo lugar.

Seguridad	Posible pérdida de información La información es susceptible de que usuarios no autorizados puedan tener acceso	Acceso seguro a la información, solo personal autorizado según roles de usuario, información disponible.
------------------	--	--

Diagrama 6 Estado Futuro del Proceso de Autoevaluación TO-BE

En la propuesta del estado futuro TO-BE se propone un nuevo actor que sería el Sistema de Información de Autoevaluación Unadista, sobre el cual recaerían todas aquellas tareas que puedan ser automatizadas, es así como en la figura 9 se presenta el diagrama que consta de las siguientes actividades:

Inicio

- ✓ *VISAE previamente registrada en el sistema selecciona del menú de opciones busca el programa a autoevaluar, se confirma la ponderación de factores.*
- ✓ *El Sistema genera notificación tanto al líder nacional de programa y decano de escuela informando que iniciará la autoevaluación de un determinado programa.*
- ✓ *VISAE selecciona del menú elaborar instrumentos y crea los instrumentos a aplicar a los grupos de interés.*
- ✓ *El sistema guardar los instrumentos creados para el programa por grupo de interés.*
- ✓ *El líder nacional del programa recibe notificación este debe ingresar y autenticarse en el sistema, donde debe seleccionar del menú la opción ponderar las características para el programa.*
- ✓ *El sistema guarda la ponderación de las características para el programa*

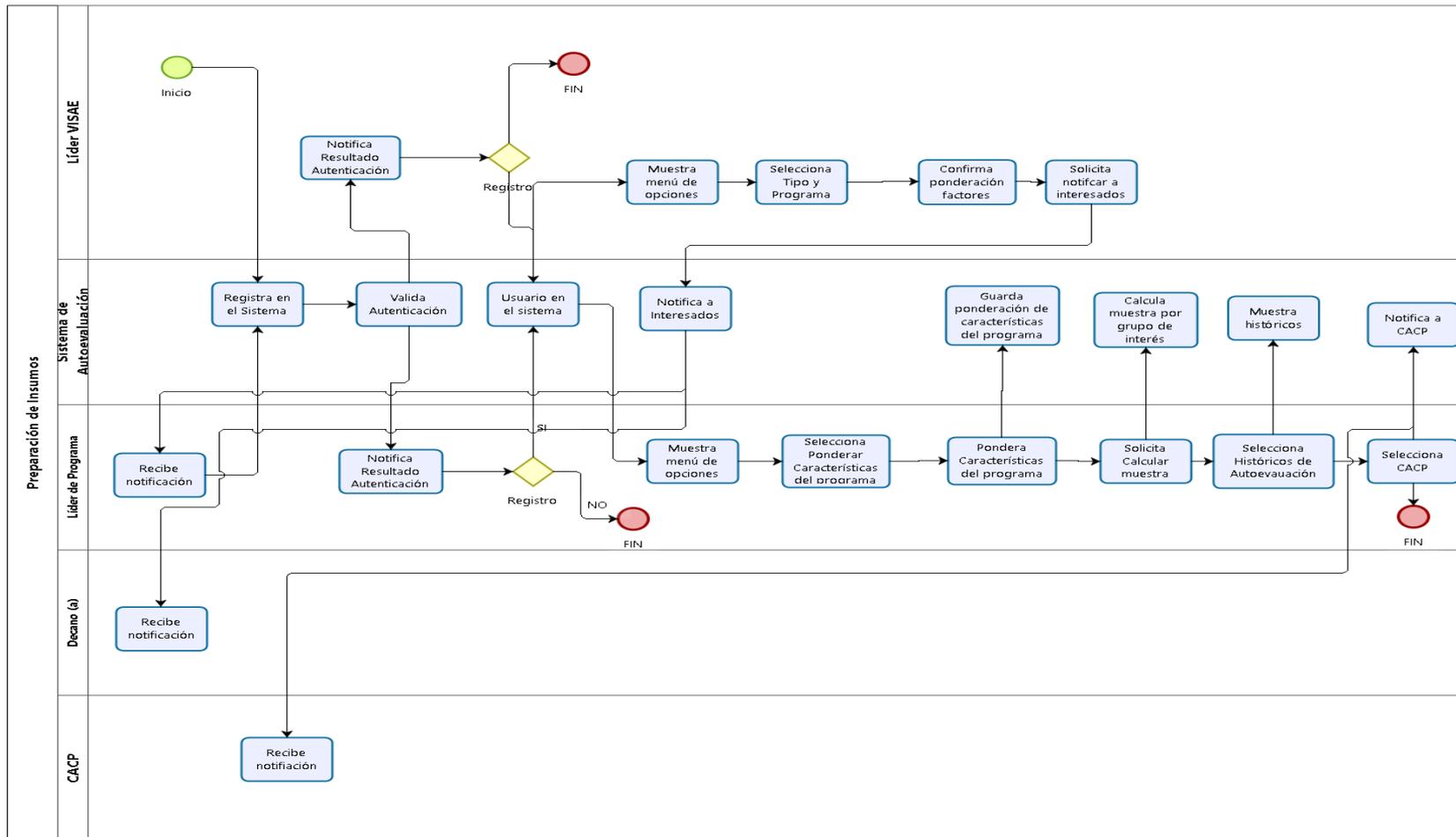
- ✓ *El líder nacional del programa selecciona del menú calcular población y muestra del programa*
- ✓ *El sistema calcula y visualiza la población y muestra del programa por zonas y grupos de interés.*
- ✓ *El líder nacional del programa selecciona del menú la opción obtener grupos de interés para aplicar instrumentos.*
- ✓ *El sistema devuelve listado con identificación los integrantes de los grupos de interés a responder los instrumentos.*
- ✓ *El líder nacional del programa selecciona del menú comité curricular de programa*
- ✓ *El sistema notifica a los miembros del comité curricular del programa*
- ✓ *Los miembros del comité curricular del programa reciben notificación del sistema*
- ✓ *El líder nacional del programa selecciona los instrumentos y los envía a los respectivos grupos de interés.*
- ✓ *Los miembros de los grupos de interés reciben en el correo institucional el enlace para acceder al instrumento a responder.*
- ✓ *Los grupos de interés responden los instrumentos*
- ✓ *El sistema hace seguimiento al número de respuestas obtenidas*
- ✓ *El sistema tabula las respuestas*
- ✓ *VISAE solicita al sistema calcular la apreciación del programa.*
- ✓ *Los miembros del comité curricular del programa autenticados en el sistema califican los documentos institucionales y del programa*
- ✓ *VISAE solicita al sistema calcular la calificación documental para el programa*
- ✓ *VISAE solicita al sistema obtener la calificación final del programa*

- ✓ *El líder del programa consulta los resultados de autoevaluación del programa*
- ✓ *El líder del programa realiza el informe final y plan de mejoramiento*
- ✓ *El líder nacional del programa formaliza en el sistema SSAM el plan de mejoramiento*
- ✓ *El decano (a) de escuela ejecuta el plan de mejoramiento*
- ✓ *El decano (a) de escuela hace seguimiento al plan de mejoramiento*

Fin

Figura 9.

Estado Futuro del Proceso de Autoevaluación TO-BE



Rediseño de los subprocesos de proceso de autoevaluación estado futuro (TO-BE)

Al igual que el proceso de autoevaluación, también fue necesario modelar los estados deseados de los subprocesos que lo componen.

Diagrama 7 Subproceso Preparación de Insumos TO-BE

La propuesta del estado futuro TO-BE del subproceso preparación de insumo se compone de las siguientes actividades:

Inicio

- ✓ *VISAE previamente registrada en el sistema selecciona del menú de opciones busca el programa a autoevaluar, se confirma la ponderación de factores.*
- ✓ *El Sistema genera notificación tanto al líder nacional de programa y decano de escuela informando que iniciará la autoevaluación de un determinado programa.*
- ✓ *El líder nacional del programa recibe notificación este debe ingresar y autenticarse en el sistema, donde debe seleccionar del menú la opción ponderar las características para el programa.*
- ✓ *El sistema guarda la ponderación de las características para el programa*
- ✓ *El líder nacional del programa selecciona del menú calcular población y muestra del programa*
- ✓ *El sistema calcula y visualiza la población y muestra del programa por zonas y grupos de interés.*
- ✓ *El líder nacional del programa selecciona del menú comité curricular de programa*
- ✓ *El sistema notifica a los miembros del comité curricular del programa*

Fin

Figura 10.

Subproceso Preparación de Insumos TO-BE

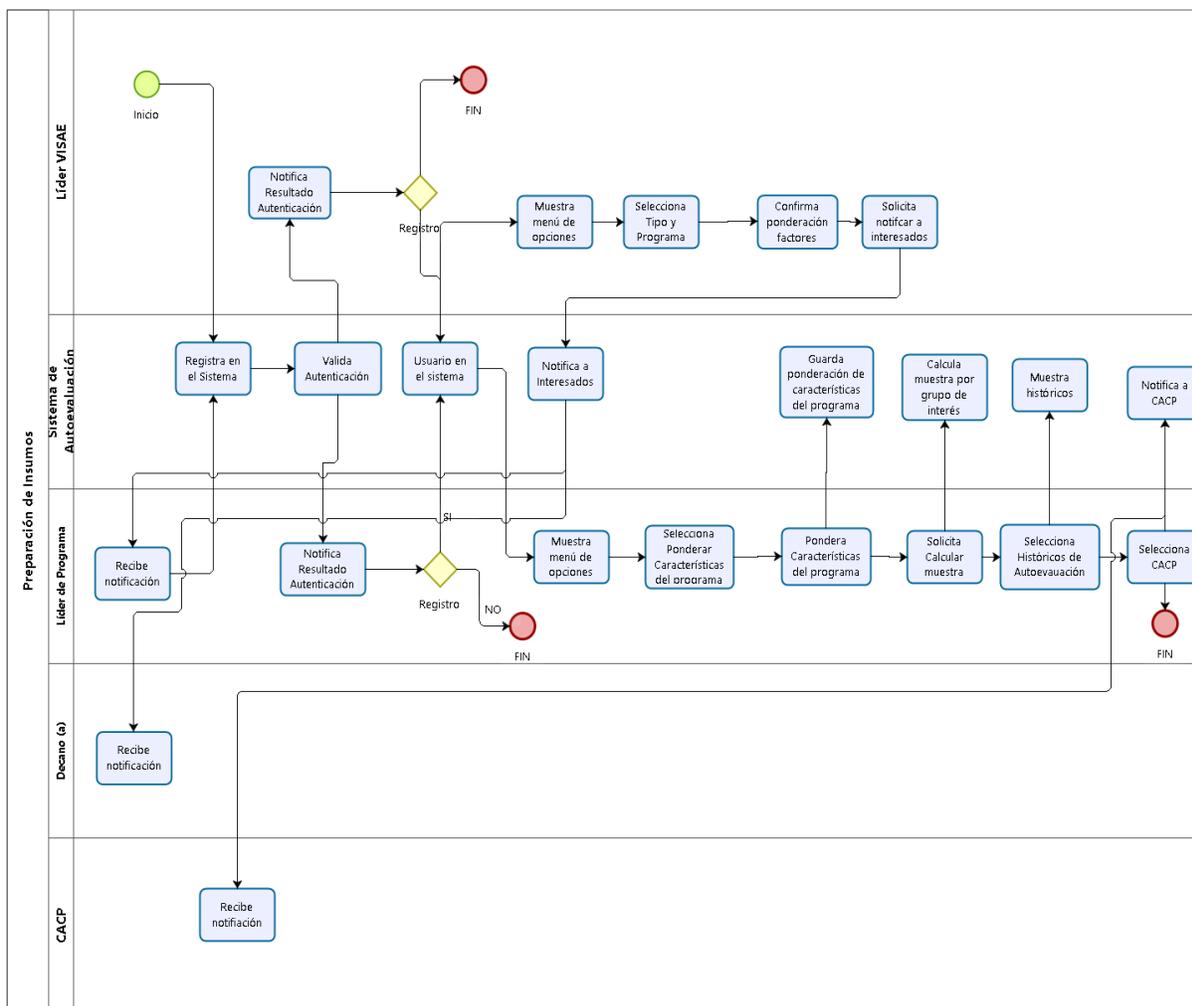


Diagrama 8 Subproceso Diseño, Aplicación y Tabulación de Instrumentos TO-BE

La propuesta del estado futuro TO-BE del subproceso diseño, aplicación y tabulación de instrumentos se compone de las siguientes actividades:

Inicio

- ✓ *VISAE selecciona del menú elaborar instrumentos y crea los instrumentos a aplicar a los grupos de interés.*
- ✓ *El sistema guardar los instrumentos creados para el programa por grupo de interés.*
- ✓ *El líder nacional del programa selecciona del menú la opción obtener grupos de interés para aplicar instrumentos.*
- ✓ *El sistema devuelve listado con identificación los integrantes de los grupos de interés a responder los instrumentos.*
- ✓ *El líder nacional del programa selecciona los instrumentos y los envía a los respectivos grupos de interés.*
- ✓ *Los miembros de los grupos de interés reciben en el correo institucional el enlace para acceder al instrumento a responder.*
- ✓ *Los grupos de interés responden los instrumentos*
- ✓ *El sistema hace seguimiento al número de respuestas obtenidas*
- ✓ *El sistema tabula las respuestas*
- ✓ *VISAE solicita al sistema calcular la apreciación del programa*

Fin

Figura 11.

Subproceso Diseño, Aplicación y Tabulación de Instrumentos TO-BE

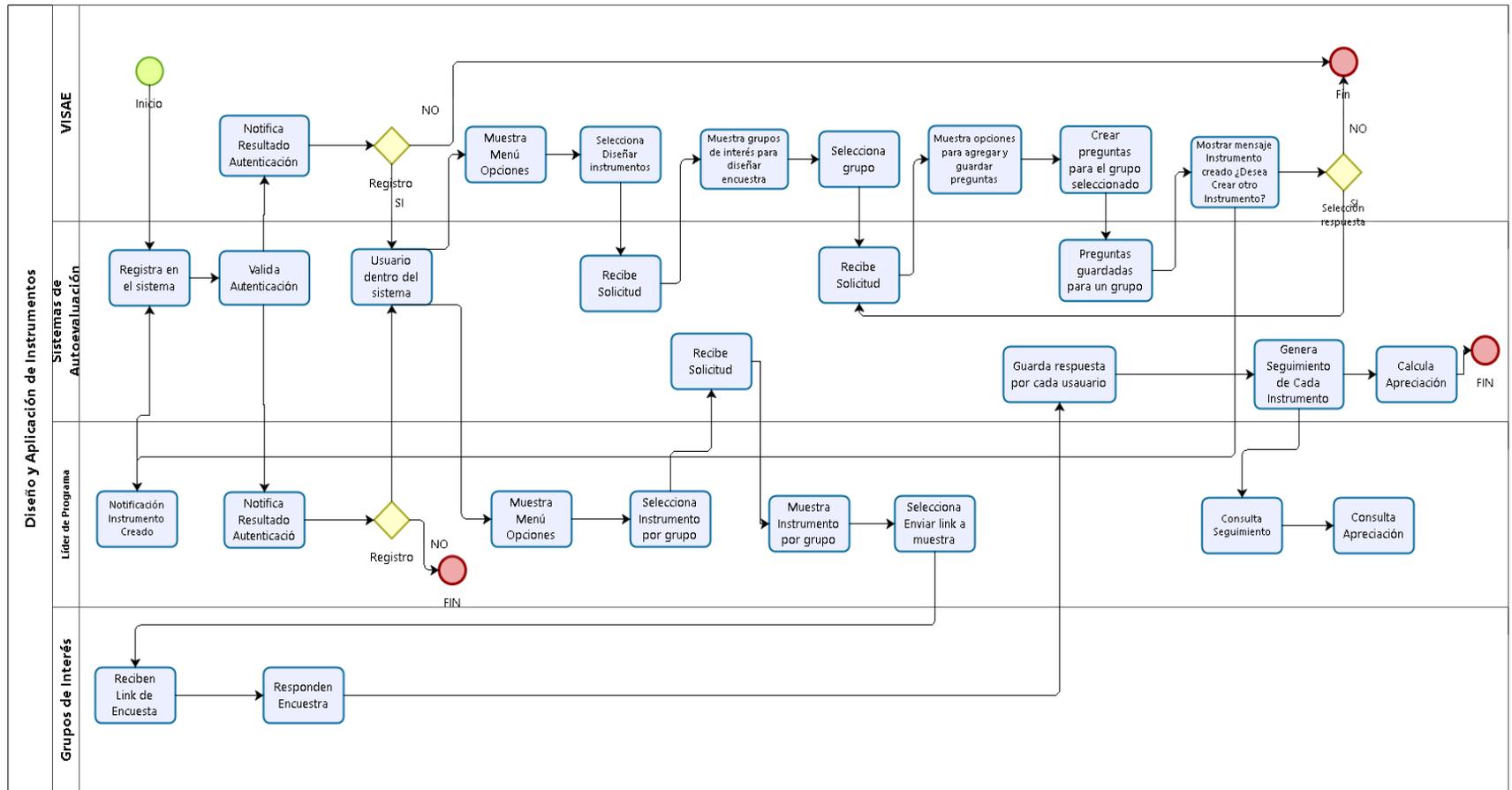


Diagrama 9 Recopilación Documental y Verificación de cumplimiento TO-BE

La propuesta del estado futuro TO-BE del subproceso recopilación documental y verificación de cumplimiento se compone de las siguientes actividades:

Inicio

- ✓ *Los miembros del comité curricular del programa autenticados en el sistema califican los documentos institucionales y del programa*
- ✓ *VISAE solicita al sistema calcular la calificación documental para el programa*
- ✓ *VISAE solicita al sistema obtener la calificación final del programa*

Fin

Figura 12.

