



Análisis y Diseño de los módulos de Ejercicios y Evaluación que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.

Eric Andrés Pinto Muñoz

Universidad Magdalena

Facultad, Ingeniería
Programa Ingeniería de Sistemas
Santa Marta, Colombia
2020



Análisis y Diseño de los módulos de Ejercicios y evaluación que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.

Eric Andrés Pinto Muñoz

Trabajo presentado como requisito parcial para optar al título de:
Ingeniero de Sistemas

Director (a):
Ingeniera de Sistemas, MsC. Emperatriz Zapata Zapata

Línea de Investigación:
Análisis y diseño de software
Grupo de Investigación:
Grupo de investigación y Desarrollo en Sistemas y Computación – GIDSYC

Facultad, Ingeniería
Programa Ingeniería de Sistemas
Santa Marta, Colombia
2020

Nota de aceptación:

Aprobado por el Consejo de Programa en cumplimiento de los requisitos exigidos por la Universidad del Magdalena para optar al título de Ingeniero de Sistemas

Jurado

Jurado

Santa Marta, ____ de ____ del ____

Resumen

El proceso de paz que vive el país hace necesario que los ciudadanos aprendan a vivir en sociedad respetando entre otros los pensamientos, orientaciones y decisiones de sus conciudadanos. Esta construcción de convivencia se viene dando desde la educación básica por lo cual por el Ministerio de Educación Nacional (MEN), además de definir las temáticas a desarrollar en los diferentes grados, también realiza evaluación de los aprendizajes esperados, a través de pruebas genéricas a nivel de educación básica, media y profesional.

La presente pasantía de investigación hace parte del proyecto que tiene como objetivo diseñar e implementar un software de evaluación, que permita apoyar el desarrollo de las habilidades, para mejorar los resultados en competencias ciudadanas de los estudiantes de la Universidad del Magdalena, en las pruebas nacionales Saber Pro para el año 2021, en específico el desarrollo del análisis y diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.

ABSTRACT

Keywords: competencias ciudadanas, software educativo, Saber Pro.

CONTENIDO

	Pág.
1. PASANTÍA DE INVESTIGACIÓN.....	11
1.1 PLANTEAMIENTO DE LA SITUACIÓN.....	11
1.2 OBJETIVO.....	11
1.3 ALCANCE	11
2. INTRODUCCIÓN	1
3. MARCO TEÓRICO.....	2
3.1. ARQUITECTURA DEL SOFTWARE.....	2
3.1.1. <i>Modelo Vista Controlador (MVC).</i>	2
3.1.2. <i>Modelo en Capas</i>	3
3.2. METODOLOGÍA DE DESARROLLO	4
3.2.1. <i>Metodología secuencial – lineal o en cascada</i>	4
3.2.2. <i>Rational Unified Process (RUP)</i>	4
4. ESTADO DEL ARTE	5
4.1. E SERVICIOS EDUCATIVOS (ESE).	5
4.2. EL GRUPOGEARD.....	6
4.3. AMIR	7
5. ANÁLISIS	8
5.1. DEFINICIÓN DE LA METODOLOGÍA DE SOLUCIÓN	8
5.1.1. <i>Metodología secuencial lineal o en cascada</i>	8
5.1.2. <i>Metodología Proceso Racional Unificado (RUP)</i>	9
5.1.3. <i>Elección de la metodología</i>	10
5.2. ANÁLISIS DE REQUERIMIENTOS	11
5.2.1. <i>Requerimientos del módulo de administración:</i>	12
5.2.2. <i>Requerimientos del módulo de alimentación del banco de pregunta</i>	13
5.2.3. <i>Requerimientos del módulo de ejercitación:</i>	13
5.2.4. <i>Requerimientos del módulo de evaluación:</i>	13
5.3. ANÁLISIS DEL SISTEMA.....	13
5.3.1. <i>Casos de uso generales</i>	13
5.3.2. <i>Casos de usos específicos y requerimiento funcionales</i>	19
5.3.3. <i>Requerimientos no funcionales</i>	25
6. DISEÑO.....	26
6.1. ARQUITECTURA GENERAL.....	26
6.1.1. <i>Patrón de diseño MVC</i>	26
6.1.2. <i>Patrón de en N-Capas</i>	27
6.1.3. <i>Patrón de diseño de la solución</i>	28
6.2. DISEÑO DE LA ESTRUCTURA DE DATOS	30
6.2.1. <i>Diagrama de base de datos</i>	30
6.2.2. <i>Entorno operativo</i>	44

6.2.2.1. Backend.....	45
6.2.2.2. Frontend.....	47
6.2.2.3. DataBase.....	48
6.2.3. Interfaz de usuario.....	49
6.2.3.1. Administración de usuarios.....	49
6.2.3.2. Instituciones o entidades educativas.....	50
6.2.3.3. Docentes y revisores.....	51
6.2.3.4. Estudiantes.....	55
7. CONCLUSIONES.....	57
7.1. CONCLUSIONES GENERALES.....	57
7.2. CONCLUSIONES ESPECIFICAS.....	57

Lista de figuras

	Pág.
Figura 1. Esquema de Fases del Desarrollo en Cascada	9
Figura 2. Esquema de Fases del Desarrollo en RUP	10
Figura 3. Caso de uso general del administrador.....	15
Figura 4. Caso de Uso general de las instituciones.	15
Figura 5. Caso de uso general de los docentes.	16
Figura 6. Caso de uso general.....	16
Figura 7. Caso de uso general de los estudiantes.	17
Figura 8. Modelo de gestión de procesos de negocio.	18
Figura 9. Interacciones del patrón MVC.....	27
Figura 10. Interacciones del patrón N-Capas.....	28
Figura 11. Flujo de arquitectura	29
Figura 12. Diagrama de base de datos del sistema	30
Figura 13. Tablas ESTUDIANTES-PREGUNTAS-RESPUESTAS	31
Figura 14. Lenguajes de programación más usados en 2019.....	45
Figura 15. Web Frameworks más usados en 2019.....	47
Figura 16. Motores de base de datos más usados en 2019.....	48
Figura 17. Pantalla de administración general	50
Figura 18. Pantalla de administración de las instituciones educativas.	51
Figura 19. Pantalla de registro y administración de preguntas.	52
Figura 20. Pantalla de corrección de preguntas.....	53
Figura 21. Pantalla de registro de preguntas.	54
Figura 22. Pantalla de inicio de los estudiantes.	56

Lista de tablas

	Pág.
Tabla 1. Actores del sistema.....	14
Tabla 2. Casos de usos	19
Tabla 3. Afirmaciones	32
Tabla 4. Áreas temáticas	32
Tabla 5. Componentes.....	33
Tabla 6. Contenidos.....	33
Tabla 7. Cuestionarios.....	34
Tabla 8. Cuestionarios_has_Respuestas.....	35
Tabla 9. Docentes.....	35
Tabla 10. Estudiantes	36
Tabla 11. Estudiantes Autorizados	37
Tabla 12. Estudiantes_has_Contenidos.....	37
Tabla 13. Estudiantes_has_Respuestas.....	38
Tabla 14. Evidencias	38
Tabla 15. Facultades	39
Tabla 16. Instituciones_Educativas.....	39
Tabla 17. Niveles	40
Tabla 18. Observaciones	41
Tabla 19. Preguntas	41
Tabla 20. Programas	42
Tabla 21. Respuestas.....	43
Tabla 22.Usuarios.....	43

Lista de Anexos

	Pág.
Anexo 1. Inicio de sesión de usuarios.....	58
Anexo 2. Cierre de sesión de usuarios	59
Anexo 3. Registrar estudiantes	61
Anexo 4. Registrar instituciones o entidades educativas.....	63
Anexo 5.Registrar docentes.....	64
Anexo 6. Configuración de perfiles y gestión de usuarios	66
Anexo 7. Administrar docentes	69
Anexo 8. Registrar componentes (competencias y habilidades)	71
Anexo 9. Administrar componente	73
Anexo 10. Registrar afirmaciones	75
Anexo 11. Administrar afirmaciones.....	77
Anexo 12. Registrar evidencias	79
Anexo 13. Administrar evidencias	81
Anexo 14. Registrar área temática.....	83
Anexo 15. Administrar áreas temáticas.....	85
Anexo 16. Registrar nivel de complejidad	88
Anexo 17. Administrar niveles de complejidad.....	89
Anexo 18. Crear contenido de entrenamiento.....	91
Anexo 19. Crear contenido para el banco de preguntas	93
Anexo 20. Asignar responsabilidades	95
Anexo 21. Sugerir corrección de preguntas	96
Anexo 22. Corregir y publicar preguntas.....	98
Anexo 23. Responder preguntas de entrenamiento	100
Anexo 24. Responder cuestionarios	102
Anexo 25. Retroalimentar	103
Anexo 26. Base de datos.....	105
Anexo 27. Diagrama Inicio de sesión.....	106
Anexo 28. Diagrama Cierre de sesión de usuarios	107
Anexo 29. Diagrama Registro estudiante.....	108
Anexo 30. Diagrama Registro institución	109
Anexo 31. Diagrama Registro docente	110
Anexo 32. Diagrama Configuración de perfiles y gestión de usuarios.....	111
Anexo 33. Diagrama Administrar docente.....	112
Anexo 34. Diagrama Registrar componentes (competencias y habilidades)	113
Anexo 35. Diagrama Administrar componente.....	114
Anexo 36. Diagrama Registrar Afirmaciones	115
Anexo 37. Diagrama Administrar afirmaciones	116
Anexo 38. Diagrama Registrar evidencia.....	117
Anexo 39. Diagrama Administrar evidencias.....	118
Anexo 40. Diagrama Registrar área temática	119
Anexo 41. Diagrama Administrar área temática.....	120

Anexo 42. Diagrama Registrar nivel de complejidad.....	121
Anexo 43. Diagrama Administrar nivel de complejidad	122
Anexo 44. Diagrama Crear contenido de entrenamiento.....	123
Anexo 45. Diagrama Crear contenido para el banco de preguntas	124
Anexo 46. Diagrama Asignar responsabilidades.....	125
Anexo 47. Diagrama Sugerir corrección de preguntas.....	126
Anexo 48. Diagrama Corregir y publicar preguntas.....	127
Anexo 49. Diagrama Responder preguntas de entrenamiento.....	128
Anexo 50. Diagrama Responder cuestionarios	129
Anexo 51. Diagrama Retroalimentar	130
Anexo 52. Aspectos funcionales	131
Anexo 53. Aspectos Técnicos.....	132
Anexo 54. Aspectos Pedagógicos	133

1. Pasantía de Investigación

1.1 Planteamiento de la situación

Las actividades a realizar se encuentran dentro del marco del proyecto de investigación titulado: Análisis de resultados, definición y aplicación de estrategias para contribuir en el mejoramiento del desempeño de los estudiantes de la Universidad del Magdalena, en el componente de Competencias Ciudadanas en las pruebas Nacionales Saber Pro para el año 2021, en especial orientado a la elaboración e implementación de software educativo con el objetivo de mejorar los niveles de desempeño, en el componente de Competencias Comunicativas, de los estudiantes que presentan pruebas Saber Pro.

1.2 Objetivo

El objetivo principal de las actividades que se realizarán en el marco de la pasantía de investigación será el desarrollo de la fase de Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.

1.3 Alcance

El alcance de las actividades que lleven a la definición del modelo que se desarrollará debe cumplir los estándares definidos en la institución relacionados con desarrollo de software, incluirá el trabajo con dos de los investigadores en la definición de las estructuras de datos y en el desarrollo del modelo para las fases de análisis y diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software. El estudiante debe investigar y validar con su par, las herramientas que se ajusten más a la propuesta de software, de acuerdo con los requerimientos del investigador temático, las cuales deberán ser aprobadas por los investigadores Ingenieros de Sistemas que hace parte del proyecto. Diseño de

interfaces a través de maquetas de muestra de los módulos asignados de acuerdo con la interfaz elegida. Definición de la estructura de datos para los módulos asignados. Diseñar los datos de prueba para la realización posterior de la prueba de campo. Entrega de documentación.

2.Introducción

En los últimos años se ha determinado la importancia de formación de los profesionales en competencias ciudadanas además de otras competencias que fortalecerán su desempeño profesional y su vida en comunidad. La evaluación realizada por el ICFES a los estudiantes ha mostrado bajos niveles de desempeño en esta competencia. Con el objetivo de dar solución esta problemática se planteó este proyecto que tiene como finalidad diseñar e implementar un software educativo que permita a los estudiantes realizar entrenamientos y simulacros a través de una plataforma colaborativa en la que usuarios-profesores puedan compartir contenido educativo y preguntas relacionados con los temas que hacen parte de esta competencia.

Se determinó como metodología a utilizar para el desarrollo del proyecto, el modelo secuencial-lineal el cual se comprende las etapas de análisis, diseño, prueba y mantenimiento; para nuestro caso, se llevarán a cabo solo las fases de análisis y diseño.

En este documento, el análisis de la propuesta a través del levantamiento de requerimientos, revisión de plataformas existentes para contextualizar el estado del arte, revisión documental y la entrevista al experto. Se Muestra el marco teórico que orienta al diseño de la solución propuesta para el posterior desarrollo de la aplicación, por último, se define el diseño de la solución de la arquitectura, la definición de bases de datos y diccionario de datos, construcción del modelo entidad-relación y la maqueta tipo wireframe que permitirá el desarrollo del aplicativo.

3.Marco teórico

En esta sección se hace una ampliación del marco teórico para el desarrollo y comprensión de las metodologías y diferentes temáticas para darle vida al proyecto.

3.1. Arquitectura del software.

3.1.1. Modelo Vista Controlador (MVC).

Citado de los documentos *“El patrón de diseño Modelo – Vista – Controlador (MVC) y su implementación en Java Swing”* por Ernesto Bascón Pantoja y *“Patrón Modelo – Vista – Controlador”* por Yenisleidy Fernández Romero, Yanette Diaz González [1] *Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas”.*

- **.El modelo.**

“El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, así por ejemplo un sistema de administración de datos climatológicos tendrá un modelo que representar a la temperatura, la humedad ambiental, el estado del tiempo esperado, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación”[2].

- **Las vistas**

“Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo; así, por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital”[2].

- **El controlador**

“El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de

modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador”[2].

3.1.2. Modelo en Capas

Citado del documento “Programación en capas” de Ricardo J. Vargas Del Valle y Juan P. Maltés Granados “ [3]*La programación por capas es una técnica de ingeniería de software propia de la programación por objetos, éstos se organizan principalmente en 3 capas: la capa de presentación o frontera, la capa de lógica de negocio o control, y la capa de datos. Siguiendo el modelo, el desarrollador se asegura avanzar en la programación del proyecto de una forma ordenada, lo cual beneficia en cuanto a reducción de costos por tiempo, debido a que se podrá avanzar de manera más segura en el desarrollo, al ser dividida la aplicación general en varios módulos y capas que pueden ser tratados de manera independiente y hasta en forma paralela”.*

- **Capa de presentación o Frontera.**

“La presentación del programa ante el usuario debe manejar interfaces que cumplan con el objetivo principal de este componente, el cual es facilitar al usuario la interacción con la aplicación. Para esto se utilizan patrones predefinidos para cada tipo de aplicación y para cada necesidad del usuario. La interfaz debe ser amigable y fácil de utilizar, ya que el usuario final es el que se va a encargar de utilizar el sistema y de dar retroalimentación al equipo de desarrollo en caso de que haya algo que mejorar”.