Recopilación Documental y Verificación de cumplimiento TO-BE

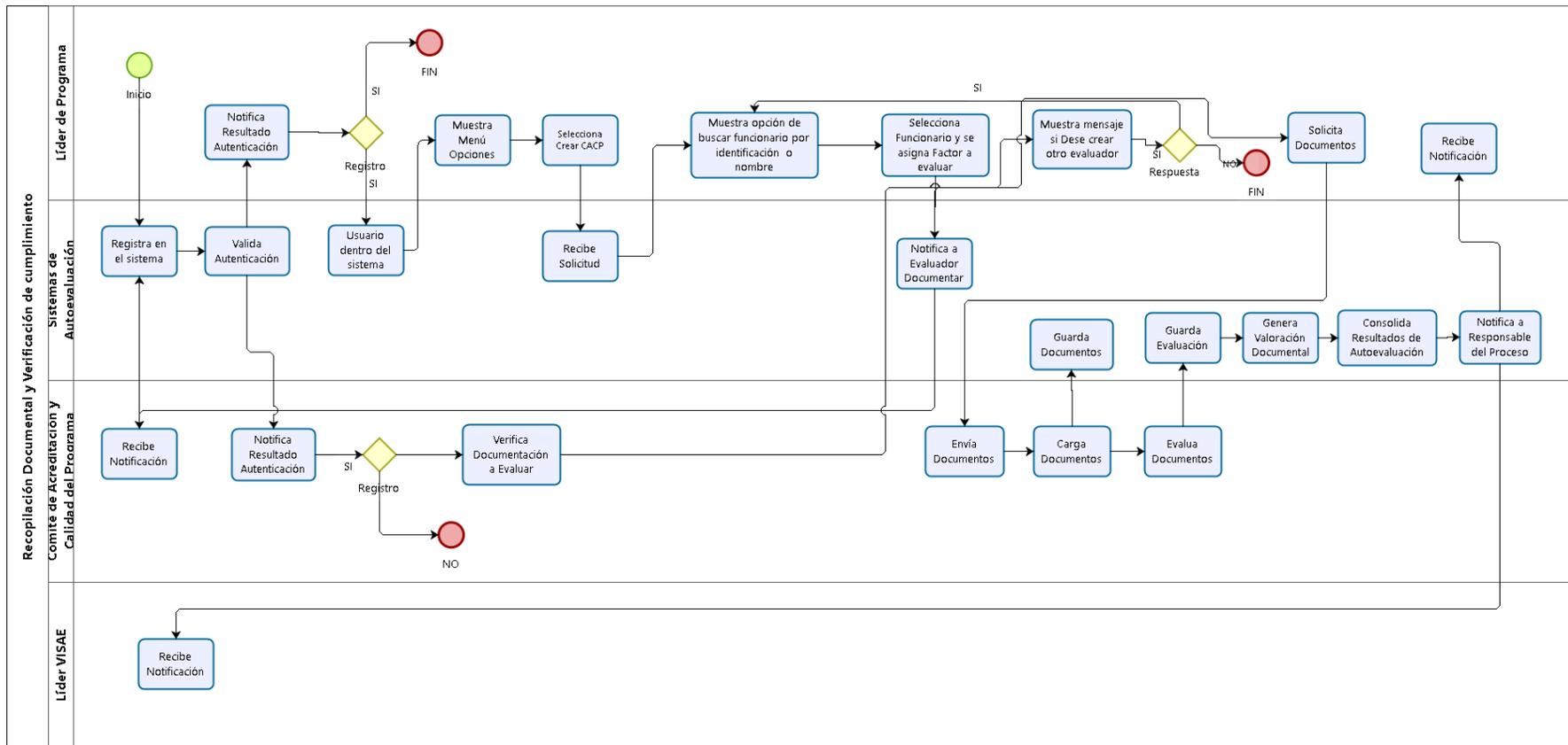


Diagrama 10 Subproceso Informe de autoevaluación y plan de mejoramiento TO-BE

La propuesta del estado futuro TO-BE del subproceso informe de autoevaluación y plan de mejoramiento se compone de las siguientes actividades:

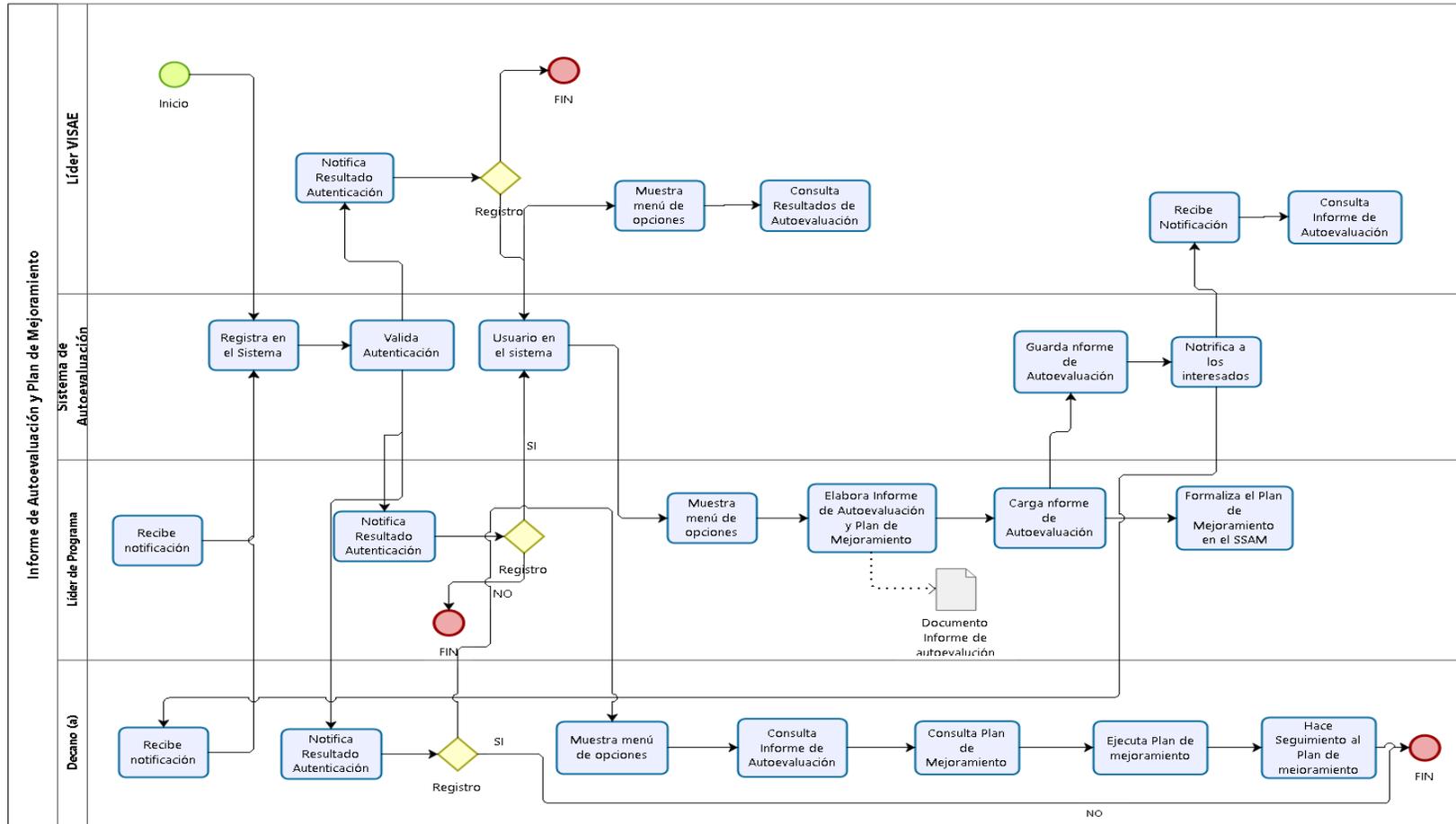
Inicio

- ✓ *VISAE solicita al sistema obtener la calificación final del programa*
- ✓ *El líder del programa consulta los resultados de autoevaluación del programa*
- ✓ *El líder del programa realiza el informe final y plan de mejoramiento*
- ✓ *El líder nacional del programa formaliza en el sistema SSAM el plan de mejoramiento*
- ✓ *El decano (a) de escuela ejecuta el plan de mejoramiento*
- ✓ *El decano (a) de escuela hace seguimiento al plan de mejoramiento*

Fin

Figura 13.

Subproceso Informe de autoevaluación y plan de mejoramiento TO-BE



Arquitectura de la solución para el proceso de Autoevaluación

La arquitectura de aplicaciones o arquitectura de software está orientada al diseño de la estructura del sistema, con el objetivo de dar solución a sus requisitos técnicos y del negocio, reduciendo los riesgos asociados a la construcción del software. Es el acoplamiento de diferentes componentes arquitecturales, de acuerdo con el tipo aplicación, que darán respuesta a las necesidades de los diferentes agentes del sistema.

En este capítulo se presenta el diseño de la arquitectura la cual incluye los diseños de los modelos para la construcción del prototipo del proceso de autoevaluación de la UNAD, basado en los puntos de vista propuestos por propuestos por (Rozanski et al., 2011), y así ofrecer una solución funcional que pueda ser mantenible y escalable con el tiempo.

Selección de la Arquitectura de Software

El diseño del prototipo de software que permitirá la automatización del proceso de autoevaluación de la UNAD se hace bajo la arquitectura de 3 capas, la cual se basa en una distribución jerárquica de roles y responsabilidades: el rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada (De La Torre Llorente, Zorrilla Castro, Ramos Barros, & Calvarro Nelson, 2010). La siguiente tabla describe las responsabilidades de cada capa:

Tabla 10.*Estructura de la arquitectura de tres (3) capas*

Capa de la Arquitectura	Responsabilidad
Capa de Presentación o Interfaz de Usuario	Está formada por los formularios y los controles que se encuentran en los formularios. Es la capa con la que interactúa el usuario.
Capa de Negocio	Formada por las entidades, que representan objetos que van a ser manejados o utilizados por toda la aplicación. En este caso, están representados por clases y “DataTables” que se crean.
Capa de Acceso a Datos	Contiene clases que interactúan con la base de datos; estas clases altamente especializadas se encuentran en la arquitectura del sistema y permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos, de forma transparente para la capa de negocio.

Se seleccionó esta arquitectura bajo el patrón Modelo Vista Controlador y de acuerdo con las características técnicas del lenguaje de programación y framework seleccionados para su desarrollo.

A continuación, se presenta la estructura del patrón MVC, el cual separa las responsabilidades a través de tres componentes, a saber:

Tabla 11.*Capas Patrón MVC*

Capa	Responsabilidad
Vista	Responsable de los elementos de la interfaz de usuario.
Controlador	Responsable de responder a las acciones de la interfaz de usuario.
Modelo	Responsable del comportamiento de la lógica del negocio.

Este patrón presenta las siguientes ventajas:

- ✓ Separación clara entre los componentes de un programa informático; lo cual permite su implementación por separado.
- ✓ Interfaz de Programación de Aplicaciones API (Application Programming Interface) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- ✓ Conexión entre el Modelo y sus Vistas dinámicas; se produce en tiempo de ejecución, no en tiempo de compilación.

Diseño de la arquitectura lógica de la aplicación

El diseño de la arquitectura de software es un proceso fundamental en la construcción de software; la diferencia entre un buen diseño o uno malo radica en el éxito o fracaso del proyecto, debido a que, además de tener en cuenta los requisitos funcionales, también es necesario que contemple los atributos de calidad, los cuales están orientados a los requisitos no funcionales del proyecto. Teniendo en cuenta esta significación, a continuación, se procede a diseñar la arquitectura de solución para la gestión del proceso de autoevaluación:

Selección de aplicación a desarrollar

En el capítulo anterior se pudo analizar el proceso de autoevaluación a través de su modelamiento del estado actual AS-IS y así proponer su estado óptimo o TO-BE, de acuerdo con análisis, y basado en que la institución es una Mega universidad, que opera en todo el territorio nacional, el tipo de aplicación más apropiado es una aplicación web, la cual “se consume mediante un navegador web ofreciendo una interfaz de usuario interoperable” (Dissanayake & Dias, 2017).

Selección de Puntos de Vista para la Arquitectura y Diseño de Artefactos

Para describir los diferentes aspectos del prototipo de software del proceso de autoevaluación, fue necesario seleccionar los puntos de vista propuestos por (Rozanski et al., 2011), para mostrar propiedades específicas, ya que cada uno representa un comportamiento en particular. En la siguiente tabla se presentan las vistas seleccionadas:

El catálogo contiene seis puntos de vista centrales: los puntos de vista funcional, de información, de concurrencia, de desarrollo, de implementación y operacional.

Tabla 12.*Puntos de Vistas Arquitecturales. Propuestos por Rozanski & Woods, 2012*

Punto de Vista	Descripción	Interesados	Diagrama
Funcional	Describe los elementos funcionales del sistema, así como las responsabilidades, interfaces e interacciones. Es el más utilizado por los interesados del proyecto de software y tiene mucha influencia en los atributos de calidad del sistema.	Todos los interesados	Modelo de componentes
Información	Describe como la arquitectura guarda, manipula, administra y distribuye la información, está dirigido a los usuarios finales, desarrolladores y administradores.	usuarios principales desarrolladores probadores o testers	Modelo de estructura estática de datos Modelos de flujo de información Modelo de ciclo de vida de información
Concurrencia	Describe la estructura de concurrencia del sistema y asigna elementos funcionales a	Comunicadores desarrolladores	Modelo de concurrencia

	<p>unidades de concurrencia para identificar claramente las partes del sistema que pueden ejecutarse simultáneamente y cómo se coordina y controla.</p>	<p>probadores o testers algunos administradores</p>	<p>Modelo de simultaneidad a nivel de sistema Modelo de estado</p>
Desarrollo	<p>Se centra en la organización real de los módulos de software en el ambiente de desarrollo del software.</p>	<p>Desarrolladores Ingenieros de producción probadores o testers</p>	<p>Modelo de estructura de módulo Diagrama de componentes</p>
Implementación	<p>Muestra cómo se distribuyen los distintos componentes software del sistema en los nodos físicos. Ilustra cómo es el ambiente en el que el sistema será instalado y ejecutado, incluyendo las dependencias que el sistema tiene en el entorno de ejecución.</p>	<p>Administradores del sistema Comunicadores desarrolladores probadores o testers asesores</p>	<p>Modelos de plataforma en tiempo de ejecución Diagrama de implementación Diagrama de cajas y líneas Diagrama de recuadros y líneas</p>

		Modelo de Despliegue
Operacional	Describe cómo se operará, administrará y apoyará el sistema cuando se esté ejecutando en su entorno de producción.	Modelos de instalación Modelos de migración Modelos de gestión de la configuración Modelos de administración Modelos de soporte
	Administradores del sistema Ingenieros de producción Desarrolladores probadores o testers asesores	

Teniendo en cuenta que del catálogo propuesto es posible seleccionar alguno de los seis (6) puntos de vista propuestos, se tomaron cuatro (4) considerando que estos son los más representativos los cuales son funcional, de información, de desarrollo e implementación los cuales se presentan a continuación:

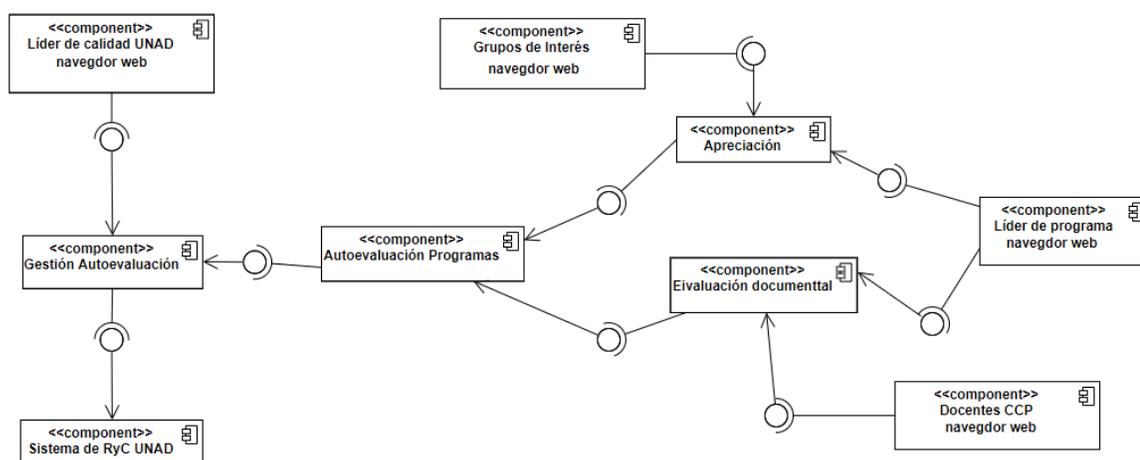
Vista Funcional del Sistema

Con base en la arquitectura seleccionada, se presenta el diagrama de componentes elaborado con la herramienta StarUML 3.1.0 donde la vista consta de nueve (9) componentes los cuales tienen una función específica dentro del sistema.

Los actores acceden a través de un navegador web, el rol de líder de calidad de la institución UNAD, a través de una interfaz tiene acceso al módulo de gestión de autoevaluación y éste a un servicio de registro y Control de la UNAD que provee la interfaz de autoevaluación de programas, interfaces de acceso a los módulos de apreciación y evaluación documental. Estos módulos también pueden ser accedidos por los líderes de los programas académicos, donde los distintos grupos de interés dejan evidencia de su apreciación y posteriormente, el CCP procede a realizar la evaluación documental.

Figura 14.

Modelo de Componentes Sistema de Autoevaluación



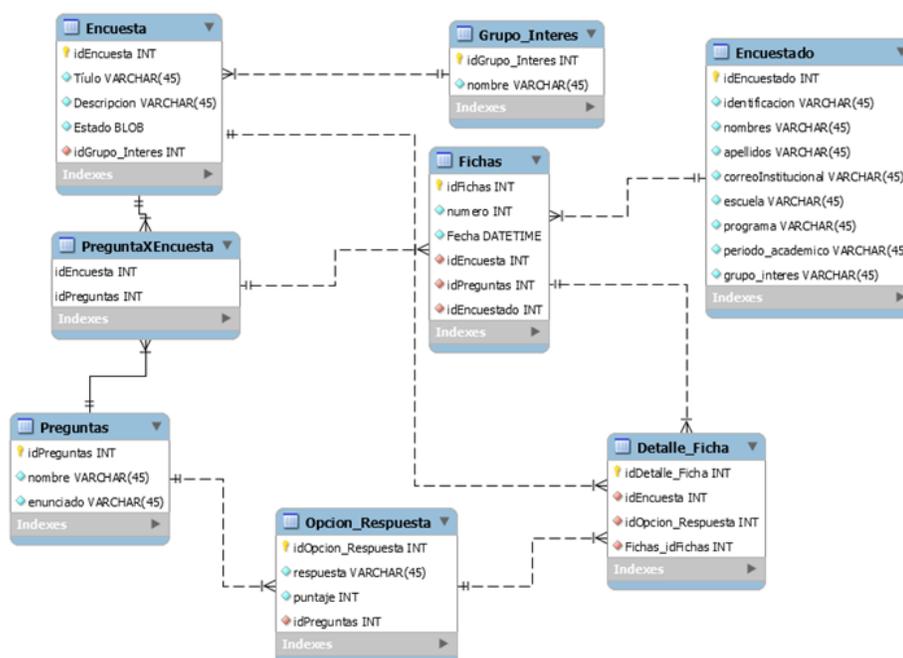
Vista de Información Sistema de Autoevaluación

Esta vista se ve representada por el modelo de datos del sistema de autoevaluación Unadista elaborado con la herramienta MySQL Workbench 8.0 CE, el cual contiene las tablas maestras y del negocio que van a garantizar la correcta gestión de la información. Parte de la solución propuesta se puede apreciar en la figura 15, donde se muestran las tablas relacionadas al proceso de la autoevaluación de los programas a través de encuestas para los diferentes grupos de

interés; entre éstas se modelan las tablas: Encuesta, Grupo_Interes, preguntas, opción_respuesta, preguntaXEncuesta, Fichas, Encuestado y Detalle_Fichas.

Figura 15.

Modelo de Datos Encuesta del proceso de Autoevaluación



Vista de Desarrollo Sistema de Autoevaluación

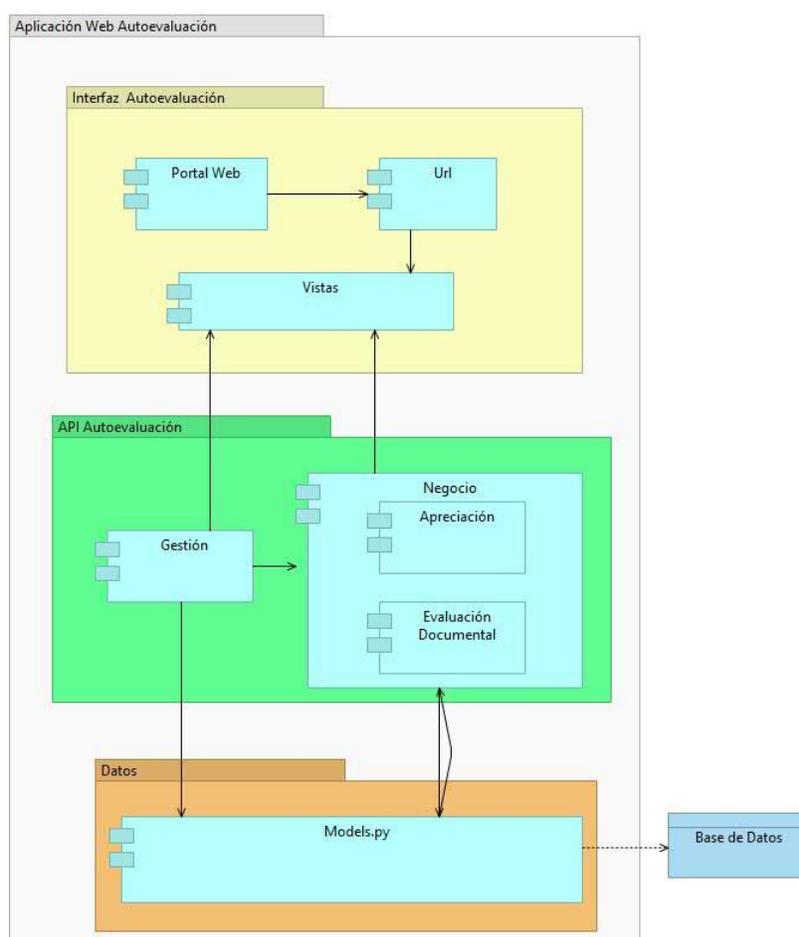
Para la diagramación de la vista de desarrollo se utilizó la herramienta Archi 4.4.0. En esta vista se presenta una separación de responsabilidades entre los componentes del sistema categorizando en tres componentes dichas responsabilidades según sus funciones, y obteniendo así la interfaz de usuario, la lógica del negocio y el acceso a los datos. De esta manera es posible aumentar la flexibilidad, mantenibilidad, y escalabilidad en el software de autoevaluación.