- **Capa Lógica de negocio o Control.**

“Es llamada capa de reglas de negocio porque en esta se definen todas las reglas que se deben cumplir para una correcta ejecución del programa. Es aquí donde se encuentra toda la lógica del programa, así como las estructuras de datos y objetos encargados para la manipulación de los datos existentes, así como el procesamiento de la información ingresada o solicitada por el usuario en la capa de presentación”.

- **Capa de Acceso a Datos**

“Es la encargada de realizar transacciones con bases de datos y con otros sistemas para obtener o ingresar información al sistema. El manejo de los datos debe realizarse de forma tal que haya consistencia en los mismos, de tal forma los datos que se ingresan, así como los que se extraen de las bases de datos, deben ser consistentes y precisos”.

3.2. Metodología de desarrollo

3.2.1. Metodología secuencial – lineal o en cascada

Citado del documento “Desarrollo en cascada (Waterfall) vs Desarrollo Agile-SCRUM” Por Jesús Demetrio Velázquez Camacho “ [4]*El esquema de desarrollo en cascada se caracteriza por proponer actividades secuenciales, claramente agrupadas dentro de fases o ciclos del desarrollo del proyecto, propone hacer un análisis intensivo de requerimientos y se vuelve complicado volver a etapas previas del proyecto cuando se encuentran diferencias significativas en el alcance definido en etapas tempranas del mismo. El levantamiento de requerimientos es muy riguroso y los Analistas definen a priori todos los requerimientos funcionales y no funcionales relacionados con el proyecto. Normalmente, una fase no puede iniciar sin que la fase anterior haya sido revisada y aceptada por el cliente o usuario final, sin que esto signifique el sistema cumplirá con sus necesidades.*

Como la mayoría de las metodologías este se compone de análisis de requisitos, diseño, implementación, verificación y mantenimiento”.

3.2.2. Rational Unified Process (RUP)

Citado del documento “Cuatro enfoques metodológicos para el desarrollo de software RUP – MSF – XP – SCRUM” por Oiver Andrés Pérez A. [5]*“RUP es una metodología que tiene como objetivo ordenar y estructurar el desarrollo de software, en la cual se tienen un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema Software (Amo, Martínez y Segovia, 2005). Inicialmente fue llamada UP (Unified Process) y luego cambió su nombre a RUP por el respaldo de Rational Software de IBM. Esta metodología fue lanzada en 1998 teniendo como sus creadores a Ivar Jacobson, Grady Booch y James Rumbaugh. El RUP nació del UML (Unified Modeling Language) y del UP (Sommerville, 2005).*

El RUP es un proceso basado en los modelos en Cascada y por Componentes, el cual presenta las siguientes características: Es dirigido por los casos de uso, es centrado en la arquitectura, iterativo e incremental (Booch, Rumbaugh y Jacobson, 2000), lo cual es fundamental para el proceso de desarrollo de software. A continuación, se explican las tres características de RUP:”

- **Casos de uso.**

“Describe un servicio que el usuario requiere del sistema, incluye la secuencia completa de interacciones entre el usuario y el sistema”.

- **Centrado en la arquitectura.**

“Comprende las diferentes vistas del sistema en desarrollo, que corresponden a los modelos del sistema: Modelos de casos de uso, de análisis, de diseño, de despliegue e implementación. La arquitectura del software es importante para comprender el sistema como un todo y a la vez en sus distintas partes (Abrahamsson, Salo, Ronkainen y Warsta, 2002), sirve para organizar el desarrollo, fomentar la reutilización de componentes y hacer evolucionar el sistema, es decir, agregarle más funcionalidad (Pressman y Murrieta, 2006)”.

- **Iterativo e incremental.**

“Significa que la aplicación se divide en pequeños proyectos, los cuales incorporan una parte de las especificaciones, y el desarrollo de esta es una iteración que va incrementando la funcionalidad del sistema de manera progresiva (Silva, Barrera, Arroyave y Pineda, 2007)”.

4.Estado del arte

4.1. E Servicios Educativos (ESE) [6].

Posee un sistema completo de entrenamiento en todas a áreas que evalúa el ICFES, incluyendo Saber 3°, 5°, 7°, 9°, Saber 11, Saber Pro y Saber TyT. Además, también cuenta con capacitaciones para docentes en diseño de evaluaciones de alto impacto.

Las diferencias encontradas con el presente trabajo son:

- Respecto a lo funcional:
 - Al contar con una interfaz de usuario minimalista mal enfocada pierde la atención del usuario al momento de revisar el contenido.
 - Esta no cuenta con sistema de seguimiento del progreso del usuario, está enfocado en la solución de los contenidos asignados hasta finalizarlos.

- Posee lecturas y videos complementarios tediosos para el usuario, las lecturas son demasiado extensas y los videos poco llamativos.
- Respecto a lo técnico:
 - El sistema presenta un diseño claro, pero no intuitivo para la navegación entre las diferentes áreas de esta.
 - ESE no cuenta con una cantidad de contenidos multimedia relevantes para el usuario, está enfocado en la lectura extensa y la solución de preguntas limitadas.
 - Al estar enfocado como un modelo de negocios privado, este necesita una rentabilidad positiva para sus inversores, lo que hace que los precios de las capacitaciones o cursos sea elevados y poco accesibles.
 - El sistema ESE solamente permite responder preguntas estáticas, que en ocasiones están repetidas.
- Respecto a lo pedagógico:
 - El sistema ofrece una cantidad de contenido limitado y monótono.
 - Las preguntas no brindan una retro alimentación de las respuestas adecuada que le de claridad al usuario porque es correcta o no su respuesta.
 - ESE no cuenta con la sugerencia de contenido basado en las falencias del usuario según le desarrollado en el entrenamiento (lectura de las preguntas y solución de las preguntas).

4.2. El Grupogear [7].

Cuenta con cursos preparatorios para las pruebas Saber 11, Saber Pro y Saber TyT con amplios contenidos informativos y relevantes para los usuarios relacionados con consejos para una mejor lectura y comprensión de textos, también cuenta con interfaces de usuarios bien formadas estructuralmente e intuitivas fáciles de navegar y atractivas a la vista; además es una de las plataformas más económicas actualmente.

Las diferencias encontradas con el presente trabajo son:

- Respecto a lo funcional:

- El Grupogeard facilita alcanzar los logros de los objetivos con diversas estrategias implementadas como blog de consejos variados en diferentes ámbitos relacionados con las áreas evaluadas por el ICFES.
- Respecto a lo pedagógico:
 - Este no cuenta con servicios de apoyo para resolver los contenidos de entrenamientos, aunque no los necesita por que los contenidos son bastante intuitivos.

4.3. AMIR [8].

Colombia cuenta con cursos de capacitación para las pruebas Saber Pro y pruebas específicas en el área de medicina, cuenta con cursos presenciales y seguimiento individualizado como alternativas para aumentar aún más las posibilidades de éxito en las pruebas, Aunque también tiene un costo elevado similar a los de la plataforma ESE.

Las diferencias encontradas con el presente trabajo son:

- Respecto a lo funcional:
 - El sistema cuenta con videos externos poco segmentados y bastante extensos que están disponibles durante 6 meses a partir de la compra del curso que cuenta con 20 de clases en video.
- Respecto a lo técnico:
 - AMIR no cuenta con una interfaz de usuario fácil de entender y tampoco proporcionan una guía didáctica para el uso de este.
- Respecto a lo pedagógico:
 - Al contar con su propia metodología de aprendizaje no brinda libertad al usuario de llevar a cabo el curso de según sus necesidades.
 - El material multimedia con el que se complementa el curso es bastante tedioso, las clases son bastante extensas y monótonas que dificultan mantener la atención.

Para más profundidad sobre el análisis a las diferentes plataformas, revisar el anexo D que contiene aspectos más detallados sobre este análisis.

5. Análisis

Este capítulo describe los procesos que se llevaron a cabo para el desarrollo de la fase de análisis de este proyecto, tales como la metodología de levantamiento de requerimientos, modelo de desarrollo, identificación de actores, etc.

Para llevar a cabo la fase de análisis se decidió optar por la mesa de trabajo como la técnica de levantamiento de requerimientos complementando con historias de usuarios de los pasantes a cargo del proyecto.

5.1. Definición de la metodología de solución

A continuación, se presentan dos propuestas de metodologías, para el desarrollo de la solución y posteriormente se argumenta respecto a la elección de una de estas propuestas.

5.1.1. Metodología secuencial lineal o en cascada

Esta metodología también conocida como ciclo de vida del software da las pautas que permiten la organización en el desarrollo del software a través de la implementación de sus diferentes etapas, el modelo en cascada se caracteriza por establecer tareas u objetivos secuenciales claros y agrupados dentro de fases o ciclos de desarrollo de un proyecto, algunas ventajas de esta metodología es que propone hacer un análisis intensivo de requerimientos dado a la naturaleza de su estructura que no permite continuar antes de terminar la fase que se está desarrollando puesto que los entregables de cada fase son la base fundamental de la siguiente. Este consta de las etapas de análisis, diseño, implementación, verificación y mantenimiento como lo muestra la FIGURA 1.

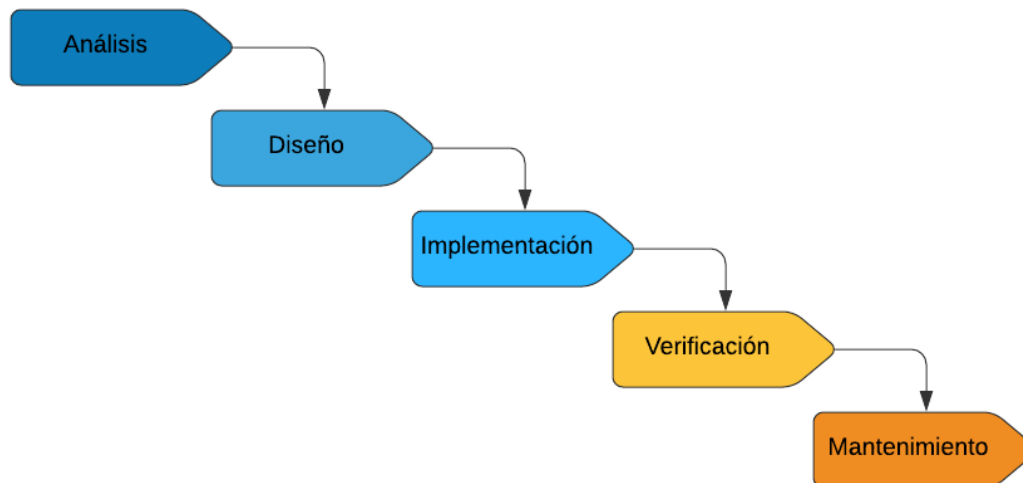


Figura 1. Esquema de Fases del Desarrollo en Cascada

Pese a que supera a otras metodologías en algunos puntos, el modelo en cascada presenta una fuerte limitación en relación con su estructura, este modelo es increíblemente rígido e inflexible lo que causa que modificar el diseño del proyecto para agregar un nuevo requerimiento en cualquier etapa sea muy complicado.

5.1.2. Metodología Proceso Racional Unificado (RUP).

RUP es una metodología de desarrollo con enfoque iterativo con una adecuada adaptación a los cambios durante el proceso de desarrollo, esta se organiza en fases y cada una demanda un conjunto de iteraciones, en cada una de ellas se van emitiendo entregables y prototipos de software que en cada iteración se van acercando más a la culminación del proyecto. Este enfoque trae como beneficios la atenuación de riesgos desde ciclos tempranos del proceso alineando las necesidades de los usuarios a las funcionalidades del producto. A su vez promueve una correcta administración del cambio y la configuración. Este consta de las fases de inicio, elaboración, desarrollo y transición expuestas en la FIGURA 2.

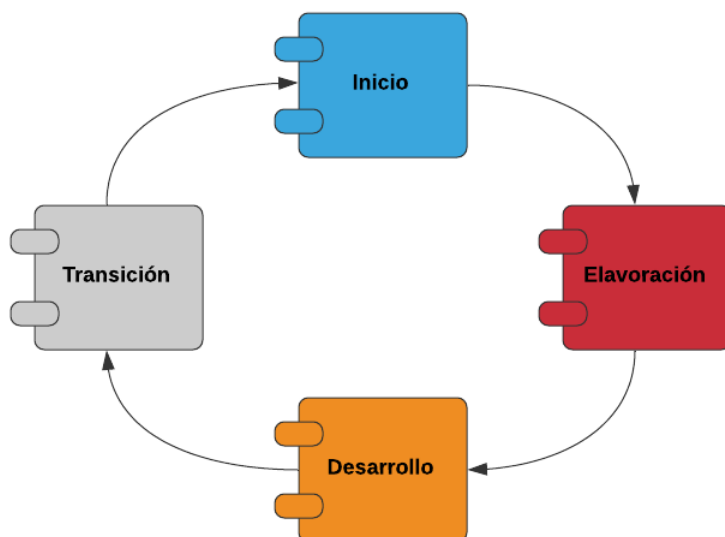


Figura 2. Esquema de Fases del Desarrollo en RUP

Por otra parte, este no es recomendable para proyectos pequeños pues este modelo genera trabajo adicional y eleva los costos de desarrollo debido a la cantidad de personal que requiere.

5.1.3. Elección de la metodología

La metodología de desarrollo seleccionada para el desarrollo de este proyecto es el modelo secuencial lineal o en cascada por las razones expuestas a continuación:

- Debido a que el proyecto está dividido en varias etapas que se llevarán a cabo por pasantes diferentes en distintos lapsos de tiempo se descarta la metodología RUP como metodología de desarrollo para llevar a cabo este proyecto.
- La estructura de la metodología secuencial lineal o en cascada se adapta perfectamente a las necesidades de este proyecto, en específico en la secuencia que debe llevarse a cabo, que en este caso solo se desarrollarán las fases de análisis y diseño.
- La cantidad de recursos que se necesitan para implementar este modelo de desarrollo de software es mínimo, además de ser extremadamente sencillo de implementar y adaptar a cualquier proyecto.

5.2. Análisis de requerimientos

Inicialmente se realizó un análisis multicriterio con el fin de evaluar algunas características que fundamentarían la selección final del aplicativo. Los criterios evaluados son características pedagógicas y funcionales, características técnicas y tiempo de desarrollo.

De la evaluación anterior, se determinó que para este proyecto el software que más se acopla a las necesidades es de tipo Hipertexto/Multimedia-Practica que funcione bajo la modalidad de software colaborativo.

- **Hipertexto/Multimedia:**

En su documento conceptos y definiciones de hipertexto Adelaide Bianchini en el 1999 quien cita a Bolter en el 1991 define:

Hipertexto consiste en tópicos y sus conexiones; los tópicos pueden ser párrafos, oraciones o palabras simples. Un hipertexto es como un libro impreso en el cual el autor tiene disponible un par de tijeras para cortar y pegar pedazos de redacción de tamaño conveniente. La diferencia es que el hipertexto electrónico no se disuelve en una desordenada carpeta de anotaciones: el autor define su estructura definiendo conexiones entre esas anotaciones[9].

- **En la definición de Conklin, 1987 se dice de hipertexto:**

“son ventanas, en una pantalla, las cuales son asociadas a objetos en una base de datos, y enlaces provistos entre estos objetos, tanto gráficamente (iconos etiquetados) como en la base de datos (apuntadores)”[9].

Tiene la finalidad de ofrecer contextos de aprendizaje no lineal, para el diseño se deben organizar el conocimiento, seleccionar los medios adecuados y definir los hiperenlaces necesarios para conectar la información.

- **Practica:** Tienen como finalidad adquirir destrezas, en cuanto al diseño se deben definir refuerzos, crear ejercicios que vayan de acuerdo con el contenido,

mantengan los niveles de dificultad adecuado y que mantengan una estructura que guíe al estudiante, Adicionalmente se debe tener una retroalimentación.