Como puede apreciarse en la Figura 16, la solución propuesta plantea tres componentes así: la interfaz de usuario, a la cual se accede mediante un navegador web, aquí se procesan las peticiones del usuario; API Autoevaluación, es el componente encargado de procesar las

peticiones solicitadas en el cliente, en este componente se incluyen a su vez dos componentes; de Gestión, el cual administra tablas maestras y del Negocio, que se encarga de la apreciación y la evaluación documental; por último tenemos la capa de datos donde están los modelos y accede a la base de datos para proveer respuesta a la API. Aplicación web Autoevaluación: corresponde a la capa de presentación, aquí se encuentran las interfaces con que los usuarios interactúan al hacer una petición mediante las URL y las vistas.

Figura 16.

Modelo de Estructura de módulos Sistema de Autoevaluación UNAD

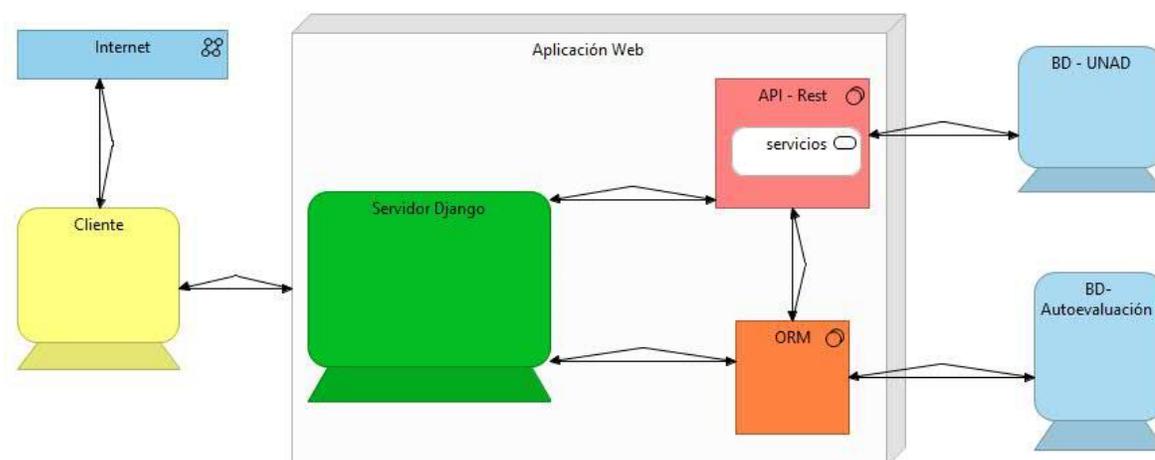


Vista de Implementación Física Sistema de Autoevaluación

Para la vista de implementación también se utilizó la herramienta Archi 4.4.0. En esta vista se presentan los componentes físicos necesarios para la ejecución del sistema como son: el Cliente: Por medio de un navegador web, hará las peticiones al servidor Django, La aplicación web a su vez se compone del Servidor Django que administra las vistas, URL, modelos de la aplicación web Autoevaluación, el ORM que gestiona la base de datos y la API – Rest que interactúa con el ORM, y a través de servicios accede a aplicaciones externas.

Figura 17.

Modelo de Despliegue Sistema de Autoevaluación UNAD



Desarrollo e Implementación del Prototipo de Software Para el Proceso de Autoevaluación

En este último capítulo, se describe en detalle cómo se realizó el proceso de desarrollo de software teniendo en cuenta la metodología de desarrollo ágil SCRUM, por lo cual para la obtención de los requerimientos se implementaron las historias de usuario, seguidamente para el desarrollo del prototipo se dividió en dos partes backen y el frontend y finalmente elaboración de pruebas funcionales.

Identificación de requerimientos por historias de Usuario

Para identificar las necesidades de los interesados, que deben cumplir con una determinada funcionalidad, que será ejecutada a través del sistema de información para procesos de autoevaluación, y en coherencia con la metodología de desarrollo ágil utilizada se hará uso de historias de usuario.

Una historia de usuario se usa, dentro de la ingeniería de requisitos ágil, para describir una funcionalidad que será útil para el usuario.(Cohn Mike, 2004), por tanto es una herramienta que combina la comunicación verbal y escrita. Entre sus ventajas se encuentran:

- *pueden implementarse rápidamente (días o semanas)*
- *Necesitan poco mantenimiento*
- *Mantienen una relación cercana con el cliente.*
- *Permiten dividir los proyectos en pequeñas entregas.*
- *Permiten estimar fácilmente el esfuerzo de desarrollo.*

Tabla 13.*Historia de Usuario HU001*

Historia de Usuario	
ID	HU001
Actividad	Gestionar Dimensión
Actor	Líder VISAE
Prioridad	Alta
Riesgo	Alto
Descripción	El líder VISAE, es el responsable de las dimensiones del modelo de autoevaluación de la UNAD, debe registrar, actualizar o eliminar alguna en caso de que hay un cambio en el modelo.
Validación	<ul style="list-style-type: none"> ✓ Se debe validar el número máximo de dimensiones (6) es decir que no se pase de ese valor. ✓ Si las dimensiones cambian estas pueden ser modificadas ✓ La dimensión tiene un código, nombre y descripción ✓ El líder VISAE puede consultar las dimensiones

Tabla 14.*Historia de Usuario HU002*

Historia de Usuario	
ID	HU002
Actividad	Gestionar Factor
Actor	Líder VISAE
Prioridad	Alta
Riesgo	Alto
Descripción	El líder VISAE, es el responsable de actualizar los factores o registrar alguno en caso de que hay un cambio en el modelo.
Validación	<ul style="list-style-type: none"> ✓ Se debe validar el número máximo de factores (15) es decir que no se pase de ese valor. ✓ Si los factores cambian estos pueden ser modificados ✓ El factor tiene un código, nombre, descripción y peso. ✓ El líder VISAE puede consultar los factores

Tabla 15.*Historia de Usuario HU004*

Historia de Usuario	
ID	HU004
Actividad	Gestionar Indicador
Actor	Líder VISAE
Prioridad	Alta
Riesgo	Alto
Descripción	El líder VISAE, es el responsable de actualizar los indicadores o en caso de que hay un cambio en el modelo.
Validación	<ul style="list-style-type: none"> ✓ Se debe validar el número máximo de indicadores de acuerdo con el nivel de formación del programa son: sesenta y seis (66) para pregrado y especialización y sesenta y dos (62) para maestría y doctorado. ✓ El indicador tiene un código, nombre, descripción y evidencias de cumplimiento. ✓ El líder VISAE puede consultar y editar los indicadores en caso de que se requiera.

Tabla 16.*Historia de Usuario HU005*

Historia de Usuario	
ID	HU005
Actividad	Gestionar Evidencia de cumplimiento
Actor	Líder VISAE
Prioridad	Alta
Riesgo	Alto
Descripción	El líder VISAE, es el responsable de actualizar las evidencias de cumplimiento o en caso de que hay un cambio en el modelo.
Validación	<ul style="list-style-type: none"> ✓ Se debe validar el número máximo de evidencias de cumplimiento de acuerdo con el nivel de formación del programa son: trescientos treinta y un (331) para programas de pregrado y especialización y doscientos sesenta (260) para programas de maestría y doctorado. ✓ La evidencia de cumplimiento tiene un código, nombre, descripción y evidencias de cumplimiento. ✓ El líder VISAE puede consultar y editar los indicadores en caso de que se requiera. ✓ Hay que tener en cuenta que cada evidencia apunta una característica.

Tabla 17.*Historia de Usuario HU006*

Historia de Usuario	
ID	HU006
Actividad	Ponderar Característica del programa
Actor	Líder de programa
Prioridad	Alta
Riesgo	Alto
Descripción	El líder del programa en compañía del comité curricular debe establecer el peso de cada características, la característica pertenece a un factor determinado.
Validación	<ul style="list-style-type: none"> ✓ Quiero que la ponderación quede guardada en el sistema y se pueda ver en vista de resultados por el líder de programa, VISAE y decano de escuela ✓ Cada ponderación es por programa ✓ Quiero que se vea como en el anexo 1 ✓ Tener en cuenta que las características que pertenecen a un mismo factor. en total deben sumar 1

Tabla 18.*Historia de Usuario HU007*

Historia de Usuario	
ID	HU007
Actividad	Generar históricos de autoevaluación
Actor	Unidades de apoyo y líder de programa
Prioridad	Alta
Riesgo	Medio
Descripción	<p>Los históricos de autoevaluación es la información de los anteriores ejercicios de autoevaluación del programa y es proporcionada por las unidades de apoyo para el líder del programa, el objetivo es poder comparar como ha sido el comportamiento de autoevaluación, es decir si se mantiene igual, si ha mejorado o si ha disminuido.</p> <p>El nuevo sistema debe permitir que esos históricos se guarden en un repositorio y se puedan consultar.</p>
Validación	<ul style="list-style-type: none"> ✓ Los históricos pueden ser consultados por año y programa, por líder VISAE, decanos y líder de programa.

Tabla 19.*Historia de Usuario HU008*

Historia de Usuario	
ID	HU008
Actividad	Obtener población y muestra por grupo de interés
Actor	Líder del programa
Prioridad	Alta
Riesgo	Alto
Descripción	Actualmente la muestra de cada grupo de interés es calculada por el líder del programa, el nuevo sistema debe calcular la muestra por zona y grupo de interés teniendo en cuenta la población de cada grupo en la zona
Validación	<ul style="list-style-type: none"> ✓ La muestra debe guardarse y tener en cuenta el año y periodo académico en que se calculó. ✓ Debe visualizarse la población por zona y grupo de interés, con su respectiva muestra como se muestra en el anexo 2.

Tabla 20.*Historia de Usuario HU009*

Historia de Usuario	
ID	HU009
Nombre	Diseñar instrumentos
Actor	Líder VISAE
Prioridad	Alta
Riesgo	Alto
Descripción	<p>El líder VISAE es la persona encargada de diseñar los instrumentos por cada grupo y compartir el enlace con los líderes zonales para enviar a la muestra obtenida por cada grupo de interés.</p> <p>El nuevo sistema debe permitir que el líder cree las encuestas con sus respectivas preguntas para cada grupo y estas están asociadas a unas características, estas preguntas deberían ser las mismas para futuros ejercicios de autoevaluación pero siempre permitir agregar nueva pregunta o modificar alguna existente.</p>
Validación	<ul style="list-style-type: none"> ✓ Pregunta guardada asociada a una característica ✓ Visualizar los instrumentos para cada grupo. ✓ Botón de confirmación para agregar pregunta.

Tabla 21.

Historia de Usuario HU010

Historia de Usuario	
ID	HU010
Nombre	Aplicar instrumentos
Actor	Líder de programa y líderes zonales
Prioridad	Alta
Riesgo	Alto
Descripción	<p>El líder del programa envía a los líder zonales los enlaces de cada encuesta de los grupos de interés, y estos a su vez los reenvían a los integrantes de la muestra seleccionada.</p> <p>El nuevo sistema debe permitir al líder de programa generar un enlaces que le llegue a cada miembro de la muestra del grupo de interés y autenticarse con su usuario y contraseña institucional para poder responder la encuesta, con respecto al sector productivo el sistema debe permitir que cree un usuario y contraseña para que pueda responder las preguntas</p>
Validación	<ul style="list-style-type: none"> ✓ Líder genera los enlaces de las encuestas y da la orden para que se envíen a la muestra seleccionada. ✓ Visualizar estadísticas en tiempo real de las respuestas ✓ Respuestas guardadas por cada participante ✓ Visualizar tabulación de resultados por grupos de interés ✓ Visualizar seguimiento de aplicación de instrumentos con base en el anexo 3 pero utilizando gráficas. ✓ Visualizar la apreciación del programa una vez cerrados los instrumentos como se muestra en el anexo 4, tener en cuenta que aquí se califica cada característica.

Tabla 22.

Historia de Usuario HU011

Historia de Usuario1	
ID	HU011
Nombre	Recopilar evidencia documental
Actor	Líder del programa y Comité de Acreditación y Calidad del Programa CACP
Prioridad	Alta
Riesgo	Alto
Descripción	<p>El líder del programa recopila la evidencia documental y la almacena en un drive, para que el Comité de Acreditación y Calidad del Programa, clasifique cada documento según el tipo,</p> <p>El nuevo sistema ya debe venir con la clasificación de la evidencia documental según la característica y proveer el espacio para subir la documentación de los documentos y relacionarlos de acuerdo con cada característica y evidencia de cumplimiento</p>
Validación	<ul style="list-style-type: none"> ✓ Documentación guardada ✓ Visualización de documentos categorizados y guardados ✓ Los documentos se categorizan de acuerdo con anexo 5

Tabla 23.*Historia de Usuario HU012*

Historia de Usuario	
ID	HU012
Nombre	Seleccionar evaluadores para el Comité de Acreditación y Calidad del Programa
Actor	Líder de programa
Prioridad	Alta
Riesgo	Alto
Descripción	El líder del programa debe seleccionar a los evaluadores que harán parte del comité de Acreditación y Calidad del Programa quienes van a evaluar los documentos correspondientes a los factores que les sean asignados. El nuevo sistema debe permitir seleccionar estos evaluadores y asignarles los factores a evaluar, una vez seleccionados les debe llegar notificación con toda esta información.
Validación	<ul style="list-style-type: none"> ✓ El líder puede buscar al evaluador por cedula o nombre. ✓ Una vez encontrado se le asignan los factores a evaluar. ✓ El sistema notifica al evaluador la asignación de los factores y el plazo para su revisión. ✓ El sistema solo debe permitir asignar evaluadores de acuerdo con número máximo que debe estar conformado el CACP

Tabla 24.*Historia de Usuario HU013*

Historia de Usuario	
ID	HU013
Nombre	Calificar evidencia documental
Actor	Comité de Acreditación y Calidad del Programa
Prioridad	Alta
Riesgo	Alto
Descripción	El comité de Acreditación y Calidad del Programa de acuerdo con el tipo de documento le da una calificación a cada uno, conforme a los atributos para calificar. El nuevo sistema debe permitir a los evaluadores del comité registrarse y ver los documentos asociados al factor y características a evaluar.
Validación	<ul style="list-style-type: none"> ✓ La interfaz de calificación de documentos debe ser amigable para el usuario, que solo vea los documentos a evaluar. ✓ Visualización de documentos calificados de acuerdo con el anexo 6. ✓ El sistema debe calcular la calificación documental de las características asociadas a los ✓ El informe final de la calificación de las evidencias documentales debe visualizarse De acuerdo con el anexo 7 ✓ El líder del programa, decanos y líder VISAE pueden visualizar los resultados

Tabla 25.*Historia de Usuario HU014*

Historia de Usuario	
ID	HU014
Nombre	Consolidar resultados
Actor	Líder VISAE
Prioridad	Alta
Riesgo	Alto
Descripción	<p>El líder VISAE toma los resultados de la apreciación y valoración documental para consolidarlos en la matriz resultado final de autoevaluación del programa en Excel.</p> <p>El nuevo sistema debe traer la calificación de cada características por apreciación de grupos de interés y valoración documental, al igual que la ponderación para que calcule la calificación definitiva de la característica y seguidamente emitir la calificación del factor y la calificación final del programa.</p>
Validación	<ul style="list-style-type: none"> ✓ Calificación total de características guardada ✓ Calificación del factor guardada ✓ Visualización del resultado final de autoevaluación del programa como se observa en el anexo 8 ✓ Emitir juicio de valor tanto para la característica como para la evaluación final. ✓ Hay que tener en cuenta que la tabla del juicio de valor cambió.

Tabla 26.*Historia de Usuario HU015*

Historia de Usuario	
ID	HU015
Nombre	Elaborar el informe final
Actor	Líder del programa
Prioridad	Alta
Riesgo	Alto
Descripción	El líder del programa elabora el informe final de autoevaluación y lo carga al sistema
Validación	<ul style="list-style-type: none"> ✓ Informe final de autoevaluación cargado en PDF ✓ Visualización del informe guardado por año

Tabla 27.*Historia de Usuario HU016*

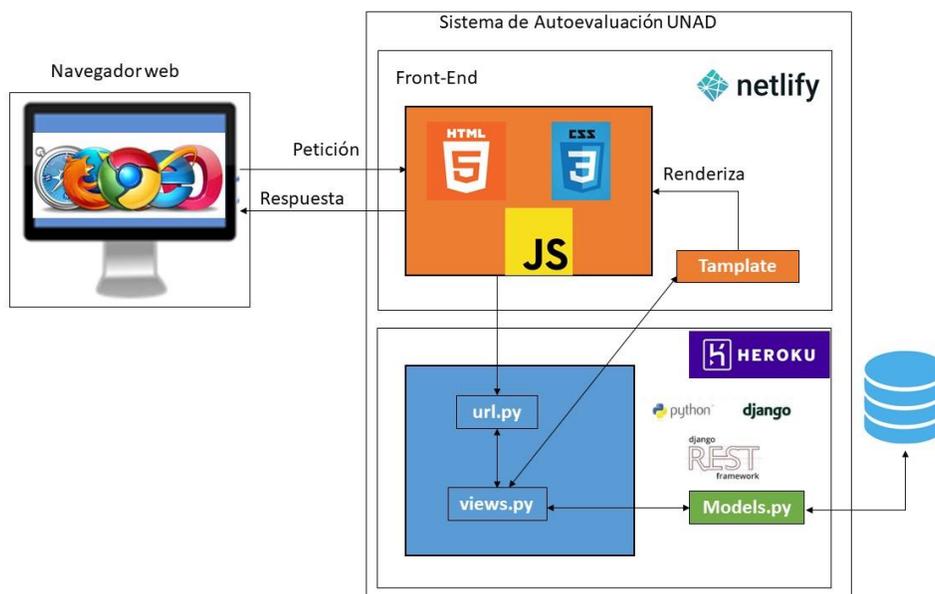
Historia de Usuario	
ID	HU016
Nombre	Elaborar plan de mejoramiento
Actor	Líder del programa
Prioridad	Alta
Riesgo	Alto
Descripción	El líder del programa elabora el plan de mejoramiento de autoevaluación y lo carga al sistema.
Validación	<ul style="list-style-type: none"> ✓ Informe final de autoevaluación cargado en PDF ✓ Visualización del informe guardado por año

Con base en el paradigma de diseño algorítmico Divide y Vencerás, el prototipo de software para el proceso de autoevaluación se divide en dos partes: la primera corresponde al back-end, que incluye la persistencia de los datos y la lógica del negocio y, la segunda el front-end, que tiene la responsabilidad de la interfaz gráfica de usuario (GUI) y con bajo acoplamiento frente al back-end al momento de la integración.

Teniendo en cuenta el patrón de arquitectura seleccionado (MVC) y el lenguaje seleccionado, se puede observar en la figura 18 la arquitectura de la aplicación.

Figura 18.