- **Hipertexto/Multimedia – Práctica:** Que contenga contextos de aprendizaje multimedia que puedan ser cargados por el docente, con prácticas sobre el desarrollo de las pruebas.
- **Software colaborativo:** según la universidad del valle “el software colaborativo es un conjunto de programas informáticos que soportan el trabajo en colaboración de grupos de personas sin importar su ubicación geográfica, con muchos usuarios concurrentes que se conectan a través de una red (internet o intranet), es decir, dos o más participantes interactúan con otro de manera que cada persona ejerce influencia y es influenciado por otra u otras personas. No existe un límite al número de participantes”[10].

Para este proyecto se entiende como software colaborativo al sistema que permite a los docentes de las diferentes instituciones educativas crear contenidos y preguntas de tipo entrenamiento y evaluación, para que los estudiantes de estas puedan revisarlo sin importar de qué institución proviene.

El desarrollo satisfactorio de este proyecto depende de los requisitos y requerimientos impuestos desde el principio. Además, al analizar detalladamente el sistema y compararlo con los sistemas ya existentes, van surgiendo restricciones y funcionalidades que se van añadiendo a la lista de requerimientos.

5.2.1. Requerimientos del módulo de administración:

- Implementar un módulo de gestión de software que permita la administración de los datos del sistema.
- Debe permitir la administración de todos los usuarios del sistema y los permisos de cada uno según su rol.
- Este debe ser capaz de permitir configurar los parámetros que define el ICFES, como las competencias que evalúan y sus derivados.

5.2.2. Requerimientos del módulo de alimentación del banco de pregunta

- Implementar un módulo de gestión de contenidos para los diferentes tipos de preguntas y los contenidos de entrenamiento.
- Cada una de las preguntas del tipo entrenamiento deberán mostrar una retroalimentación inmediata por cada una de las respuestas.
- Este debe ser capaz de almacenar y mostrar los contenidos de entrenamiento.

5.2.3.Requerimientos del módulo de ejercitación:

- Implementar un módulo que permita la visualización de los contenidos y preguntas de entrenamiento al usuario.
- Debe permitir al usuario responder las preguntas y almacenar las respuestas.
- Cada intento de respuesta debe generar una retroalimentación de valor para el usuario que le permita llegar más fácilmente a la respuesta correcta.

5.2.4.Requerimientos del módulo de evaluación:

- Implementar un módulo que simule las pruebas ICFES en el componente de las de competencias ciudadanas.
- Debe permitir al usuario evaluar su conocimiento antes y después de haber revisado los contenidos y preguntas de entrenamiento.

5.3. Análisis del sistema

Aquí se pretende identificar y desarrollar los posibles casos de uso, mostrando una representación visual de cómo están constituidos y de cómo interactúa con el sistema final.

5.3.1.Casos de uso generales

Al observar los destinos requerimientos y las funcionalidades básicas del sistema se pueden identificar 4 tipos de usuarios, teniendo en cuenta que Docente y Docente Revisor son el mismo tipo de usuario y se muestran a continuación:

Tabla 1. Actores del sistema

ACTORES DEL SISTEMA		
Actor	Descripción	Principales actividades
Administrador general.	Es el encargado de dirigir los procesos y parametrización del sistema, tiene acceso completo a todos los datos del sistema sin ninguna restricción.	<ul style="list-style-type: none"> • Administración del sistema. • Configuración de parámetros.
Institución o entidad educativa	Administrador representante de la institución o entidad educativa en el sistema, encargado de gestionar el acceso de sus estudiantes y el registro y administración de sus docentes.	<ul style="list-style-type: none"> • Registro y administración de docentes. • Autorización de estudiantes a registrar.
Docente	Encargado de la creación de contenidos para la alimentación del banco de preguntas y contenidos para el entrenamiento de los estudiantes tanto lecturas como contenidos multimedia.	<ul style="list-style-type: none"> • Alimentar el banco de preguntas y contenidos.
Docente revisor	Encargado de la revisión de la integridad de la creación de las preguntas de diferentes tipos.	<ul style="list-style-type: none"> • Revisar y hacer observaciones a las preguntas antes de ser publicadas.
Estudiante	Es el beneficiario del sistema, es el que podrá usar las herramientas para su entrenamiento en las pruebas genéricas saber pro específicamente en las competencias ciudadanas.	<ul style="list-style-type: none"> • Entrenamiento mediante la resolución de preguntas y cuestionarios.

Tabla 1.

Estos serán explicados a rasgos muy generales a continuación para tener un mejor entendimiento del sistema final:

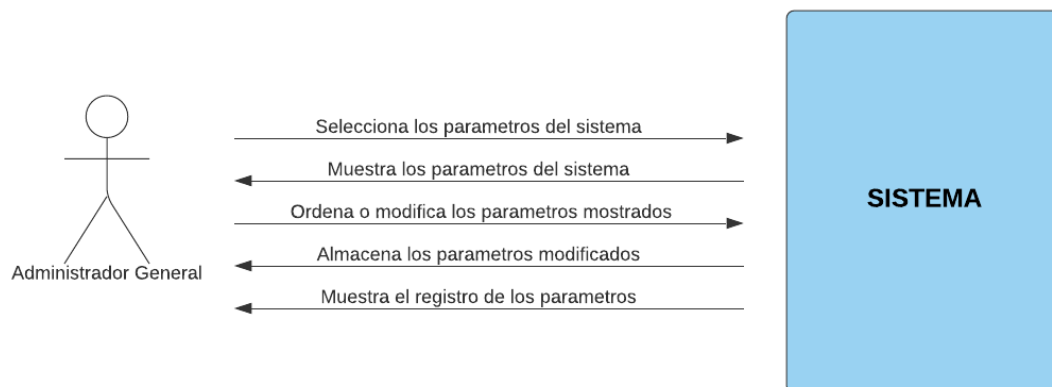


Figura 3. Caso de uso general del administrador.

El módulo de administración también permite gestionar los perfiles de usuarios, administrar los datos, hacer backup del sistema y todo lo relacionado con la administración general de un software. Aquí se establece el primer tipo de usuario, 'el administrador general' cuyo aporte de valor al sistema es la configuración de los parámetros que define el ICFES como lo son las habilidades y competencias por evaluar y otros necesarios para la óptima configuración de este.

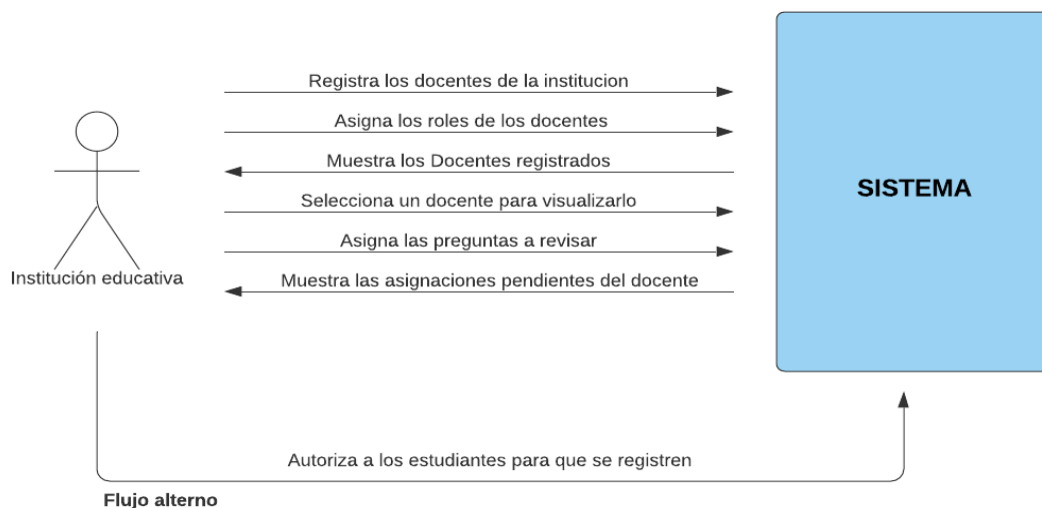


Figura 4. Caso de Uso general de las instituciones.

Las instituciones educativas son las encargadas de gestionar los docentes que estarán a cargo de la alimentación de del banco de preguntas, por lo tanto, estas deberán asignar a los docentes encargados de crear os contenido de entrenamiento y evaluación, como de asignar sus responsabilidades a los docentes como docentes revisores.

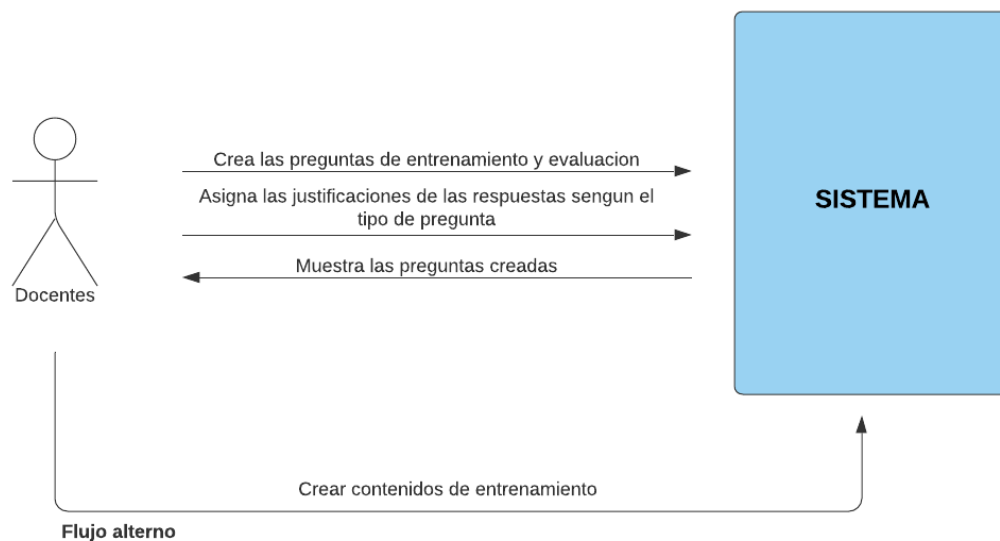


Figura 5. Caso de uso general de los docentes.

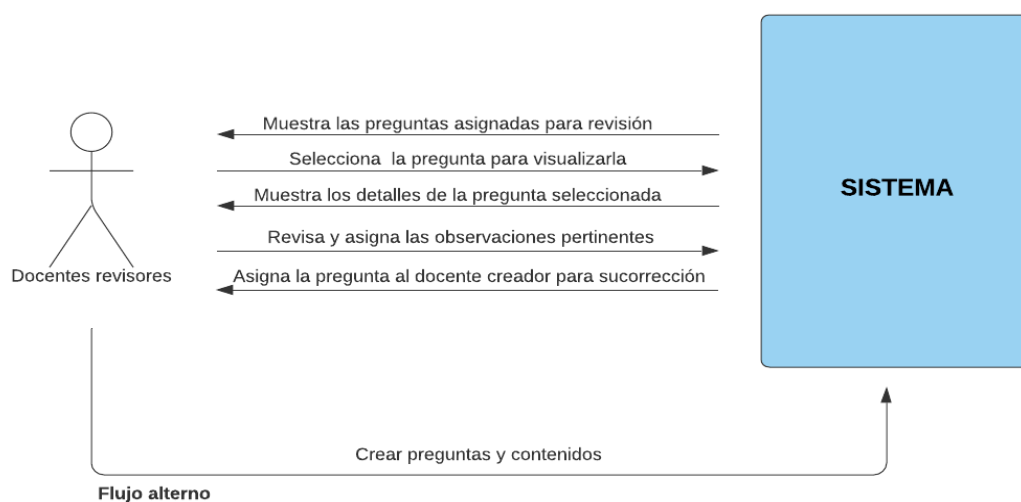


Figura 6. Caso de uso general.

Los docentes y los docentes revisores cumplen funciones en común como lo son la creación de preguntas de diferentes tipos y la creación de contenidos sin embargo sus objetivos son diferentes mientras que el docente cumple como función principal la creación de contenidos para la alimentación del banco de pregunta el docente revisor cumple como función principal la revisión de todas las preguntas creadas que se le han asignado.

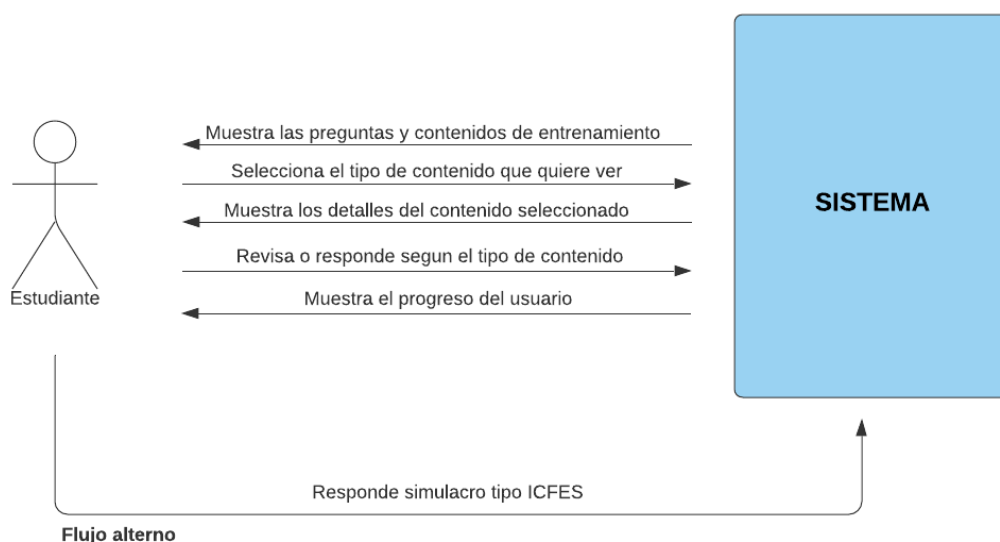


Figura 7. Caso de uso general de los estudiantes.

Los estudiantes son los beneficiarios del sistema, pues son estos los que lo utilizarán para ampliar su conocimiento en el componente de competencias ciudadanas que evalúa el ICFES, además de la posibilidad de medir su progreso en el entrenamiento mediante la realización de simulacros virtuales tipo ICFES.

Con base en lo anterior se diseñó el modelo de gestión de procesos de negocios (BPM), para definir el proceso que deberá permitir seguir el software, en el cual se define detalladamente la secuencia de los procesos que deberá seguir cada uno de los usuarios del sistema para cumplir con su función dentro del sistema.