Arquitectura de la aplicación

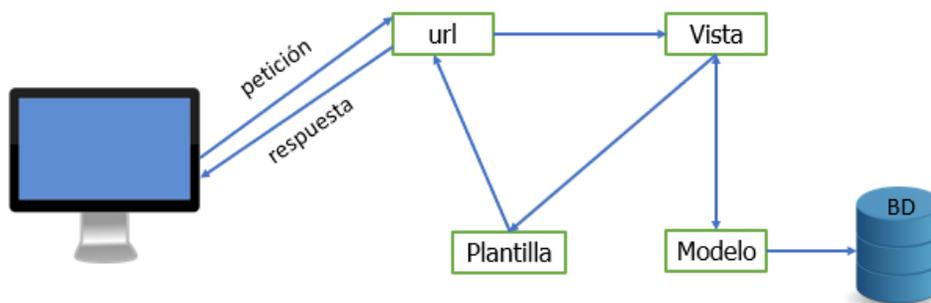


Desarrollo del Back-End.

De acuerdo con (Valdivia Caballero, 2016), en el contexto del desarrollo de aplicaciones están implicadas las actividades realizadas del lado del servidor; es decir, la programación de algoritmos orientados a la captura, actualización, eliminación y listar información para su

almacenamiento en la base de datos, así como la automatización de tareas relacionadas con el negocio.

La solución propuesta para gestionar ejercicios de autoevaluación a programas académicos de educación es un software orientado a la web, en este sentido, para el desarrollo del back-end se seleccionó la herramienta de desarrollo Django, que de acuerdo con (Molina Ríos et al., 2016) es el mejor framework para aplicaciones web escrito en Python, respeta el patrón de diseño modelo vista controlador, permitiendo la separación de responsabilidades, pero adaptándose a este enfoque bajo el patrón de arquitectura Modelo -Vista - Plantilla (MVT), porque hay tres capas independientes en Django, responsables de administrar diferentes partes de una aplicación. La figura 19 describe cómo funciona el Modelo MVT, donde el cliente por medio del navegador se comunica con una URL y hace una petición, por su parte la URL lee una vista que hace una funcionalidad, seguidamente, la vista llama a los modelos para obtener los datos de la BD y ejecuta alguna funcionalidad y obtiene los datos, finalmente ya con los datos la vista llama a la plantilla y le envía los datos; la plantilla es una página HTML y decide como se va a ver la información. Cabe aclarar que la comunicación entre las capas solo es posible mediante la programación de una Interfaz de aplicaciones API.

Figura 19.*Modelo Vista Plantilla -MVT-*

Continuando con el Patrón MVT en la tabla 28 se presenta la descripción de los elementos mostrados en la imagen anterior:

Tabla 28.*Estructura del Modelo MVT*

Capa	Responsabilidad
Plantilla	Es el elemento HTML que decide cómo se van a mostrar los datos devueltos por la vista. Cada plantilla obtiene una respuesta proveniente de la vista correspondiente y contiene la estructura necesaria para presentar los datos en la forma necesaria.
Vista	Aquí se procesan las peticiones o solicitudes (request) que accederán al Modelo (Model) para poder obtener o entregar datos. Es decir, en la vista se establece la lógica necesaria para devolver una respuesta (y que datos conforma dicha respuesta) hacia el cliente que la solicita.

Modelo Es la fuente única y definitiva de los datos de una aplicación que permite realizar operaciones con ellos. Dicho de otra forma, un modelo se entiende como una tabla de la base de datos que soporta la aplicación.

Fuente: (Condori Ayala, 2012)

Tecnologías Empleadas para el back-end

Dado lo anterior, el back-end corresponde al servidor y su función es procesar la información recibida por los usuarios del sistema de información, por tanto, con base en la arquitectura y el lenguaje de programación seleccionados, MVT y Python respectivamente, y como se mencionó anteriormente, el framework de desarrollo para este lenguaje que se escogió fue framework Django (Django Software Foundation, n.d.) en su versión 2.2.6, SQLite para la base de datos y Django Rest Framework (Encode OSS Ltd., 2016) para generar la API y dfr-yasg 1.17.0. (Python Software Foundation, 2020), que permite documentar la API en Swagger(SmartBear Software., 2021). Los entornos de trabajo para el desarrollo del servidor son: Pycharm (JetBrains s.r.o., 2020) como editor de código multiplataforma desarrollado por JetBrains y para los servicios de un sistema de control de versiones VCS mediante un repositorio remoto que hace uso de git y GitLab (Software Freedom Conservancy, n.d.).

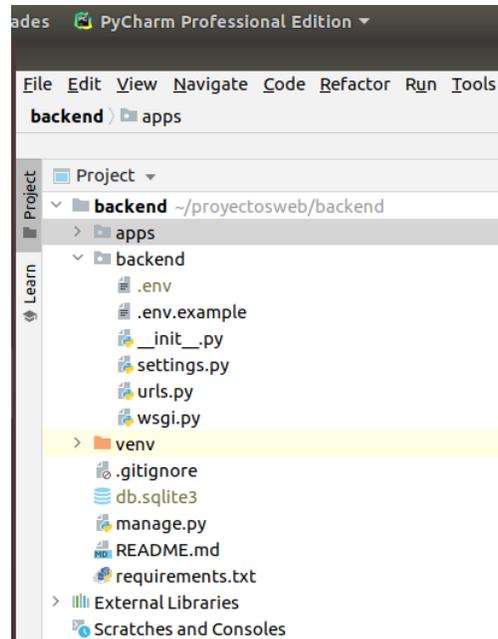
A continuación, se describen las etapas que incluyeron el desarrollo del back-end.

Creación del proyecto y configuración en el repositorio remoto GitLab

Una vez instalados los requerimientos de software antes mencionados, se procedió a la creación del proyecto con el nombre backend (fig. 20), donde se encuentra un directorio con el mismo nombre, que contiene los archivos settings.py para la configuración, url.py que, como su nombre lo indica, maneja las URL que se enviarán a las vistas, el directorio APP; donde se crearán las clases del proyecto, y el directorio venv maneja las dependencias del entorno virtual:

Figura 20.

Configuración del back-End



Después de creado y configurado el proyecto se procedió a crear y configurar repositorio remoto en GitLab para subir el proyecto, el cual también lleva el mismo nombre, y se crearon dos ramas principales Develop, donde se mezclan las ramas generadas de las tareas creadas para el desarrollo, tan pronto están probadas y aprobadas. Y la rama Máster, la cual mezcla los cambios cargados en Develop, previamente revisados y aprobados; así mismo, se comparte el repositorio con el equipo de trabajo de desarrolladores, quienes clonan el proyecto en su equipo local para trabajar en forma remota.

Planificación y desarrollo de los Sprints en el repositorio remoto GitLab

Bajo el marco de desarrollo ágil Scrum, contando ya con las historias de usuario, cuyo objetivo fue analizar y entender la funcionalidad de los usuarios junto con lo que esperan del

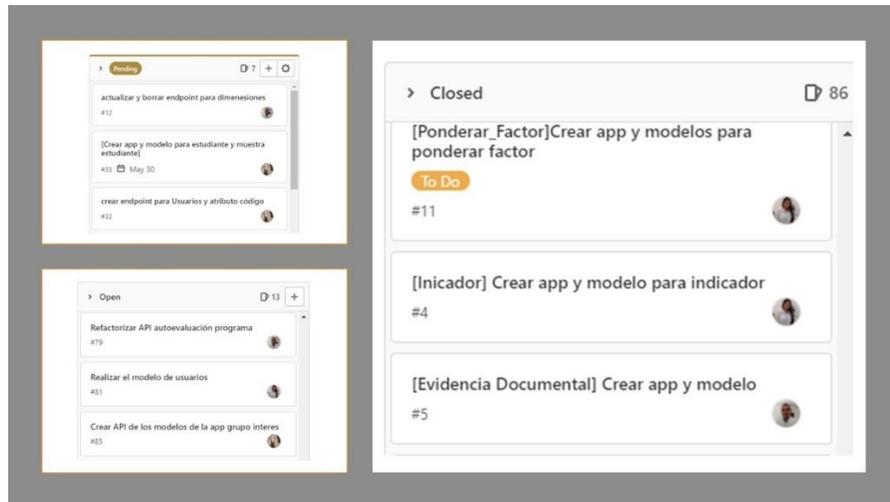
sistema, y después de configurar el proyecto, se continuó con el desarrollo de los Sprints, mediante la planificación de las tareas en el repositorio remoto de GitLab, que permite crear tareas y asignárselas a los miembros del equipo de desarrollo, para que estos creen el merge request o solicitud de fusión, lo que genera una nueva rama en el repositorio, la cual se fusiona con la rama Develop. Para acceder al repositorio remoto se hace a través de la url:

<https://gitlab.com/autoevaluacion/backend>; sin embargo, teniendo en cuenta que es un proyecto institucional, el acceso está restringido para personas externas al proyecto.

La siguiente imagen muestra el tablero de asignación de tareas y sus diferentes estados: abierto, pendiente y cerrado.

Figura 21.

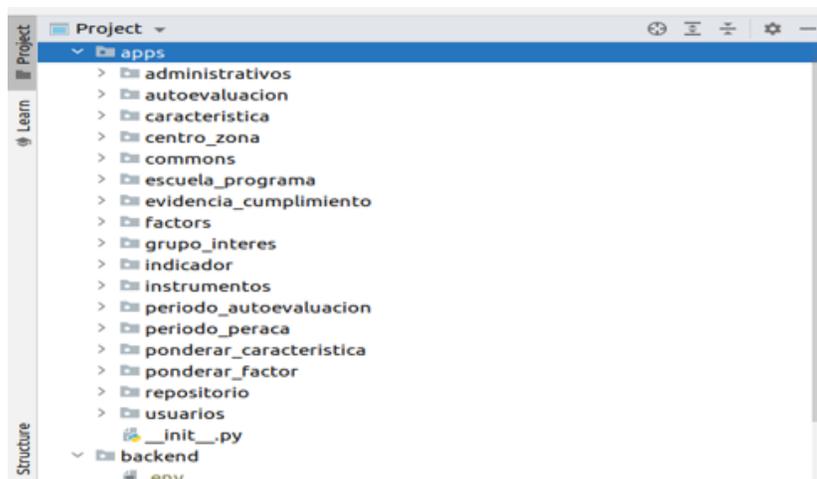
Planificación de los sprint en el repositorio

**Diseño de los modelos de las tablas maestras del negocio y el repositorio de archivos.**

“Un modelo es la fuente única y definitiva de información sobre sus datos. Contiene los campos y comportamientos esenciales de los datos que está almacenando. Generalmente, cada modelo se asigna a una sola tabla de base de datos ” (Garcia & Django Software Corporation, 2015).

Figura 22.

Modelos del proyecto autoevaluación



El sistema de información para el proceso de autoevaluación en la UNAD se compone de siete tablas maestras, a saber: Dimensión, Factor, Característica, Indicador, Evidencia de cumplimiento, programas y encuestas, las cuales son necesarias para poder automatizar dicho proceso; por tanto, para su desarrollo, una vez planificadas como tareas en el repositorio y asignadas a los miembros del equipo de desarrollo, se procedió a su elaboración localmente para subirse en GitLab (Ver anexo 6), cada tabla maestra contiene su respectiva API para probar previamente y documentar en Swagger. Igualmente, el negocio del proceso de autoevaluación involucra las tablas Autoevaluación, periodo de autoevaluación, usuarios y lo correspondientes al repositorio de archivos son archivo, atributo_tipo_documento, categoria, etiqueta_archivo y tipo_documento.

Implementación del servidor web

Para poder desplegar el backend en un servidor remoto se hizo uso de la plataforma Heroku que es un PaaS (Platform as a Service), el cual permite desplegar la aplicación sin

preocuparnos por la infraestructura, soportando además múltiples lenguajes de programación. Todos los servicios que ofrece Heroku están orientados a startups o pequeñas empresas. Para acceder al servidor se hace mediante las siguiente URL;

<http://autoevaluaciones.herokuapp.com/admin>

Desarrollo del Front-End.

El front-end es la parte de la aplicación web que interactúa con los usuarios del sistema, por tanto, está del lado del cliente, corre en el navegador y se encarga de la interactividad con los usuarios. “El objetivo es desarrollar la interfaz gráfica de usuario (GUI), buscando una experiencia de uso bien valorada por el usuario final, siendo en algunos casos necesario hacer investigación, estudios y pruebas para llegar a este fin” (Valdivia Caballero, 2016).

Tecnologías Empleadas para el front-end

La presentación de la información se hace mediante las interfaces de usuario que es manejado por el cliente mediante un navegador web; para diseñar los primeros bosquejos se utilizó Balsamiq Wireframes (Balsamiq Studios, 2008a), para el desarrollo de front-end se requirió de diversas tecnologías de desarrollo, que además de proveer la interfaz de usuario, establecieron comunicación con la API del Back-End y mostrar las respuestas a las solicitudes hechas por los usuarios.

Estas tecnologías se pueden apreciar en la tabla 29:

Tabla 29.*Tecnologías Empleadas diseño de Front-End*

Tecnología	Función
HTML 5	Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language es el componente más básico de la Web. Define el significado y la estructura del contenido web.
CSS3	Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US), describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios.
JavaScript	JavaScript (js) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. es conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.
Bootstrap Framework	Es un framework que combina CSS y JavaScript para estilizar los elementos de una página HTML utilizado en aplicaciones front-end es decir, en la pantalla de interfaz con el usuario, permite definir el componente de la interfaz con patrones predefinidos, su principal objetivo

es permitir la construcción de sitios web responsive para dispositivos móviles.

React

Es una librería open source de JavaScript para desarrollar interfaces de usuario. Fue lanzada en el año 2013 y desarrollada por Facebook, quienes también la mantienen actualmente junto a una comunidad de desarrolladores independientes y compañías.

Diseño de wireframes

Como se mencionó, el programa permite crear bocetos de las interfaces del prototipo de software; al proceso de la creación de la estructura de la interfaz se le conoce como Wireframing, el cual es un esquema, un plano, o boceto útil para ayudar a pensar y comunicarse sobre la estructura del software o sitio web que se está creando; su aspecto es de baja fidelidad porque queda claro que no es el diseño final, transmite que todo está en discusión, y deja claro que aún no se ha escrito ningún código (Balsamiq Studios, 2008b). A continuación se presentan algunos de los wireframes diseñados para el prototipo de autoevaluación.

Figura 23.*Wireframe Iniciar Sesión*

A Web Page

https://

UNAD
Universidad Nacional
Abierta y a Distancia

Se debe validar que el usuario sea el prefijo del correo institucional

Usuario

Contraseña

Ingresar

[Recuperar contraseña](#)

Al pulsar el botón ingresar debe llevar al menú de la aplicación de autoevaluación

Figura 24.*Wireframe crear Dimensión*

A Web Page

https://

Ingrese Dimensión

Identificador

Nombre Dimensión

Descripción

(POST) https://autoevaluaciones.herokuapp.com/api/factors/dimensions/

Guardar

```
{
  "identificador": "Dimensión3",
  "nombre": "cccc",
  "descripcion": "xxxxxxx"
}
```

Figura 25.
Wireframe Listar Dimensión

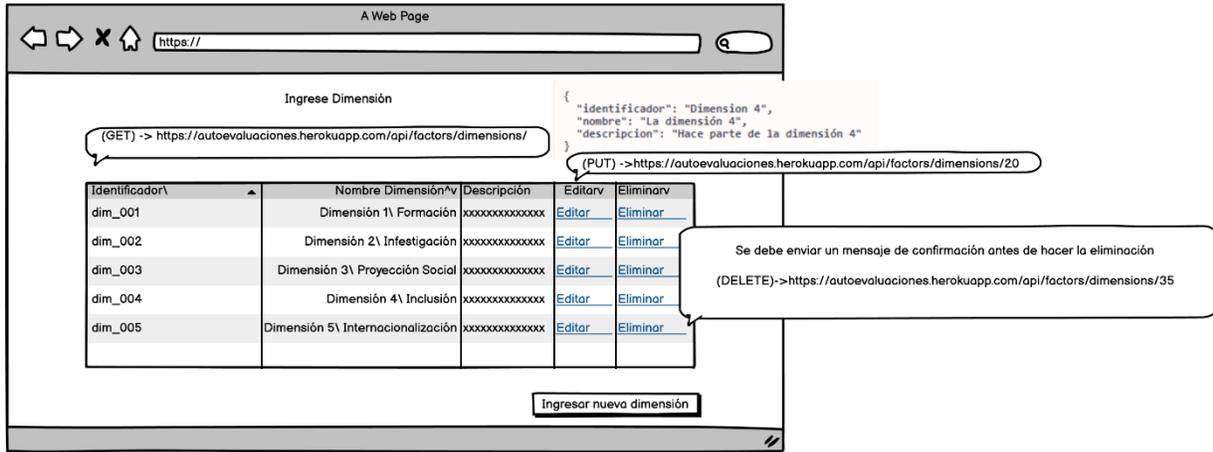


Figura 26.
Wireframe crear factor

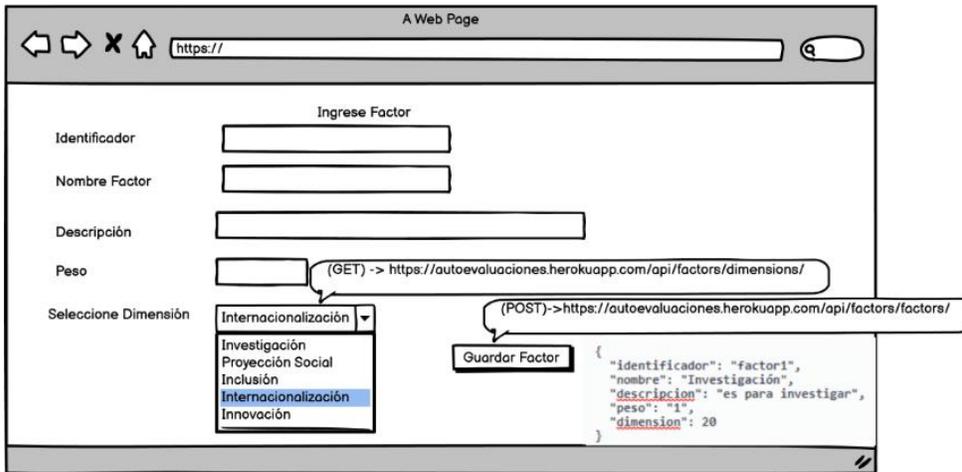


Figura 27.

Wireframe crear características

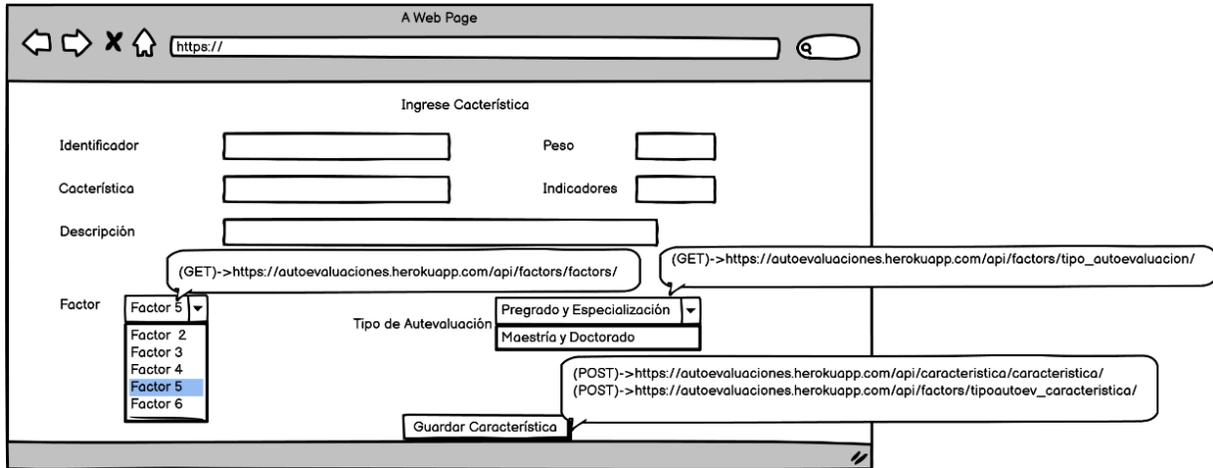


Figura 28.

Wireframe crear indicador

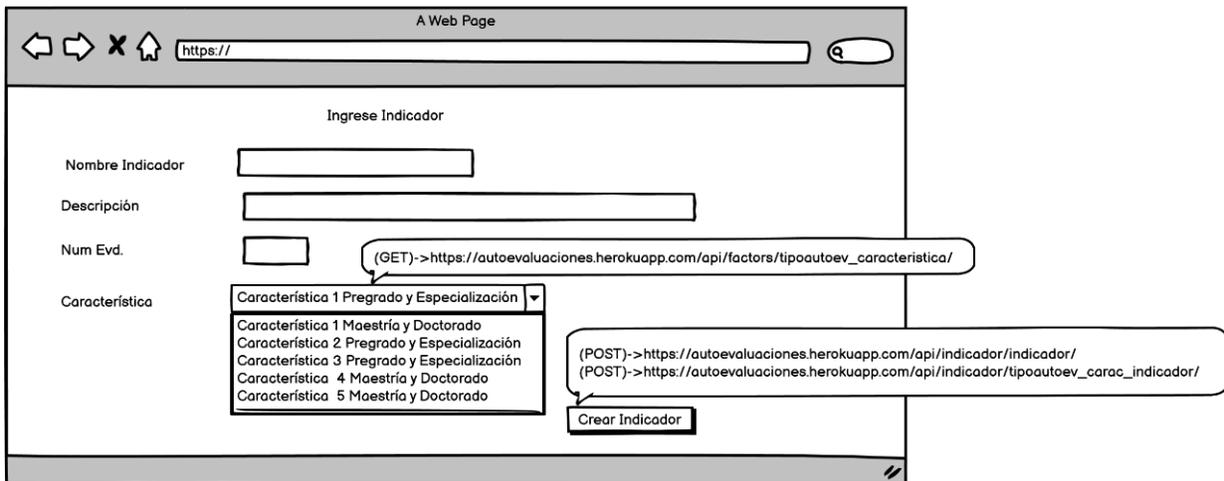
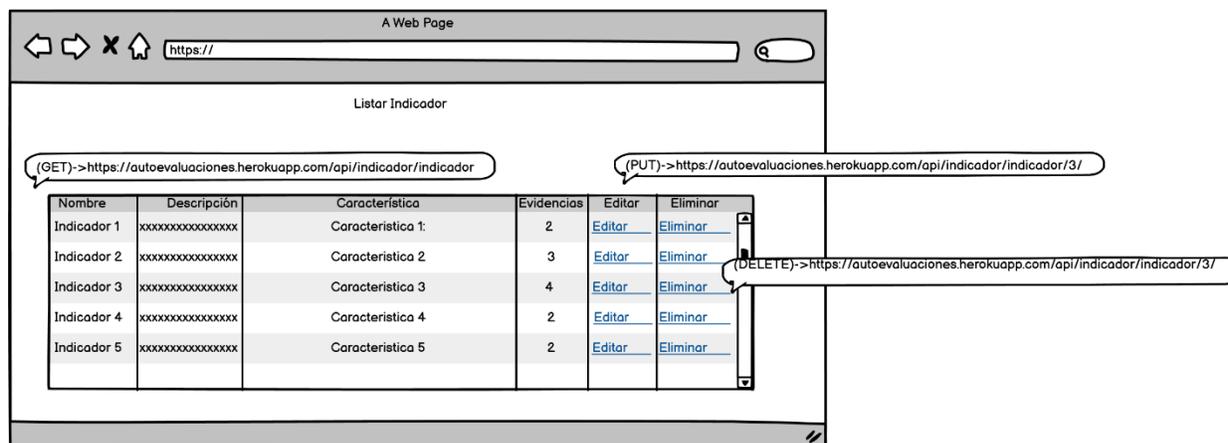
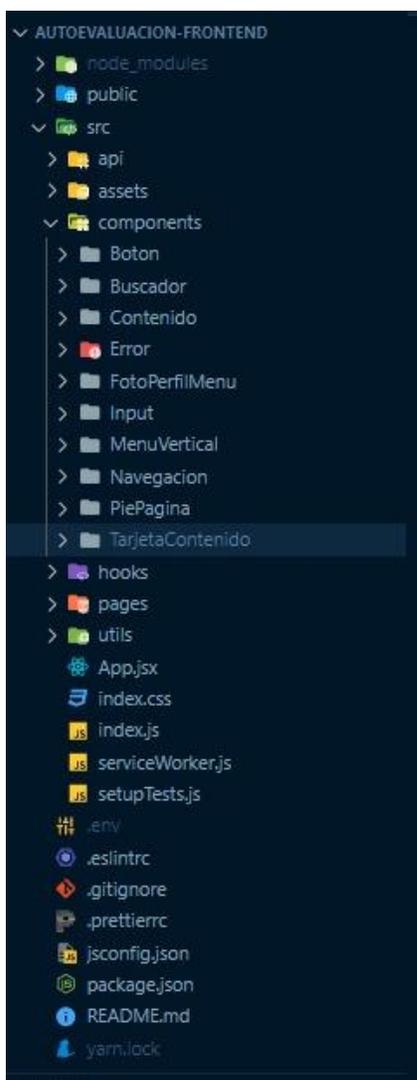


Figura 29.**Wireframe CRUD indicador****Configuración del Frontend**

Al igual que en el back-end también hay que configurar el entorno de desarrollo del frontend, como se observa en la figura 30.

Figura 30.

Configuración del front-end



Implementación del cliente web

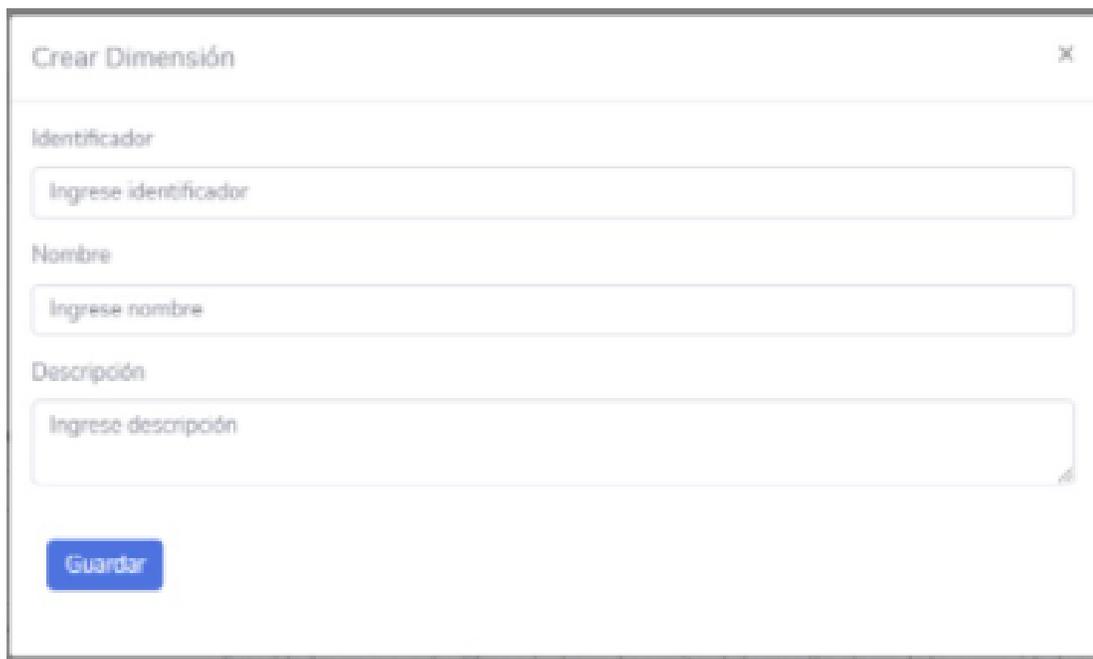
Para poder implementar el cliente se utilizó la plataforma Netlify, la cual permite automatizar proyectos web; con Deploy Previews permite experimentar cambios en cualquier

parte de su sitio sin tener que publicarlos en producción. La url para acceder al clientes es:

<https://autoevaluaciones.netlify.app/>

Figura 31.

Vista Crear Dimensión



Crear Dimensión ✕

Identificador

Nombre

Descripción

Figura 32.*Vista Crear Factor*

The image shows a web form titled "Crear Factor" with a close button (X) in the top right corner. The form contains the following fields:

- Identificador:** A text input field with the placeholder text "Ingrese identificador".
- Nombre:** A text input field with the placeholder text "Ingrese nombre".
- Descripción:** A text area with the placeholder text "Ingrese descripción".
- Peso:** A text input field with the placeholder text "Ingrese peso".
- Dimensión:** A dropdown menu with the placeholder text "Seleccione...".

At the bottom left of the form is a blue button labeled "Guardar".

Figura 33.*Vista crear característica*

Crear Característica ✕

Identificador

Característica

Descripción

Indicadores

Peso

Factor

Tipo Autoevaluación

Elaboración de pruebas de funcionalidad con usuarios

Para verificar la funcionalidad del prototipo de software se elaboraron pruebas con los usuarios responsables del proceso los cuales fueron (3) integrantes del equipo de la VISAE, a

quienes previamente se les entregó la URL del cliente web junto con las credenciales de la acceso para ingresar al sistema, y las instrucciones de las pruebas a realizar, estas consistían en validar la funcional del inicio de sesión y la funcionalidad de las opciones dimensión, factor, característica, indicador y evidencia de cumplimiento del modulo de gestión, donde por cada una se debió validar agregar, listar, editar, buscar y eliminar.

Para el registro de las pruebas correspondiente al módulo de gestión se utilizó el formato de la tabla 30 y los resultados se encuentran en el anexo 7, en ese orden de ideas los resultados fueron positivos, sin embargo, hicieron dos recomendaciones: la primera es que al listar la descripción por cuestiones de presentación se vea abreviada y no todo el texto, la segunda es que en las características se pueda observar a qué tipo de autoevaluación pertenece.

De acuerdo con lo anterior, se dio solución a la primero pero la segunda no se visualiza porque esta primera entrega solo apunta al tipo de autoevaluación pregrado y especialización y ellas informaron en reuniones previas que se van a unificar las características.

Tabla 30.*Formato Elaboración de pruebas Funcionales*

Nombre del proyecto:							
ID de caso de prueba:				Casos de prueba:			
Prueba diseñada por:				Fecha de prueba de diseño:	DD	MM	AA
Prioridad de prueba				Prueba ejecutada por:	Diana Rocío Ramírez Pérez		
	Baja	Media	Alta	Fecha de ejecución de la prueba:	DD	MM	AA
Nombre del Módulo:				Nombre del Submenú:			
Título de la prueba:							
Descripción:							
Condiciones previas:							
Dependencias:							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota	
				SI	NO		

Resultados

La implementación de la metodología BPM para el análisis del proceso de autoevaluación, permitiendo identificar en detalle sus actividades mediante el modelo AS-IS, y seguidamente establecer una propuesta de solución en el modelo TO-BE la cual involucra a un nuevo actor un sistema de información sobre el cual recaen todas aquellas tareas que pueden ser automatizadas.

Al modelar estado futuro se incorporó como actor en el proceso: **el sistema de información**, sobre el cual recaen todas las tareas que actualmente son realizadas por los actores previamente identificados, generando mejoras en el proceso, sin que esto afecte el flujo normal del mismo.

Se diseñó una arquitectura desacoplada para el proceso de autoevaluación Unadista, tomando como referencia la arquitectura de 3 capas y el patrón de arquitectura MVC, y junto a la perspectiva de puntos de vista planteado por (Rozanski et al., 2011), esta identifica los módulos, la organización entre ellos y las tareas que se espera que realicen, de la misma manera, la gestión de la información y la distribución de los diferentes elementos software en los nodos físicos requeridos para su óptima ejecución.

Se logró la automatización del proceso mediante el prototipo de una aplicación orientada a la web para la gestión del proceso de autoevaluación de la UNAD, la cual se compone en dos partes back-end y front-end, aplicando el principio “divide y vencerás” y en coherencia con el diseño de arquitectura obteniendo así bajo acoplamiento del cliente frente al servidor y una alta cohesión entre los módulos.

Se elaboraron pruebas de funcionalidad de la propuesta del prototipo de software con el equipo responsable del aseguramiento de la calidad de la institución, las cuales demostraron el cumplimiento de las actividades para las cuales fueron desarrolladas.

Observando el estado de avance del prototipo de software y las pruebas realizadas con los usuarios se ha considerado oportuno realizar el registro de los derechos de autor y patrimoniales del software (Ver anexo 8).

Conclusiones

El propósito de este proyecto fue Implementar la metodología BPM y una herramienta de software para la gestión del proceso de autoevaluación institucional y de programas académicos, como fortalecimiento a los procesos de aseguramiento de la calidad en la Universidad Nacional Abierta y a Distancia – UNAD, esto teniendo en cuenta su compleja operación y la necesidad de automatizar y optimizarlo de acuerdo con plan de desarrollo institucional 2019-2023. Los hallazgos que se observan en la revisión de literatura en los estudios relacionados con la implementación de la metodología BPM combinado con las tecnologías de la información y comunicación demostraron que facilitan la automatización de procesos relacionados con recolección de dato y evidencias para la evaluación de los programas, brindado así una oportunidad para ser aplicadas en este proyecto institucional.

Es así como el uso de la metodología BPM fue determinante para el análisis del proceso de autoevaluación, permitiendo identificar en detalle sus actividades mediante el modelo AS-IS, y seguidamente establecer una propuesta de solución en el modelo TO-BE, la cual, al implementarse, permitirá a la institución una mejora con respecto a la eficiencia y productividad del proceso, logrando su monitorización completa, visible y transparente, e igualmente permitirá la rapidez en la toma de decisiones y lógicamente con la mejora continua.

Otro objetivo fue el diseño de la arquitectura de software, la cual se realizó bajo la perspectiva de puntos de vista planteado por (Rozanski et al., 2011), en ella fue posible identificar sus módulos y la organización entre ellos, en coherencia con el patrón de arquitectura MVC, al igual que las tareas que se espera que realicen, de la misma manera, la gestión de la información y la distribución de los diferentes elementos software en los nodos físicos requeridos para su óptima ejecución, obteniéndose así una arquitectura desacoplada,

base sólida para el prototipo de software, lo que garantiza su funcionalidad, disponibilidad, seguridad, usabilidad, escalabilidad ahorro de gastos, y mantenimiento eficaz.

El último objetivo estuvo orientado al desarrollo e implementación de un prototipo de software lo que fue posible aplicando la metodología SCRUM, donde a través de las iteraciones se ha logrado avanzar a su primera versión funcional, y las primeras pruebas han manifestado aceptación y satisfacción por parte de los interesados del proyecto, lo cual demuestra que será una herramienta valiosa que a futuro sea el soporte todo el proceso de Autoevaluación Unadista.

En este documento se evidenció la necesidad analizar el proceso de autoevaluación en la UNAD e implementar una herramienta software con el ánimo de optimizar y automatizar proceso. En orden a lo anterior, se puede concluir que mediante la implementación de diferentes metodologías se logró dar un aporte a los procesos de aseguramientos de calidad, con trazabilidad, claridad y rapidez, específicamente en relación con ejercicios de autoevaluación ofreciendo así una solución pertinente, funcional, amigable, y escalable para el proceso de autoevaluación de programas en la UNAD.

Recomendaciones

Aunque la metodología BPM es propia de la gestión de procesos, es apropiado su uso para proyectos de desarrollo de software que requieran la automatización de procesos, porque como ya se aclaró en apartados anteriores, esta permite analizar en detalle el proceso y así proponer una propuesta de mejora que involucra un sistema de información garantizando su optimización.

Si bien existen diferentes catálogos y estilos de arquitectura de software, y no es camisa de fuerza utilizar un patrón determinado, el estilo por puntos de vista utilizado en este proyecto es muy recomendado para grandes sistemas de información, permitiendo obtener unas vistas arquitectónicas dando una visión global y en detalle del producto a obtener dado que asegura una cobertura amplia de preocupaciones de sus interesados.

Siendo este proyecto es bastante extenso y en coherencia con la mejora continua, se recomienda que a futuro los estudiantes de maestría que quieran vincularse a este propósito, se logre ampliar el sistema incluyendo los parámetros e indicadores de las organizaciones encargadas de procesos de acreditación internacional.

Referencias

- Arias Rondon, F. G. (2012). El proyecto de investigación Introducción a la metodología Científica. In <https://www.researchgate.net/> (6ta ed., Vol. 1). Editorial EPISTEME, C.A.
https://www.researchgate.net/publication/301894369_EL_PROYECTO_DE_INVESTIGACION_6a_EDICION
- Balsamiq Studios. (2008a). *Balsamiq for Desktop*. <https://balsamiq.com/wireframes/>
- Balsamiq Studios. (2008b). *What Are Wireframes? | Wireframing Academy | Balsamiq*.
<https://balsamiq.com/learn/articles/what-are-wireframes/>
- Blancarte Iturralde, O. J. (2020). *Introducción a la arquitectura de software - un enfoque práctico* (O. J. Blancarte Iturralde (ed.); primera).
<https://reactiveprogramming.io/books/software-architecture/es>
- Calle, X., Mayorga, F., Flores, A., & Lavín, J. (2014). *Aplicación de la metodología BPM: RAD en una institución de educación superior*. 223–234.
- Camargo, J., Otalora, J., & Alvarado, A. (2010). Todo alrededor de BPM. *Ingenio Libre*, 9.
<http://www.unilibre.edu.co/revistaingeniolibre/revista9/articulos/Todo-alrededor-de-BPM.pdf>
- Cañón Rodríguez, J. C., & Salazar Contreras, J. (2011). The quality of engineering education is a key factor in development La calidad de la educación en ingeniería: un factor clave para el desarrollo. *Ingeniería e Investigación*, 31(1 SUP), 40–50.
<https://www.redalyc.org/pdf/643/64322341002.pdf>

- Cetina Riaño, M. A. (2016). Gestión de procesos con BPM. *Tecnología Investigación y Academia*, 4(2), 45–56. <https://revistas.udistrital.edu.co/index.php/tia/article/view/8387>
- Cevallos Tito, M. P., Zabala Villarreal, W. A., & Miranda Realpe, J. H. (2018). Uso de BPM en la automatización de procesos. *SATHIRI*, 13(2), 198–218. <https://doi.org/10.32645/13906925.765>
- Cohn Mike. (2004). *User Stories Applied For Agile Software Development* (Edición Ki).
- Concejo Nacional de Acreditación CNA. (2013). *Lineamientos Para la Acreditación de Programas de Pregrado* (p. 52). Concejo Nacional de Acreditación - CNA. https://www.cna.gov.co/1741/articles-186359_pregrado_2013.pdf
- Congacha Aushay, A. E., García, V. J., & Congacha Aushay, A. E. (2017). Modelación, simulación y automatización de procesos en la gestión de servicios académicos universitarios. *3c Tecnología: Glosas de Innovación Aplicadas a La Pyme*, ISSN-e 2254-4143, Vol. 6, N°. 2, 2017, Págs. 32-51, 6(2), 32–51. <https://doi.org/10.17993/3ctecno.2017.v6n2e22.32-51>
- Const. (1991). *Constitución Política de Colombia [Const.].Artículo [67].[Titulo II]*. (Legis (ed.); 2da Ed.). [http://www.corteconstitucional.gov.co/inicio/Constitucion politica de Colombia.pdf](http://www.corteconstitucional.gov.co/inicio/Constitucion%20politica%20de%20Colombia.pdf)
- Dissanayake, N. R., & Dias, K. A. (2017, August). Web-based Applications: Extending the General Perspective of the Service of Web. *10th International Research Conference of KDU (KDU-IRC 2017) on Changing Dynamics in the Global Environment: Challenges and Opportunities*. https://www.researchgate.net/publication/319058851_Web-based_Applications_Extending_the_General_Perspective_of_the_Service_of_Web

- Django Software Foundation. (n.d.). *The Web framework for perfectionists with deadlines / Django*. Retrieved May 16, 2021, from <https://www.djangoproject.com/>
- Duro, V., & Gilart, V. (2016). *La competitividad en las instituciones de educación superior. Aplicación de filosofías de gestión empresarial: LEAN, SIX SIGMA y BUSINESS PROCESS MANAGEMENT (BPM)*. 75(3), 84–88. <https://doi.org/10.1590/0104-1169.0288.2591>
- Encode OSS Ltd. (2016). *Django REST framework*. <https://www.django-rest-framework.org/>
- Fajardo Rozo, J., & Izquierdo Portela, J. E. (2019). *Propuesta de aplicación del enfoque de gestión por procesos (BPM) en el modelado de seguimiento etapa productiva de programas teleinformática del CEET-SENA. - 10596/26801* [Universidad Nacional Abierta y a Distancia UNAD]. <https://repository.unad.edu.co/handle/10596/26801>
- Ferruzola Gómez, E., Cárdenas Rodríguez, M., & Bermeo Almeida, O. (2017). Descripción y Modelamiento de los Procesos Administrativos en la Educación Superior Mediante BPM (Modelo y Notación de Procesos de Negocio). *Atlante: Cuadernos de Educación y Desarrollo*. <https://www.eumed.net/rev/atlante/2017/09/procesos-administrativos-educacion.html>
- Fleacă, E. (2017). Core processes roadmap to deploy the higher education institution's internationalization strategy. *TEM Journal*, 6(1), 85–92. <https://doi.org/10.18421/TEM61-12>
- Flores, A., Lavín, M. J., Calle, X., & Alvarez, E. (2014). *Buscando la excelencia educativa: gestión de procesos académicos y administrativos en instituciones públicas de educación mediante BPM*. 199–209.
- Garcia, S. M., & Django Software Corporation. (2015). *La guía definitiva de Django: Desarrolla aplicaciones web de forma rápida y sencilla*.

http://80.30.58.16/files/1501150999_guiaDefinitivaDjango.pdf

Garimella, K., Lees, M., & Williams, B. (2011). BPM (GERENCIA DE PROCESOS DE NEGOCIO). In *Circulation research* (Vol. 13). <https://doi.org/10.1161/01.RES.13.6.537>

JetBrains s.r.o. (2020). *PyCharm: el IDE de Python para desarrolladores profesionales, por JetBrains*. <https://www.jetbrains.com/es-es/pycharm/>

Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews*. <http://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf>

Ministerio de Educación Nacional - MEN-. (2010). *LINEAMIENTOS GENERALES PARA LA AUTOEVALUACION EN EL MARCO DEL ARTÍCULO 6, NUMERAL 6.3 DEL DECRETO 1295 DE 2010 - Sistemas información*. <https://www.mineduccion.gov.co/sistemasdeinformacion/1735/article-298334.html>

Ministerio de Educación Nacional -MEN-. (2019a). *DECRETO 1330 DE 2019 Por el cual se sustituye el Capítulo 2 y se suprime el Capítulo 7 del Título 3 de la Parte 5 del Libro 2 del Decreto 1075 de 2015 -Único Reglamentario del Sector Educación*.