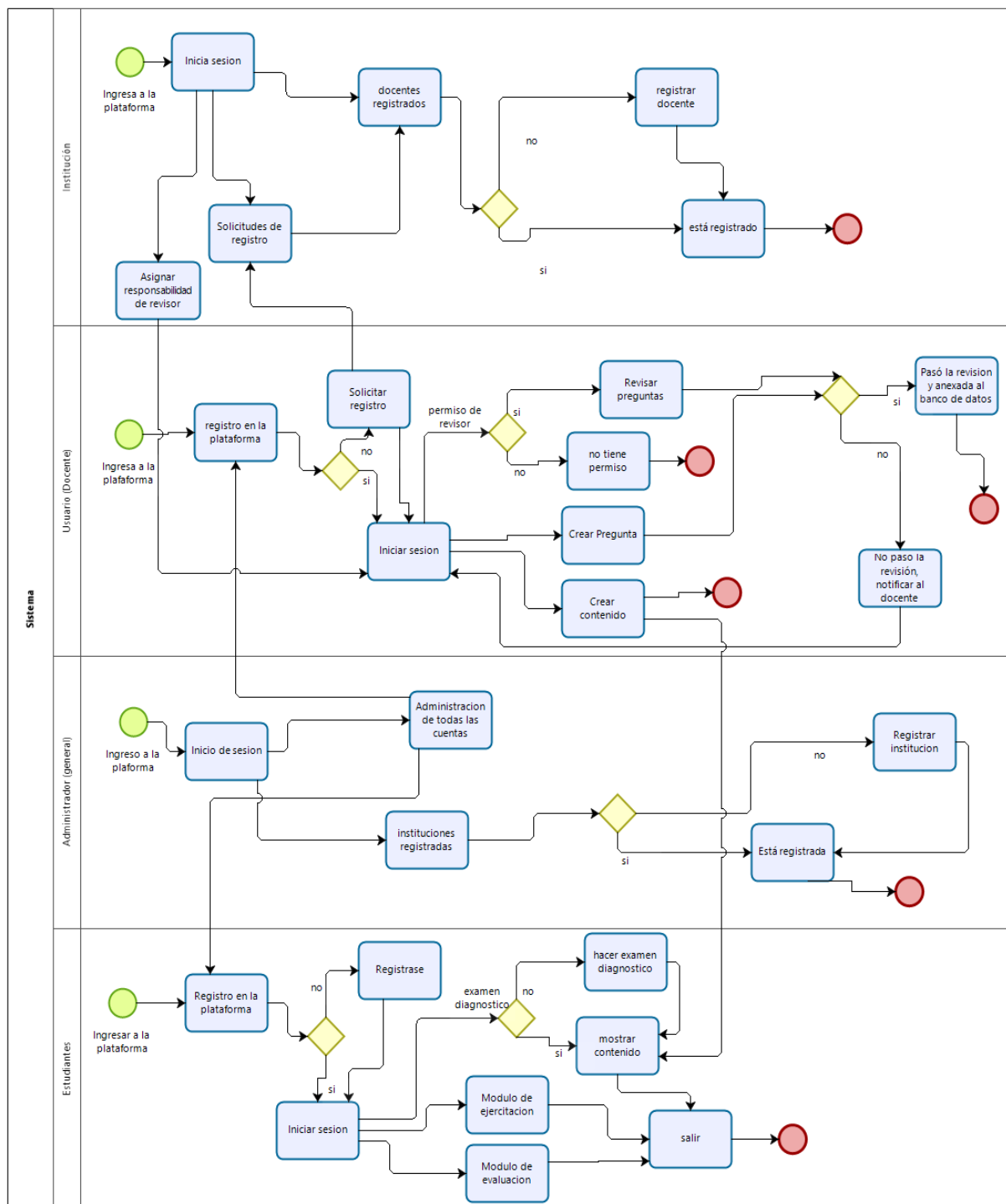


Figura 8. Modelo de gestión de procesos de negocio.

5.3.2. Casos de usos específicos y requerimiento funcionales

Teniendo en cuenta lo descrito en la Figura 6 se llevó a cabo el levantamiento detallado de los requerimientos del sistema, los cuales se enuncian a continuación y se detallaran en el Anexo A.

Tabla 2. Casos de usos

CASOS DE USO		REQUERIMIENTOS	
ID-CUS	Caso de uso	ID-REQ	Requerimiento
CUS(01)	Inicio de sesión de usuarios	REQ(01)	El administrador podrá iniciar sesión.
		REQ(02)	El personal de administración podrá iniciar sesión.
		REQ(03)	Los docentes podrán iniciar sesión.
		REQ(04)	Los estudiantes podrán iniciar sesión.
CUS(02)	Cierre de sesión de usuarios	REQ(05)	El administrador podrá cerrar sesión.
		REQ(06)	El personal de administración podrá cerrar sesión.
		REQ(07)	Los docentes podrán cerrar sesión.
		REQ(08)	Los estudiantes podrán cerrar sesión.
CUS(03)	Registrar estudiantes	REQ(09)	Los estudiantes podrán registrarse como usuarios del sistema.
CUS(04)	Registrar instituciones educativas o entidades	REQ(10)	El administrador podrá realizar el registro de las instituciones educativas o entidades.

CASOS DE USO		REQUERIMIENTOS	
ID-CUS	Caso de uso	ID-REQ	Requerimiento
CUS(05)	Registrar docentes	REQ(11)	El personal de administración podrá realizar el registro de docentes.
CUS(06)	Configuración de perfiles y gestión de usuarios	REQ(12)	El administrador podrá administrar las cuentas de las instituciones educativas o entidades.
		REQ(13)	El administrador podrá administrar las cuentas de los docentes.
		REQ(14)	El administrador podrá administrar las cuentas de los estudiantes.
		REQ(15)	el administrador podrá visualizar los usuarios registrados en el sistema.
CUS(07)	Administrar docentes	REQ(16)	El personal de administración podrá visualizar los docentes registrados en el sistema.
		REQ(17)	El personal de administración podrá administrar las cuentas de los docentes.
CUS(08)	Registrar componentes (competencias y habilidades)	REQ(18)	El administrador podrá registrar los componentes.
CUS(09)	Administrar componente	REQ(19)	El administrador podrá modificar los ítems de cada componente.
		REQ(20)	El administrador podrá visualizar el listado de los componentes registrados en el sistema.
CUS(10)	Registrar afirmaciones	REQ(21)	El administrador podrá registrar las afirmaciones
CUS(11)	Administrar afirmaciones	REQ(22)	El administrador podrá visualizar el listado de las afirmaciones registradas en el sistema.
		REQ(23)	El administrador podrá modificar los ítems de cada componente.

CASOS DE USO		REQUERIMIENTOS	
ID-CUS	Caso de uso	ID-REQ	Requerimiento
CUS(12)	Registrar evidencias	REQ(24)	El administrador podrá registrar las evidencias.
CUS(13)	Administrar evidencias	REQ(25)	El administrador podrá modificar los ítems de cada evidencia.
		REQ(26)	El administrador podrá visualizar el listado de las evidencias registradas en el sistema.
CUS(14)	Registrar temática	REQ(27)	El administrador podrá registrar las áreas temáticas.
CUS(15)	Administrar áreas temáticas	REQ(28)	El administrador podrá modificar los ítems de cada área.
		REQ(29)	El administrador podrá visualizar el listado de las áreas registradas en el sistema.
CUS(16)	Registrar nivel de complejidad	REQ(30)	El administrador podrá registrar los niveles de complejidad de las preguntas.
CUS(17)	Administrar niveles de complejidad	REQ(31)	El administrador podrá modificar los niveles de complejidad de las preguntas.
		REQ(32)	El administrador podrá visualizar el listado de los niveles de complejidad de las preguntas.
CUS(18)	Crear contenido de entrenamiento	REQ(33)	El docente podrá crear contenidos de entrenamiento
		REQ(34)	El docente podrá crear archivos de tipo texto (docx, odt, pdf etc.).
CUS(19)	Crear contenido para el banco de preguntas	REQ(35)	El docente podrá crear preguntas de tipo entrenamiento y de tipo evaluación.

CASOS DE USO		REQUERIMIENTOS	
ID-CUS	Caso de uso	ID-REQ	Requerimiento
CUS(20)	Asignar responsabilidades	REQ(36)	El personal de administración podrá asignar responsabilidades al docente (Asignar las preguntas que deben revisar).
		REQ(37)	Las preguntas se ordenan según el estado de revisión.
		REQ(38)	El personal de administración podrá ver las preguntas creadas por los docentes.
CUS(21)	Sugerir corrección de preguntas	REQ(39)	Los docentes podrán ver las preguntas que se le han asignado para revisión.
		REQ(40)	Los docentes autorizados para revisar podrán hacer observaciones de las preguntas que le sean asignadas.