Ministerio de Educación Nacional -MEN-. (2019b). *Sistema de aseguramiento de la calidad de la educación superior*. <https://www.mineduccion.gov.co/Portal/>. https://www.mineduccion.gov.co/1759/w3-article-235585.html?_noredirect=1

Ministerio. de las Tecnologías. de la Información. y Comunicación -MINTIC- (2018). *Manual para la implementación de Gobierno Digital Dirigido a entidades públicas del orden nacional y territorial*. http://www.archivogeneral.gov.co/sites/default/files/2018-05/articles-73081_documento.pdf

Molina Ríos, J. R., Loja Mora, N. M., Zea Ordóñez, M. P., & Loaiza Sojos, E. L. (2016).

Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python. *Revista Latinoamericana de Ingeniería de Software*, 4(4), 201.

<https://doi.org/10.18294/relais.2016.201-207>

NEXTECH. (2020). *¿Qué es BPMN y para qué sirve?* <https://nextech.pe/que-es-bpmn-y-para-que-sirve/>

Organización para la Cooperación y el Desarrollo Económico OCDE. (2013). *Metodología propuesta para la definición de la Investigación y el Desarrollo Experimental Manual de Frascati*. 286. www.oecdbookshop.org www.oecd-ilibrary.org www.oecd.org/oecddirect

Peña Hidalgo, H. J., & López Ortiz, I. A. (2018). *Modelado de procesos de negocio para la gestión de la oferta educativa en sus procedimientos: gestión de nuevos registros calificados y gestión para la renovación de registro calificado de los programas académicos de la ECBTI de la UNAD*. - 10596/23455 [Universidad Nacional Abierta y a Distancia UNAD]. <https://repository.unad.edu.co/handle/10596/23455>

Pimenta Rioga, D. do C., & Abrantes Baracho, R. M. (2018). The BPMN as a tool for the development of the information flow of the reception sector and welcome program of Minas Gerais Federal University. *IMCIC 2018 - 9th International Multi-Conference on Complexity, Informatics and Cybernetics, Proceedings*, 2(Imcic), 139–144. <http://www.iiis.org/CDs2018/CD2018Spring/papers/ZA792VF.pdf>

Python Software Foundation. (2020). *drf-yasg · PyPI*. <https://pypi.org/project/drf-yasg/>

Rodríguez Ríos, C. Y., Gutiérrez Oviedo, A. J., & Lasso Buesaquillo, J. A. (2020). *Vista de*

EXPERIENCIA DE UNA PRÁCTICA REAL EN UNA ORGANIZACIÓN, APLICANDO EL CICLO DE VIDA BPM AL PROCESO CRÍTICO DE UNA EMPRESA DE FLORES, PARA AFIANZAR EL CONOCIMIENTO DE LOS PROCESOS Y GENERAR UNA CULTURA BPM. <https://acofipapers.org/index.php/eiei/article/view/783/788>

Rozanski, N., Woods Marks, E., & Plc, S. (2011). *Applying Viewpoints and Views to Software Architecture.* <http://www.viewpoints-and-perspectives.info>

Schwaber, K., & Sutherland, J. (2020). *La Guía Definitiva de Scrum.*

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Shorokhov, S. G., & Khaptakhanova, V. V. (2019). *Web based application for operational loss collection and value-at-risk and expected shortfall calculation.* <http://ceur-ws.org>

SmartBear Software. (2021). *API Documentation & Design Tools for Teams | Swagger.* <https://swagger.io/>

Software Freedom Conservancy. (n.d.). *GitLab Pricing | GitLab.* Retrieved May 16, 2021, from <https://about.gitlab.com/pricing/#gitlab-com>

UNAD. (2017). *F-2-1-3 Procedimiento de Autoevaluación de Programas de Educación Superior.* Universidad Nacional Abierta y a Distancia - UNAD.

UNAD Universidad Nacional Abierta y a Distancia. (2019a). *Plan de Desarrollo 2019 - 2023 MÁS UNAD, MÁS PAÍS.*

https://informacion.unad.edu.co/images/planeacion/2020/PLAN_DESARROLLO_2019_-_2023_-_V2_F_compressed.pdf

UNAD Universidad Nacional Abierta y a Distancia. (2021). *PROCEDIMIENTO: GESTIÓN DE LA AUTOEVALUACIÓN DE PROGRAMAS* ". <https://sig.unad.edu.co>

UNAD Universidad Nacional Abierta y a Distancia (2011). *Modelo de autoevaluación y acreditación unadista*.

UNAD Universidad Nacional Abierta y a Distancia (2019b). *Plan de Desarrollo 2019-2020*.
<https://informacion.unad.edu.co/images/PLAN-DESARROLLO-2019-2023-f.pdf>

Estatuto Académico, (2020). www.unad.edu.co

Valdivia Caballero, J. J. (2016). Modelo de procesos para el desarrollo del front-end de aplicaciones web. *Universidad de Lima*.
<https://revistas.ulima.edu.pe/index.php/Interfases/article/view/1245/1205>

Anexos

Anexo 1 Ficha Ponderación de Características

PONDERACIÓN CARACTERÍSTICAS				
PROGRAMA XXXXXXXXXXXXXXXXXXXX				
	Peso Factor	CARACTERÍSTICAS	Peso Características	Peso Características Ajustado
FACTOR 1	0,08	<i>Característica 1: Coherencia y pertinencia de la Misión</i>	0,3333	0,3
		<i>Característica 2. Proyecto Educativo del Programa.</i>	0,3333	0,3
		<i>Característica 3. Relevancia académica y pertinencia social del programa.</i>	0,3400	0,3
		TOTAL	1,0066	1,0
FACTOR 2	0,07	<i>Característica 4: Mecanismos de ingreso.</i>	0,3330	0,3
		<i>Característica 5. Número y Calidad de estudiantes admitidos</i>		0,0
		<i>Característica 6: Competencias.</i>		0,0
		<i>Característica 7: Reglamento estudiantil y Académico</i>		0,0
TOTAL		0,3330	0,3	
FACTOR 3	0,07	<i>Característica 8: Selección, Vinculación y Permanencia de los profesores</i>		0,0
		<i>Característica 9: Estatuto Profesor</i>		0,0
		<i>Característica 10: Número, dedicación, nivel de formación y experiencia de los profesores</i>		0,0
		<i>Característica 11: Desarrollo Profesor</i>		0,0
		<i>Característica 12: Estímulos a la docencia, investigación, extensión o proyección social y a la cooperación internacional</i>		0,0
		<i>Característica 13: Producción, pertinencia y utilización e impacto de material docente</i>		0,0
		<i>Característica 14: Remuneración por méritos</i>		0,0
		<i>Característica 15: Evaluación de los profesores</i>		0,0
<i>Característica 16: Interacción académica de los profesores</i>			0,0	

	TOTAL		0,0000	0,0
FACTOR 4	0,07	<i>Característica 17: Egresados e Institución</i>		0,0
		<i>Característica 18: Seguimiento a Egresados</i>		0,0
		<i>Característica 19: Impacto del egresado en el medio social y académico</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 5	0,08	<i>Característica 20: Modelo Pedagógico</i>		0,0
		<i>Característica 21: Impacto del Modelo Pedagógico Unadista en el medio social y académico</i>		0,0
		<i>Característica 22: Integralidad del Currículo</i>		0,0
		<i>Característica 23: Flexibilidad del Currículo</i>		0,0
		<i>Característica 24: Interdisciplinariedad</i>		0,0
		<i>Característica 25: Estrategias de enseñanza – Aprendizaje</i>		0,0
		<i>Característica 26: Sistema de evaluación a estudiantes</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 6	0,07	<i>Característica 28: Planificación pedagógica</i>		0,0
		<i>Característica 29: Diseño académico</i>		0,0
		<i>Característica 30: Planificación didáctica</i>		0,0
		<i>Característica 31: Planificación de la gestión tecno pedagógica</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 7	0,07	<i>Característica 32: Recursos de apoyo académico y bibliográfico</i>		0,0
		<i>Característica 33: Recursos físicos</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 8	0,07	<i>Característica 34: Infraestructura de información y comunicación</i>		0,0
		<i>Característica 35: Infraestructura de Campus Virtual</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 9	0,07	<i>Característica 36: Formación para la Investigación, Innovación</i>		0,0
		<i>Característica 37: Compromiso con la investigación</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 10	0,06	<i>Característica 38: Inserción del programa en contextos académicos nacionales e internacionales</i>		0,0
		<i>Característica 39: Relaciones externas de profesores y estudiantes</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 11	0,08	<i>Característica 40: Institución, programa y entorno</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 12	0,06	<i>Característica 41: Políticas, programas y servicios de Bienestar Institucional</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 13	0,05	<i>Característica 42: Sistemas de Evaluación y autorregulación</i>		0,0

	TOTAL		0,0000	0,0
FACTOR 14	0,05	<i>Característica 43: Administración, Gestión y funciones institucionales</i>		0,0
		<i>Característica 44: Dirección del programa</i>		0,0
	TOTAL		0,0000	0,0
FACTOR 15	0,05	<i>Característica 45: Fuentes de financiación y patrimonio del programa</i>		0,0
		<i>Característica 46: Presupuesto del programa</i>		0,0
		<i>Característica 47: Administración de recursos</i>		0,0
	TOTAL		0,0000	0,0

Anexo 3 Ficha de Seguimiento Aplicación de Instrumentos

SEGUIMIENTO APLICACIÓN INSTRUMENTOS AUTOEVALUACIÓN

PROGRAMA XXXXXXXXXXXXXXXXX

ESTAMENTO	MUESTRA POR ZONAS																										
	AMAZONIA Y ORINOQUIA			CARIBE			CENTRO, BOGOTÁ Y C/MARCA			CENTRO BOYACÁ			CENTRO ORIENTE			CENTRO SUR			OCCIDENTE			SUR			T O T A L E S		
	Calculadas	Aplicadas	Faltantes	Calculadas	Aplicadas	Faltantes	Calculadas	Aplicadas	Faltantes	Calculadas	Aplicadas	Faltantes	Calculadas	Aplicadas	Faltantes	Calculadas	Aplicadas	Faltantes	Calculadas	Aplicadas	Faltantes	Calculadas	Aplicadas	Faltantes	CALCULADAS	APLICADAS	FALTANTES
Estudiantes	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0	0	0
Egresados	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0	0	0
Profesores	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0	0	0
Expansores	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0	0	0
Directivos	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0	0	0
Administrativos	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0		0	0	0	0
T O T A L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Digite la información en los recuadros amarillos según la información del número de personas por estamento que has diligenciado la encuesta.

Anexo 4 Matriz Apreciación de Estamentos

EJERCICIO DE AUTOEVALUACIÓN - APRECIACIÓN DE LOS ESTAMENTOS						
					4,0	
Factor		Calificación Factor	Característica		Estamento	Calificación Característica
F1	Misión y proyecto académico pedagógico solidario	4,4	C1	Coherencia y pertinencia de la Misión.	Director	4,6
					Estudiante	3,9
					Profesor	4,6
					Administrativo	4,5
					Egresado	4,2
			C3	Relevancia académica y pertinencia social del programa	Director	4,2
Empresario	4,8					
F2	Estudiantes	3,9	C5	Número y Calidad de estudiantes admitidos	Profesor	4,1
					Estudiante	3,7
			C7	Reglamento estudiantil y Académico	Director	4,0
					Profesor	4,2
					Estudiante	3,6
			F3	Profesores	3,9	C8
Profesor	4,2					
Estudiante	3,8					
C9	Estatuto Profesorial	Director				4,2
		Profesor				4,0
		Estudiante				3,7
C10	Número, dedicación, nivel de formación y experiencia de los profesores	Director				4,6
		Profesor				3,9
		Estudiante				3,4
C12	Estímulos a la docencia, investigación, extensión o proyección social y a la cooperación internacional.	Director	4,2			
		Profesor	4,0			
C13	Producción, pertinencia y utilización e impacto de material docente.	Estudiante	3,5			
C15	Evaluación de los profesores	Profesor	4,1			

			C16	Interacción académica de los profesores	Director	3,0
					Estudiante	3,5
					Profesor	4,0
					Egresado	3,6
F4	Egresados	3,9	C17	Egresados e Institución	Egresado	3,5
			C19	Impacto del egresado en el medio social y académico	Egresado	3,7
					Empresario	4,5
F5	Modelo Pedagógico	4,2	C20	Modelo Pedagógico	Director	4,4
					Profesor	4,4
					Estudiante	3,7
			C21	Impacto del Modelo Pedagógico Unadista en el medio social y académico	Estudiante	3,9
			C22	Integralidad del Currículo	Director	4,4
					Profesor	4,5
					Estudiante	3,9
			C23	Flexibilidad del Currículo	Director	4,6
					Profesor	4,4
					Estudiante	3,9
			C24	Interdisciplinariedad	Director	4,6
					Profesor	4,4
					Estudiante	3,8
			C25	Estrategias de enseñanza – Aprendizaje	Director	4,3
					Profesor	4,3
					Estudiante	3,8
			C26	Sistema de evaluación a estudiantes	Estudiante	3,9
			C27	Trabajo de los estudiantes	Director	3,6
					Profesor	4,0
F6	Planificación y diseño	4,1	C30	Planificación didáctica	Profesor	4,1
F7	Medios, Mediaciones y Escenarios de práctica	3,8	C32	Recursos de apoyo académico y bibliográfico	Profesor	3,9
					Estudiante	3,5
			C33	Recursos físicos	Profesor	4,1
					Estudiante	3,6
F8	Infraestructura Tecnológica	4,0	C34	Infraestructura de información y comunicación	Director	4,4
					Profesor	4,1
					Estudiante	3,4

			C35	Infraestructura de Campus Virtual	Director	4,2
					Profesor	4,1
					Estudiante	3,8
F9	Investigación	3,9	C36	Formación para la Investigación, Innovación	Profesor	4,2
					Estudiante	3,6
F10	Visibilidad Nacional e Internacional	3,9	C39	Relaciones externas de profesores y estudiantes	Profesor	3,8
					Estudiante	3,7
					Empresario	4,3
F11	Pertinencia e Impacto Social	4,4	C40	Institución, programa y entorno	Empresario	4,4
F12	Bienestar Institucional	4,0	C41	Políticas, programas y servicios de Bienestar Institucional	Director	3,8
					Estudiante	3,8
					Profesor	4,2
					Administrativo	4,3
F13	Autoevaluación y autorregulación	4,1	C42	Sistemas de Evaluación y autorregulación	Director	4,2
					Profesor	4,3
					Estudiante	3,8
F14	Organización, Gestión y Administración	4,1	C43	Administración, Gestión y funciones institucionales	Director	4,4
					Estudiante	3,7
					Profesor	4,2
					Administrativo	4,0
			C44	Dirección del programa	Director	4,2
					Estudiante	3,6
					Profesor	4,1
					Administrativo	4,3
F15	Recursos financieros	3,9	C45	Fuentes de financiación y patrimonio del programa	Director	4,0
					Estudiante	3,6
					Profesor	4,0
					Administrativo	4,0
			C46	Presupuesto del programa	Director	3,9
					Profesor	4,0

CNA	ESCALA	
A: Se cumple plenamente.	4,50	5,00
B: Se cumple en alto grado.	4,00	4,49
C: Se cumple aceptablemente.	3,00	3,99
D: Se cumple insatisfactoriamente.	2,00	2,99
E: No se cumple.	1,00	1,99

Anexo 5 Ficha Análisis Documental

Herramienta para análisis documental

TIPO DE DOCUMENTO		ATRIBUTOS PARA CALIFICAR			
REGULATORIOS	Normas	Existencia	Pertinencia	Grado Apropiación	Grado de Aplicación
Un documento regulatorio, en un documento donde se establecen normas, reglas o leyes dentro de un determinado ámbito, cuyo objetivo es mantener un orden, llevar un control y garantizar los derechos de todos los integrantes de una comunidad.	Reglamentos				
	Acuerdos				
	Estatutos				
	Políticas				
	Procedimientos				

TIPO DE DOCUMENTO		ATRIBUTOS PARA CALIFICAR		
DESCRIPTIVOS	Informes	Existencia	Precisión/ Exactitud	Pertinencia
Un documento descriptivo es un Escrito que informa sobre el comportamiento de un hecho o situación determinado.	Estadísticos			
	Audiovisuales			

TIPO DE DOCUMENTO		ATRIBUTOS PARA CALIFICAR	
SOPORTE	Actas	Existencia	Suficiencia
Un documento de soporte es un documento oficial que respalda las declaraciones específicas de la aplicación de las normas y procesos, con los cuales se lleva el control y dan veracidad a las decisiones tomadas en la institución	Convenios y contratos		
	Bases datos, facturas, presupuestos		
	Planos		
	Syllabus		
	Encuestas		

ESCALA DE CALIFICACIÓN	
4.0 - 5.0	Se cumple plenamente
3.5 - 3.99	Se cumple
3.0 - 3.49	Se cumple parcialmente
1.0 - 2.99	No se cumple

Anexo 6 Código Fuente

Factor.py

```

from django.core.validators import MinValueValidator, MaxValueValidator
from django.db import models
from django.utils.translation import ugettext_lazy as _
from apps.common.models import BaseModel
from . import Dimension
class Factor(BaseModel):
    identificador = models.CharField(
        verbose_name=_('Identificador'),
        max_length=100,
        null=False,
        blank=False
    )
    nombre = models.CharField(
        verbose_name=_('Nombre'),
        max_length=100,
        unique=True)
    descripcion = models.TextField(
        max_length=800,
        verbose_name=_('Descripción'),
        null=False,
        blank=False)
    peso = models.DecimalField(
        verbose_name=_('Peso'),
        max_digits=3,
        decimal_places=2,
        null=False,
        blank=False,
        validators=[
            MinValueValidator(0.009),
            MaxValueValidator(0.99)
        ]
    )
    dimension = models.ForeignKey(
        Dimension,
        on_delete=models.CASCADE,
        verbose_name=_('Dimensión'))
    def __str__(self):
        return self.identificador
class Meta:
    db_table = 'factors_factors'
    verbose_name = _('Factor')
    verbose_name_plural = _('Factores')
    unique_together = [['nombre', 'descripcion', 'is_active'], ['identificador', 'is_active']]

```

dimension.py

```

from django.db import models
from django.utils.translation import ugettext_lazy as _

from apps.common.models import BaseModel

class Dimension(BaseModel):
    """ Este es el modelo de dimesion """
    identificador = models.CharField(
        verbose_name=_('Identificador'),
        max_length=100,
        null=False,
        blank=False
    )

    nombre = models.CharField(
        max_length=20,
        verbose_name=_('Nombre'),
        unique=True)

    descripcion = models.TextField(
        max_length=800,
        verbose_name=_('Descripción'),
        null=False,
        blank=False)

    def __str__(self):
        return self.identificador

    class Meta:
        db_table = 'dimension_dimensions'
        verbose_name = _('Dimensión')
        verbose_name_plural = _('Dimensiones')
        unique_together = ['nombre', 'descripcion', 'is_active']
        unique_together = ['identificador', 'is_active']

```

caracteristica.py

```

from django.core.validators import MinValueValidator, MaxValueValidator
from django.db import models
from django.utils.translation import ugettext_lazy as _
from apps.common.models import BaseModel
from apps.factors.models import Factor

```

```
class Caracteristica(BaseModel):
    identificador = models.CharField(
        verbose_name=_('Identificador'),
        max_length=100,
        null=False,
        blank=False
    )

    nombre = models.CharField(
        verbose_name=_('Nombre'),
        max_length=100,
    )

    descripcion = models.TextField(
        verbose_name=_('Descripción'),
        null=False,
        blank=False
    )

    indicadores = models.PositiveSmallIntegerField(
        verbose_name=_('Indicadores'),
        null=False,
        blank=False,
        validators=[
            MinValueValidator(1)
        ]
    )

    peso = models.DecimalField(
        verbose_name=_('Peso'),
        max_digits=3,
        decimal_places=2,
        null=False,
        blank=False,
        validators=[
            MinValueValidator(0.009),
            MaxValueValidator(0.99)
        ]
    )

    factor = models.ForeignKey(
        Factor,
        on_delete=models.CASCADE,
```

```

        verbose_name=_('Factor'))

tipo_autoevaluacion = None

def __str__(self):
    return self.identificador

class Meta:
    db_table = 'caracteristica_caracteristica'
    verbose_name = _('Caracteristica')
    verbose_name_plural = _('Caracteristicas')
    unique_together = ['identificador', 'nombre', 'descripcion', 'is_active']

```

indicador.py

```

from django.db import models
from django.core.validators import MinValueValidator
from django.utils.translation import ugettext_lazy as _
from apps.commons.models import BaseModel
from apps.caracteristica.models import Caracteristica

```

```

class Indicador(BaseModel):
    """ Este es el modelo de indicador """
    nombre = models.CharField(
        verbose_name=_('Nombre'),
        max_length=100,
        unique=False,
        blank=False,
        null=False
    )

    descripcion = models.TextField(
        verbose_name=_('Descripción'),
        max_length=350,
        blank=False,
        null=False
    )

    evidencias = models.PositiveSmallIntegerField(
        verbose_name=_('Evidencias'),
        null=False,
        blank=False,
        validators=[
            MinValueValidator(1)
        ]
    )

```

```

)

caracteristica = models.ForeignKey(
    Caracteristica,
    on_delete=models.CASCADE,
    verbose_name=_('Caracteristica'))

def __str__(self):
    return self.nombre

class Meta:
    db_table = 'indicador_indicadores'
    verbose_name = _('Indicador')
    verbose_name_plural = _('Indicadores')
    unique_together = ['nombre', 'descripcion', 'is_active']

```

evidencia_cumplimiento.py

```

from django.db import models
from django.utils.translation import ugettext_lazy as _

from apps.common.models import BaseModel
from ...indicador.models import Indicador

class EvidenciaCumplimiento(BaseModel):

    nombre = models.CharField(
        verbose_name=_('Nombre'),
        max_length=100,
        unique=False,
        blank=False,
        null=False
    )

    evidencia = models.TextField(
        verbose_name=_('Evidencia'),
        null=False,
        blank=False
    )

    indicador = models.ForeignKey(
        Indicador,
        on_delete=models.CASCADE,

```

```

        verbose_name=_('Indicador'))

    def __str__(self):
        return self.nombre
        return self.evidencia

    class Meta:
        db_table = 'evidencia_cumplimiento_evidencias_cumplimiento'
        verbose_name = _('Evidencia cumplimiento')
        verbose_name_plural = _('Evidencias de cumplimiento')
        unique_together = ['nombre', 'evidencia', 'indicador', 'is_active']

apps/escuela_programa/models/escuela.py

from django.db import models
from django.utils.translation import ugettext_lazy as _

from apps.commons.models import BaseModel

class Escuela(BaseModel):
    """ Este es el modelo de escuela """
    nombre = models.CharField(
        max_length=100,
        verbose_name=_('Nombre'),
        unique=True)

    codigo = models.IntegerField(
        verbose_name=_('Código'),
        null=False,
        blank=False)

    decano = models.CharField(
        max_length=100,
        verbose_name=_('Nombre del Decano'),
        unique=True)

    def __str__(self):
        return self.nombre

    class Meta:
        db_table = 'escuela_escuela'
        verbose_name = _('Escuela')
        verbose_name_plural = _('Escuelas')

```

```

apps/escuela_programa/models/programa.py

from django.db import models
from django.utils.translation import ugettext_lazy as _

from apps.commons.models import BaseModel
from . import Escuela

formacion = (
    ('Tecnológico', 'Tecnológico'),
    ('Profesional', 'Profesional'),
    ('Especialización', 'Especialización'),
    ('Maestría', 'Maestría'),
    ('Doctorado', 'Doctorado'),
)

metodologia_formacion = (
    ('Distancia', 'Distancia'),
    ('Virtual', 'Virtual'),
)

num_periodos = (
    ('1', '1'),
    ('2', '2'),
    ('3', '3'),
    ('4', '4'),
    ('5', '5'),
    ('6', '6'),
    ('7', '7'),
    ('8', '8'),
    ('9', '9'),
    ('10', '10'),
    ('11', '11'),
    ('12', '12'),
)

class Programa(BaseModel):

    escuela = models.ForeignKey(
        Escuela,
        on_delete=models.CASCADE,

```

```

    verbose_name=_('Escuela'))

    snies = models.IntegerField(
        verbose_name=_('Código SNIES'),
        null=False,
        blank=False)

    nombre = models.CharField(
        verbose_name=_('Nombre'),
        max_length=200,
        unique=True)

    nivel_formacion = models.CharField(
        verbose_name=_('Nivel de Formación'),
        max_length=50,
        choices=formacion)

    metodologia = models.CharField(
        verbose_name=_('Metodología'),
        max_length=50,
        choices=metodologia_formacion)

    creditos = models.IntegerField(
        verbose_name=_('Total de Créditos'),
        null=False,
        blank=False)

    no_periodos = models.CharField(
        verbose_name=_('Número de Peridos Académicos'),
        max_length=50,
        choices=num_periodos)

    programa_acreditado = models.BooleanField(
        verbose_name=_('Programa Acreditado'),
        null=False,
        blank=False
    )

    def __str__(self):
        return self.nombre

    class Meta:
        db_table = 'programa_programas'
        verbose_name = _('Programa')
        verbose_name_plural = _('Programas')
        unique_together = ['nombre', 'is_active']

```

apps/escuela_programa/models/acdo_creacion.py

```
from django.db import models
from django.utils.translation import ugettext_lazy as _
from apps.common.models import BaseModel
from . import Programa
```

```
class Acdo_Creacion(BaseModel):
```

```
    programa = models.ForeignKey(
        Programa,
        on_delete=models.CASCADE,
        verbose_name=_('Programa'))
```

```
    resolucion = models.IntegerField(
        verbose_name=_('Resolución'),
        unique=True)
```

```
    fecha_creacion = models.DateField(
        verbose_name=_('Fecha de Creación'),
        auto_now=False,
        auto_now_add=False,
        null=True)
```

```
    organo = models.CharField(
        verbose_name=_('Organo'),
        max_length=20,
    )
```

```
    def __str__(self):
        return f'{self.resolucion}'
```

```
class Meta:
    db_table = 'acdo_creacion'
    verbose_name = _('acuerdo de creación')
    verbose_name_plural = _('acuerdos de creación')
    unique_together = ['resolucion', 'is_active']
    unique_together = ['programa', 'is_active']
```

apps/escuela_programa/models/reg_calificado.py

```
from django.db import models
from django.utils.translation import ugettext_lazy as _
from apps.common.models import BaseModel
```

```

from . import Programa

class Reg_Calificado(BaseModel):

    programa = models.ForeignKey(
        Programa,
        on_delete=models.CASCADE,
        verbose_name=_('Programa'))

    resolucio = models.IntegerField(
        verbose_name=_('Resoluci3n'),
        unique=True)

    fecha_creacion = models.DateField(
        verbose_name=_('Fecha de Creaci3n'),
        auto_now=False,
        auto_now_add=False,
        null=True)

    fecha_vencimiento = models.DateField(
        verbose_name=_('Fecha de Vencimiento'),
        auto_now=False,
        auto_now_add=False,
        null=True)

    a1os = models.IntegerField(
        verbose_name=_('a1os'),
        unique=False,
        null=False,
        )

    activo = models.BooleanField(

    )

    def __str__(self):
        return f'{self.resolucio}'

    class Meta:
        db_table = 'reg_calificado'
        verbose_name = _('Registro Calificado')
        verbose_name_plural = _('Registros Calificados')
        unique_together = ['resolucio', 'is_active']

```

apps/escuela_programa/models/acdo_creacion.py

```
from django.db import models
from django.utils.translation import ugettext_lazy as _
from apps.common.models import BaseModel
from . import Programa

class Acdo_Creacion(BaseModel):

    programa = models.ForeignKey(
        Programa,
        on_delete=models.CASCADE,
        verbose_name=_('Programa'))

    resolucion = models.IntegerField(
        verbose_name=_('Resolución'),
        unique=True)

    fecha_creacion = models.DateField(
        verbose_name=_('Fecha de Creación'),
        auto_now=False,
        auto_now_add=False,
        null=True)

    organo = models.CharField(
        verbose_name=_('Organo'),
        max_length=20,
        )

    def __str__(self):
        return f'{self.resolucion}'

class Meta:
    db_table = 'acdo_creacion'
    verbose_name = _('acuerdo de creación')
    verbose_name_plural = _('acuerdos de creación')
    unique_together = ['resolucion', 'is_active']
    unique_together = ['programa', 'is_active']
```

Anexo 7 Formato de validación de pruebas –Usuario VISAE-Calidad

Caso de Prueba navegación inicio de sesión Iniciar Sesión

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación									
ID de caso de prueba: BU_001				Casos de prueba: Probar funcionalidad iniciar sesión					
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA	
						17	11	2021	
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez				
			X				Fecha de ejecución de la prueba:		DD
							26	11	2021
Nombre del Módulo: Interfaz Inicio de Sesión				Nombre del Submenú: NA					
Título de la prueba: Probar funcionalidad inicio de sesión									
Descripción: Verifique que al ingresar sus credenciales de acceso ingresa al sistema									
Condiciones previas: El usuario tiene un nombre de usuario y contraseña válidos									
Dependencias:									
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota			
				SI	NO				
1	Ingresar a la página de inicio de sesión	https://autoevaluaciones.netlify.app/	El usuario debe poder iniciar sesión	X					
2	Ingresar el usuario de intranet	visae.calidad	Se puede ingresar credencial	X					
3	Ingresar contraseña	*****	Se puede ingresar credencial	X					
4	Hacer clic en el botón de Iniciar sesión		Usuario registrado	X					

Casos de Prueba de navegación panel de Gestión opción Dimensión Crear Dimensión

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación									
ID de caso de prueba: BU_002				Título de la prueba: Probar funcionalidad crear Dimensión					
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		17	11	21	
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez				
				X			Fecha de ejecución de la prueba:		DD
							26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Dimensión					
Descripción: Verificar que se pueda crear una nueva dimensión									
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión									
Dependencias:									

Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCID E		Nota
				SI	NO	
1	Acceder al módulo de gestión e ir a la opción Dimensión	NA	la interfaz de dimensión visualizada	X		
2	Acceder al ícono para crear Dimensión	Se ingresan los datos Identificador, nombre y descripción de la Dimensión	Campos para crear una dimensión	X		
3	Hacer clic en el botón guardar	Datos de la dimensión en los campos	Mensaje de guardado exitoso y visualización de la dimensión creada	X		

Editar Dimensión

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación							
ID de caso de prueba: BU_003			Título de la prueba: Probar funcionalidad editar Dimensión				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz			Fecha de prueba de diseño:		DD	MM	AA
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez		
					Fecha de ejecución de la prueba:		
					DD	MM	AA
					26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Dimensión			
Descripción: Verifique que todos los registros de la interfaz de dimensión permitan editarse.							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias: Debe existir por lo menos una dimensión							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota	
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción Dimensión	NA	Interfaz de dimensión listando las dimensiones creadas	X			
2	Acceder al ícono para editar Dimensión	Toma una dimensión de la lista para editarla	Campos de la dimensión en modo de edición	X			
3	Hacer clic en el botón editar	Datos de la dimensión en los campos modificados	Mensaje: ¡Operación exitosa La dimensión fue actualizada satisfactoria mente!	X			
4	Clic botón ok		Interfaz de dimensión listando las dimensiones	X			

Buscar Dimensión

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación								
ID de caso de prueba: BU_004				Título de la prueba: Probar funcionalidad buscar Dimensión				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez			
			X					
Fecha de ejecución de la prueba:					DD	MM	AA	
					26	11	2021	
Nombre del Menú: Gestión				Nombre del Submenú: Dimensión				
Descripción: Verifique que todos los registros de la interfaz de dimensión permitan editarse.								
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión								
Dependencias: Debe existir por lo menos una dimensión								
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota		
				SI	NO			
1	Acceder al módulo de gestión e ir a la opción Dimensión	NA	Interfaz de dimensión listando las dimensiones creadas	X				
2	Acceder al cuadro de texto de buscar y colocar el nombre o parte y pulsar el icono buscar	Colocar el nombre o parte del nombre de la dimensión a buscar que exista	Sólo se muestra en pantalla la dimensión solicitada	X				

Eliminar Dimensión

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación								
ID de caso de prueba: BU_005				Título de la prueba: Probar funcionalidad eliminar Dimensión				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA
						17	11	2021
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez			
Fecha de ejecución de la prueba:					DD	MM	AA	
Nombre del Menú: Gestión				Nombre del Submenú: Dimensión				
Descripción: Verifique que todos los registros de la interfaz de dimensión permitan eliminarse								
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión								
Dependencias: Debe existir por lo menos una dimensión								
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota		
				SI	NO			
1	Acceder al módulo de gestión e ir a la opción Dimensión	NA	Interfaz de dimensión listando las dimensiones creadas	X				
2	Acceder al ícono para eliminar Dimensión	Toma una dimensión de la lista para editarla	Mensaje del sistema de confirmación	X				
3	Hacer clic en el botón sí elimínalo		Mensaje: ¡Operación	X				

			exitosa La dimensión fue eliminada satisfactoriam ente!			
4	Clic botón ok		Interfaz de dimensión listando las dimensiones	X		

Casos de Prueba de navegación panel de Gestión opción Factor Crear Factor

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación							
ID de caso de prueba: BU_006			Título de la prueba: Probar funcionalidad crear Factor				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz			Fecha de prueba de diseño:		17	11	21
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez		
				X	Fecha de ejecución de la prueba:		
					DD	MM	AA
					26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Factor			
Descripción: Verificar que se pueda crea un nuevo factor							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias:							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCID		Nota	
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción Factor	NA	la interfaz de factor visualizada	X			
2	Acceder al ícono para crear un Factor	Datos reales de un factor Identificador, nombre y descripción peso y Dimensión	Campos para crear un factor	X			
3	Hacer clic en el botón guardar	Datos del factor en los campos	Mensaje de guardado exitoso y visualización del factor creado	X			

Editar Factor

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación							
ID de caso de prueba: BU_007			Título de la prueba: Probar funcionalidad editar Factor				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz			Fecha de prueba de diseño:		DD	MM	AA
					17	11	2021
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez		
					DD	MM	AA

				Fecha de ejecución de la prueba:	26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Factor			
Descripción: Verifique que todos los registros de la interfaz de factor permitan editarse.							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias: Debe existir por lo menos un factor							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota	
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción Factor	NA	Interfaz de factor listando los factores creados	X			
2	Acceder al ícono para editar factor	Toma un factor de la lista para editarla	Campos de factor en modo de edición	X			
3	Hacer clic en el botón editar	Datos del factor en los campos modificados	Mensaje: ¡Operación exitosa El factor fue actualizado satisfactoria mente!	X			
4	Clic botón ok		Interfaz de factor listando factores	X			

Buscar Factor

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación							
ID de caso de prueba: BU_008				Título de la prueba: Probar funcionalidad buscar Factor			
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:			
				DD	MM	AA	
				17	11	2021	
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez		
			X		Fecha de ejecución de la prueba:		
					DD	MM	AA
					26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Factor			
Descripción: Verifique que todos los registros de la interfaz de factor permitan editarse.							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias: Debe existir por lo menos un factor							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota	
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción Factor	NA	Interfaz de factor listando los factores creados	X			
2	Acceder al cuadro de texto de buscar y colocar el nombre o	Colocar el nombre o parte del nombre del factor a buscar que exista	Sólo se muestra en pantalla el	X			

	parte y pulsar el icono buscar		factor solicitado.			
--	--------------------------------	--	--------------------	--	--	--

Eliminar Factor

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación								
ID de caso de prueba: BU_009				Título de la prueba: Probar funcionalidad eliminar Factor				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA
						17	11	2021
Prioridad de prueba	Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez		
						Fecha de ejecución de la prueba:		DD
								26
Nombre del Menú: Gestión				Nombre del Submenú: Factor				
Descripción: Verifique que todos los registros de la interfaz de factor permitan eliminarse								
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión								
Dependencias: Debe existir por lo menos un factor								
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota		
				SI	NO			
1	Acceder al módulo de gestión e ir a la opción Factor	NA	Interfaz de factor listando factores creados	X				
2	Acceder al ícono para eliminar Factor	Toma un factor de la lista para editarla	Mensaje del sistema de confirmación	X				
3	Hacer clic en el botón sí elimínalo		Mensaje: ¡Operación exitosa El factor fue eliminado satisfactoriam ente!	X				
4	Clic botón ok		Interfaz de factor listando factores	X				

Casos de Prueba de navegación panel de Gestión opción Característica Crear Característica

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación								
ID de caso de prueba: BU_010				Título de la prueba: Probar funcionalidad crear característica				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		17	11	21
Prioridad de prueba	Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez		
			X			Fecha de ejecución de la prueba:		DD
								26
Nombre del Menú: Gestión				Nombre del Submenú: característica				
Descripción: Verificar que se pueda crear una nueva característica								

Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión						
Dependencias:						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCID E		Nota
				SI	NO	
1	Acceder al módulo de gestión e ir a la opción característica	NA	la interfaz de característica visualizada	X		
2	Acceder al ícono para crear característica	Se ingresan los datos Identificador, nombre y descripción, factor, peso y número de indicadores	Campos para crear una característica	X		
3	Hacer clic en el botón guardar	Datos de la característica en los campos	Mensaje de guardado exitoso y visualización de la característica creada	X		

Editar Característica

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación						
ID de caso de prueba: BU_011				Título de la prueba: Probar funcionalidad editar Característica		
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		
				DD	MM	AA
				17	11	2021
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por:	
					Diana Rocío Ramírez Pérez	
				Fecha de ejecución de la prueba:		
				DD	MM	AA
				26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Característica		
Descripción: Verifique que todos los registros de la interfaz de característica permitan editarse.						
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión						
Dependencias: Debe existir por lo menos una característica						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota
				SI	NO	
1	Acceder al módulo de gestión e ir a la opción Característica	NA	Interfaz de característica listando las características creadas	X		
2	Acceder al ícono para editar Característica	Toma una característica de la lista para editarla	Campos de la característica en modo de edición	X		
3	Hacer clic en el botón editar	Datos de la característica en los campos modificados	Mensaje: ¡Operación exitosa La característica fue actualizada satisfactoriamente!	X		Pese a que aparece el mensaje de guardado exitoso no guarda el tipo de autoevaluación, mientras que en indicadores no aparece el número sino que se repite el identificador como se indicó en el cuadro anterior.

4	Clic botón ok		Interfaz de característica listando las dimensiones	X		
---	---------------	--	---	---	--	--

Buscar Característica

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación							
ID de caso de prueba: BU_012			Título de la prueba: Probar funcionalidad buscar Características				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz			Fecha de prueba de diseño:		DD	MM	AA
					17	11	2021
Prioridad de prueba	Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez	
		X				Fecha de ejecución de la prueba:	
				26	11		
Nombre del Menú: Gestión			Nombre del Submenú: Característica				
Descripción: Verifique que todos los registros de la interfaz de característica permitan editarse.							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias: Debe existir por lo menos una característica							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota	
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción Característica	NA	Interfaz de característica listando las características creadas	X			
2	Acceder al cuadro de texto de buscar y colocar el nombre o parte y pulsar el icono buscar	Colocar el nombre o parte del nombre de la característica a buscar que exista	Sólo se muestra en pantalla la característica solicitada	X			

Eliminar Característica

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación							
ID de caso de prueba: BU_013			Título de la prueba: Probar funcionalidad eliminar Característica				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz			Fecha de prueba de diseño:		DD	MM	AA
					17	11	2021
Prioridad de prueba	Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez	
						Fecha de ejecución de la prueba:	
				26	11		
Nombre del Menú: Gestión			Nombre del Submenú: Característica				
Descripción: Verifique que todos los registros de la interfaz de característica permitan eliminarse							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias: Debe existir por lo menos una característica							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota	
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción Característica	NA	Interfaz de característica listando las características creadas	X			

2	Acceder al ícono para eliminar Característica	Toma una característica de la lista para editarla	Mensaje del sistema de confirmación	X		
3	Hacer clic en el botón sí elimínala		Mensaje: ¡Operación exitosa La característica fue eliminada satisfactoriam ente!	X		
4	Clic botón ok		Interfaz de característica listando las características	X		

Casos de Prueba de navegación panel de Gestión opción Indicador Crear Indicador

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación						
ID de caso de prueba: BU_014				Título de la prueba: Probar funcionalidad crear Indicador		
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		
				17	11	21
Prioridad de prueba				Prueba ejecutada por: Diana Rocío Ramírez Pérez		
Baja		Media	Alta	Fecha de ejecución de la prueba:		
			X	DD	MM	AA
				26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Indicador		
Descripción: Verificar que se pueda crea un nuevo indicador						
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión						
Dependencias:						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCID		Nota
				SI	NO	
1	Acceder al módulo de gestión e ir a la opción Indicador	NA	la interfaz de indicador visualizada	X		
2	Acceder al ícono para crear un Indicador	Datos reales de un indicador nombre, descripción # de evidencias y característica	Campos para crear un indicador	X		

3	Hacer clic en el botón guardar	Datos del indicador en los campos	Mensaje de guardado exitoso y visualización del indicador creado	X		
---	--------------------------------	-----------------------------------	--	---	--	--

Editar Indicador

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación								
ID de caso de prueba: BU_015				Título de la prueba: Probar funcionalidad editar Indicador				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA
						17	11	2021
Prioridad de prueba	Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez		
						Fecha de ejecución de la prueba:		DD
								26
Nombre del Menú: Gestión				Nombre del Submenú: Indicador				
Descripción: Verifique que todos los registros de la interfaz de indicador permitan editarse.								
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión								
Dependencias: Debe existir por lo menos un indicador								
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota		
				SI	NO			
1	Acceder al módulo de gestión e ir a la opción Indicador	NA	Interfaz de indicador listando los indicadores creados	X				
2	Acceder al ícono para editar indicador	Toma un indicador de la lista para editarla	Campos de indicador en modo de edición	X				
3	Hacer clic en el botón editar	Datos del indicador en los campos modificados	Mensaje: ¡Operación exitosa El indicador fue actualizado satisfactoria mente!	X				
4	Clic botón ok		Interfaz de indicador listando indicadores	X				

Buscar Indicador

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación								
ID de caso de prueba: BU_016				Título de la prueba: Probar funcionalidad buscar Indicador				
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA
						17	11	2021
Prioridad de prueba	Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez		

		X		Fecha de ejecución de la prueba:	DD	MM	AA
Nombre del Menú: Gestión				Nombre del Submenú: Indicador			
Descripción: Verifique que todos los registros de la interfaz de indicador permitan editarse.							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias: Debe existir por lo menos un indicador							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota	
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción Indicador	NA	Interfaz de indicador listando los factores creados	X			
2	Acceder al cuadro de texto de buscar y colocar el nombre o parte y pulsar el icono buscar	Colocar el nombre o parte del nombre del indicador a buscar que exista	Sólo se muestra en pantalla el indicador solicitado.	X			

Eliminar Indicador

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación									
ID de caso de prueba: BU_017				Título de la prueba: Probar funcionalidad eliminar Indicador					
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA	
						17	11	2021	
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez		
					Fecha de ejecución de la prueba:		DD	MM	AA
							26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Indicador					
Descripción: Verifique que todos los registros de la interfaz de indicador permitan eliminarse									
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión									
Dependencias: Debe existir por lo menos un indicador									
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota			
				SI	NO				
1	Acceder al módulo de gestión e ir a la opción Indicador	NA	Interfaz de indicador listando indicadores creados	X					
2	Acceder al ícono para eliminar Indicador	Toma un indicador de la lista para editarla	Mensaje del sistema de confirmación	X					
3	Hacer clic en el botón sí elimínalo		Mensaje: ¡Operación exitosa El indicador fue eliminado satisfactoriam ente!	X					
4	Clic botón ok		Interfaz de indicador listando indicadores	X					

Casos de Prueba de navegación panel de Gestión opción Evidencia de Cumplimiento
 Crear Evidencia de Cumplimiento

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación							
ID de caso de prueba: BU_018				Título de la prueba: Probar funcionalidad crear evidencia de cumplimiento			
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:	17	11	21
Prioridad de prueba	Baja	Media	Alta	Prueba ejecutada por:	Diana Rocío Ramírez Pérez		
			X	Fecha de ejecución de la prueba:	DD	MM	AA
					26	11	2021
Nombre del Menú: Gestión				Nombre del Submenú: Evidencia de Cumplimiento			
Descripción: Verificar que se pueda crea una nueva evidencia							
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión							
Dependencias:							
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCID		Nota	
				E			
				SI	NO		
1	Acceder al módulo de gestión e ir a la opción evidencia de cumplimiento	NA	la interfaz de evidencia de cumplimiento visualizada	X			
2	Acceder al ícono para crear evidencia de cumplimiento	Se ingresan los datos nombre, evidencia y el indicador al que pertenece	Campos para crear una evidencia de cumplimiento	X			
3	Hacer clic en el botón guardar	Datos de la evidencia de cumplimiento en los campos	Mensaje de guardado exitoso y visualización de la evidencia de cumplimiento creada	X			

Editar Evidencia de Cumplimiento

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación

ID de caso de prueba: BU_019				Título de la prueba: Probar funcionalidad editar Evidencia de Cumplimiento					
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA	
						17	11	2021	
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez		
					Fecha de ejecución de la prueba:		DD	MM	AA
							26	11	2021
Nombre del Menú: Gestión					Nombre del Submenú: Evidencia de Cumplimiento				
Descripción: Verifique que todos los registros de la interfaz de evidencia de cumplimiento permitan editarse.									
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión									
Dependencias: Debe existir por lo menos una evidencia de cumplimiento									
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota			
				SI	NO				
1	Acceder al módulo de gestión e ir a la opción evidencia de cumplimiento	NA	Interfaz de evidencia de cumplimiento listando las características creadas	X					
2	Acceder al ícono para editar evidencia de cumplimiento	Toma una evidencia de cumplimiento de la lista para editarla	Campos de la evidencia de cumplimiento en modo de edición	X					
3	Hacer clic en el botón editar	Datos de la evidencia de cumplimiento en los campos modificados	Mensaje: ¡Operación exitosa La evidencia de cumplimiento fue actualizada satisfactoriamente!	X					
4	Clic botón ok		Interfaz de evidencia de cumplimiento listando las evidencias de cumplimientos	X					

Buscar Evidencia de Cumplimiento

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación									
ID de caso de prueba: BU_020				Título de la prueba: Probar funcionalidad buscar Evidencia de Cumplimiento					
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD	MM	AA	
						17	11	2021	
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por:		Diana Rocío Ramírez Pérez		
			X		Fecha de ejecución de la prueba:		DD	MM	AA
							26	11	2021
Nombre del Menú: Gestión					Nombre del Submenú: Evidencia de Cumplimiento				
Descripción: Verifique que todos los registros de la interfaz de evidencia de cumplimiento permitan editarse.									
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión									
Dependencias: Debe existir por lo menos una evidencia de cumplimiento									
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota			
				SI	NO				
1	Acceder al módulo de gestión e ir a la	NA	Interfaz de evidencia de cumplimiento	X					

	<i>opción evidencia de cumplimiento</i>		listando las evidencias creadas			
2	<i>Acceder al cuadro de texto de buscar y colocar el nombre o parte y pulsar el icono buscar</i>	<i>Colocar el nombre o parte del nombre de la evidencia de cumplimiento a buscar que exista</i>	Sólo se muestra en pantalla la evidencia de cumplimiento solicitada	X		

Eliminar Evidencia de Cumplimiento

Nombre del proyecto: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación						
ID de caso de prueba: BU_021				Título de la prueba: Probar funcionalidad eliminar Evidencia de Cumplimiento		
Prueba diseñada por: Equipo de desarrollo Arellys de Jesús Correa Rodríguez Nathalia Andrea Cuervo Díaz				Fecha de prueba de diseño:		DD 17
						MM 11
						AA 2021
Prioridad de prueba		Baja	Media	Alta	Prueba ejecutada por: Diana Rocío Ramírez Pérez	
					Fecha de ejecución de la prueba:	
					DD 26	MM 11
					AA 2021	
Nombre del Menú: Gestión				Nombre del Submenú: Evidencia de Cumplimiento		
Descripción: Verifique que todos los registros de la interfaz de evidencia de cumplimiento permitan eliminarse						
Condiciones previas: El usuario debe estar registrado y con acceso al módulo de gestión						
Dependencias: Debe existir por lo menos una evidencia de cumplimiento						
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	COINCIDE		Nota
				SI	NO	
1	<i>Acceder al módulo de gestión e ir a la opción evidencia de cumplimiento</i>	NA	Interfaz de evidencia de cumplimiento listando las evidencias de cumplimiento creadas	X		
2	<i>Acceder al ícono para eliminar evidencia de cumplimiento</i>	<i>Toma una evidencia de cumplimiento de la lista para editarla</i>	Mensaje del sistema de confirmación	X		
3	<i>Hacer clic en el botón sí elimínala</i>		Mensaje: ¡Operación exitosa La evidencia de cumplimiento fue eliminada satisfactoriam ente!	X		
4	<i>Clic botón ok</i>		Interfaz de evidencia de cumplimiento listando las evidencias de cumplimiento creadas	X		

Anexo 8 Cesión de derechos patrimoniales

CONTRATO DE CESIÓN DE DERECHOS PATRIMONIALES DE AUTOR

Suscrito entre, por una parte, **LEONARDO EVEMELETH SANCHEZ TORRES**, identificado con cédula de ciudadanía No. 79.909.423 expedida en Bogotá, debidamente facultado para suscribir convenios conforme a lo manifestado en la Resolución No. 006 del 3 de enero de 2020, quien de conformidad con la Resolución 015316 del 2 de diciembre de 2019 se desempeña como Vicerrector de Relaciones Internacionales de la UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA -UNAD, Ente Universitario Autónomo de orden Nacional, creado por Ley 52 de 1981 transformado mediante Decreto Ley 2770 del 16 de Agosto de 2006, con régimen especial en los términos de la Ley 30 de 1992, personería jurídica, autonomía académica, administrativa y financiera; patrimonio independiente y capacidad para gobernarse, vinculado al Ministerio de Educación Nacional, identificada con NIT 860.512.780-4 quien adelante se denominará **CESIONARIO** y por otra parte **Arellys de Jesús Correa Rodríguez** mayor de edad, identificada con cédula de ciudadanía número 45.750.408 expedida en la ciudad de Cartagena, **Claudio Camilo González Clavijo** mayor de edad, identificado con cédula de ciudadanía número 79.287.058 expedida en la ciudad de Bogotá y **Nathalia Andrea Cuervo Díaz** mayor de edad, identificada con cédula de ciudadanía número 1.049.631.725 expedida en la ciudad de Tunja y quienes en lo sucesivo y para los efectos de este contrato se designarán los **CEDENTES**, (conjuntamente las “Partes” y cada una individualmente como la “Parte”) han acordado celebrar **CONTRATO DE CESIÓN DE DERECHOS PATRIMONIALES DE AUTOR** (en adelante el “CONTRATO”) que se regirá por las disposiciones legales aplicables y en especial por las siguientes cláusulas:

CONSIDERANDO

- I) Que los **CEDENTES** realizan la ejecución del proyecto “ECBTIPIE312019 Implementación de la metodología BPM y herramientas software para ejercicios de autoevaluación institucional y de programas académicas de la calidad en la Universidad Nacional Abierta y a Distancia - UNAD”

- II) Que, como resultado del proyecto y contrato, los **CEDENTES** son autores de la obra: Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación, el cual se lista en este contrato y cuya descripción y características se adjuntan a este instrumento. (en adelante **LA OBRA**).
- III) Que el **CESIONARIO** desea ser el titular único de todos los derechos patrimoniales de autor, en todo el mundo y de manera permanente, entre otros el derecho de reproducción y explotación en todas sus modalidades, el derecho de transformación o adaptación, comunicación pública, distribución y, en general, cualquier tipo de explotación que de **LA OBRA** se pueda realizar por cualquier medio conocido o por conocer.
- IV) Que los **CEDENTES** ceden la totalidad de sus derechos patrimoniales sobre **LA OBRA** según los términos del presente contrato.

En vista de los considerandos anteriores, las partes del presente **CONTRATO** se comprometen con las siguientes cláusulas:

CLÁUSULA PRIMERA. OBJETO. - Por medio de este contrato los **CEDENTES** transfieren al **CESIONARIO** la totalidad de los derechos patrimoniales de autor sobre **LA OBRA**, en todo el mundo y de manera permanente, que se lista a continuación: “Software para gestión del proceso de recolección y tratamiento de apreciaciones de la comunidad académica en procesos de autoevaluación”, el cual se encarga de gestionar el procedimiento de apreciación que consiste en la estimación de los programas académicos y la institución por parte de los grupos de interés, en ese sentido permite, diseñar los instrumentos para enviarlos a los estamentos y así obtener una calificación por cada factor y característica de acuerdo al modelo de autoevaluación Unadista, al final se obtiene una calificación final para el programa la cual está asociada a una escala de juicio cualitativo.

CLÁUSULA SEGUNDA. DURACIÓN. - La presente Cesión aplica tanto en Colombia como en cualquier otro país y se realiza a perpetuidad y/o en su defecto, por el tiempo máximo que establezca la Ley nacional e internacional.

CLAUSULA TERCERA. REMUNERACION. - La presente cesión se entiende concedida como resultado del cumplimiento de las obligaciones contractuales vigentes entre las partes establecidas en la resolución 3908 del día 29 del mes enero del año 2021, resolución 258 del día 15 del mes enero del año 2021, suscritos entre **Arellys de Jesús Correa Rodríguez, Claudio Camilo González Clavijo, Nathalia Andrea Cuervo Díaz y LA UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA- UNAD**

CLÁUSULA CUARTA. TERRITORIO. - Los derechos se ceden para todo el mundo y de manera permanente e indefinida, incluyendo todo el territorio colombiano.

CLÁUSULA QUINTA. INTEGRIDAD CONTRACTUAL. - Si en cualquier momento, una de las cláusulas de este contrato es declarada ilegal, inoponible o nula e inválida, la legalidad, validez y efectividad de las demás disposiciones de este contrato no serán afectadas o deterioradas por esta nulidad.

CLÁUSULA SEXTA. Los **CEDENTES** se comprometen a no impedir o restringir la producción, distribución, promoción o explotación de **LA OBRA** en ninguna forma, mientras el uso de los

derechos aquí concedidos por los **CEDENTES**, se haga dentro de las normas legales y las que se pactan en el presente contrato.

CLÁUSULA SÉPTIMA. Los **CEDENTES** garantizan que detenta todos los derechos que le confieren las leyes sobre **LA OBRA** en especial, todos los derechos de autor asociados a la misma, y, en consecuencia, garantiza que puede hacer la presente cesión sin ningún tipo de limitación, por no pesar sobre **LA OBRA** ningún tipo de gravamen o limitación a los derechos sobre la disposición que los **CEDENTES** puede hacer de la misma. En caso de que se presente algún reclamo fundamentado por un tercero, ante autoridad competente, los **CEDENTES** responderán al **CESIONARIO** por los perjuicios causados.

CLÁUSULA OCTAVA. - SOLUCIÓN DE CONTROVERSIAS. Las diferencias que surjan entre las partes por la aplicación e interpretación de este contrato serán resueltas mediante la justicia ordinaria.

CLÁUSULA NOVENA. LEY APLICABLE. Será aplicable la ley colombiana.

Para constancia de lo aquí estipulado, las Partes firman el contrato el día 18 de noviembre de 2021 en la ciudad de Bogotá.

EL CESIONARIO

Leonardo Evemeleth Sánchez Torres
C.C. 79.909.423 de Bogotá
Vicerrector de Relaciones Internacionales

LOS CEDENTES

Nombre: Arellys de Jesús Correa Rodríguez
C.C.: 45.750.408 de Cartagena
Correo: arellys.correa@unad.edu.co
Dirección: Calle 29D #21C146 Apt. 2ª
Edf. José Félix, Cartagena
Celular: 3012507377

Nombre: Claudio Camilo González Clavijo
C.C.: 79.287.058 de Bogotá
Correo: claudio.gonzalez@unad.edu.co
Dirección: Calle 14 Sur #14-23, Bogotá
Celular: 3107506713

Nombre: Nathalia Andrea Cuervo Díaz
1.049.631.225 de Tunja
Correo: nathalia.cuervo@unad.edu.co
Dirección: Calle 42 # 1b-a Terrazas de
santa ines, Tunja

C.C.:

Celular: 3043448585