CASOS DE USO		REQUERIMIENTOS	
ID-CUS	Caso de uso	ID-REQ	Requerimiento
CUS(22)	Corregir y publicar preguntas	REQ(41)	Los docentes podrán ver las preguntas que se le han creado.
		REQ(42)	Los docentes podrán corregir las preguntas después de ser revisadas, en el caso que sea necesario.
		REQ(43)	Las preguntas no podrán ser publicadas después de corregidas, si la acción asignada es <i>volver a revisar</i> .
		REQ(44)	Las preguntas con acción <i>volver a revisar</i> , después de corregidas, tienen que ser asignadas a revisión nuevamente.
		REQ(45)	Las preguntas con acción <i>no volver a revisar</i> podrán ser publicadas después de la revisión.

CASOS DE USO		REQUERIMIENTOS	
ID-CUS	Caso de uso	ID-REQ	Requerimiento
CUS(23)	Responder preguntas de entrenamiento	REQ(46)	Los estudiantes podrán responder las preguntas de entrenamiento.
		REQ(47)	Las preguntas de entrenamiento solo podrán ser respondidas una por una de manera aleatoria.
		REQ(48)	Los estudiantes podrán realizar los intentos necesarios (de acuerdo con el número de opciones de respuesta) hasta responder la pregunta de entrenamiento correctamente.
		REQ(49)	Se debe almacenar la respuesta solo del primer intento.
		REQ(50)	Para cada pregunta el estudiante podrá: <ul style="list-style-type: none"> • Cambiar de temática. • Revisar el contenido relacionado. • Continuar a la siguiente pregunta. • Finalizar el entrenamiento.
CUS(24)	Responder cuestionarios	REQ(50)	El sistema deberá crear cuestionarios de evaluación y diagnósticos aleatorios.
		REQ(51)	Los cuestionarios de evaluación y diagnósticos, sólo puede ser generado a partir de preguntas de tipo evaluación.
		REQ(52)	Los cuestionarios deberán estar formados (en número de preguntas por cada componente y porcentaje de estas) tal cual lo define el ICFES.
		REQ(53)	Los estudiantes podrán responder los cuestionarios de evaluación.
		REQ(54)	Los estudiantes podrán visualizar el historial de resultados de los cuestionarios completados por ellos.

CASOS DE USO		REQUERIMIENTOS	
ID-CUS	Caso de uso	ID-REQ	Requerimiento
CUS(25)	Retroalimentar	REQ(55)	Los estudiantes podrán visualizar contenido relacionado a sus falencias.

5.3.3.Requerimientos no funcionales

Req(56) El registro de instituciones o entidades educativas sólo debe ser realizado por el personal encargado de la administración general.

Req(57) Tanto las respuestas a las preguntas de la evaluación como las respuestas a las preguntas de entrenamiento, quedarán registradas para su explotación estadística.

Req(58) El sistema debe ser desarrollado completamente en un entorno web, para su accesibilidad en cualquier tiempo y lugar.

Req(59) Para el desarrollo se tendrá en cuenta la utilización de herramientas de desarrollo open source.

Req(60) El sistema debe cumplir las disposiciones recogidas en la ley orgánica de datos personales y en el reglamento de medidas de seguridad.

Req(61) El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente acceder (depende del número de usuarios que puedan acceder al mismo tiempo).

Req(62) El sistema debe poseer interfaces gráficas bien formadas y agradables (sugerencia: pedir apoyo a otros programas como cine para desarrollar este ítem).

Req(63) El sistema debe proporcionar mensajes de error que sean informativos y orientados al usuario final.

Req(64) Los videos que se utilicen en el aplicativo deben ser cargados a la plataforma YouTube para posteriormente anexarlos a los contenidos.

6. Diseño

Este capítulo muestra las soluciones a las funcionalidades y requerimientos mostrados anteriormente, aquí se definen los módulos necesarios para el óptimo desarrollo del sistema, así como las relaciones entre ellos.

6.1. Arquitectura general

A continuación, se describen dos patrones de diseño de software candidatos para el diseño de alto nivel del sistema web que se pretende desarrollar. Ambas propuestas cuentan con la tecnología óptima para la realización del proyecto, sin embargo, difieren en el modo de comunicación entre componentes lógicos del sistema.

6.1.1. Patrón de diseño MVC

La estructura MVC ("Model-View-Controller") es un paradigma utilizado en el desarrollo de diversos software, a través de este patrón se logra una división de las diferentes partes que conforman una aplicación, permitiendo la actualización y mantenimiento del software de una forma sencilla y en un reducido espacio de tiempo[11].

- **Modelo:** Es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo[11].
- **Vista:** Es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo[11].
- **Controlador:** Es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo[11].

Entre las ventajas del patrón de diseño MVC se destaca la separación del modelo de datos de las vistas lo que permite que existan diferentes representaciones visuales de los datos sin afectar el modelo, por ejemplo: para sistemas con soporte para distintas plataformas no altera de sobremanera el comportamiento del modelo. Por otro lado, este patrón tiene una fuerte dependencia de los eventos de las vistas.

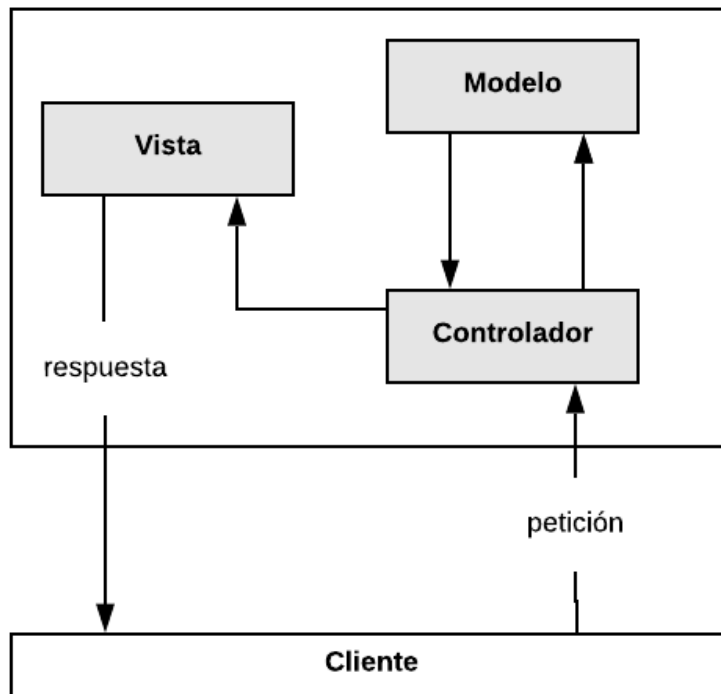


Figura 9. Interacciones del patrón MVC

6.1.2. Patrón de en N-Capas

El estilo arquitectural en n capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de la interacción con otras capas y las responsabilidades la funcionalidad que implementan[12].

- **Capa de presentación:** Es la responsable de la presentación visual de la aplicación. La capa de presentación enviará mensajes a los objetos de esta capa de negocios o intermedia, la cual o bien responderá entonces directamente o mantendrá un diálogo con la capa de la base de datos, la cual proporcionará los datos que se mandarían como respuesta a la capa de presentación[13].
- **Capa de negocios:** Es la responsable del procesamiento que tiene lugar en la aplicación. Por ejemplo, en una aplicación bancaria el código de la capa de presentación se relacionaría simplemente con la monitorización de sucesos y con el envío de datos a la capa de procesamiento. Esta capa intermedia contendría los objetos que se corresponden con las entidades de la aplicación. Esta capa intermedia es la que conlleva capacidad de mantenimiento y de reutilización[14].

- **Capa de datos:** Esta capa se encarga de acceder a los datos, se debe usar la capa de datos para almacenar y recuperar toda la información de sincronización del Sistema[12].

Es aquí donde se implementa las conexiones al servidor y la base de datos propiamente dicha, se invoca a los procedimientos almacenados los cuales reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

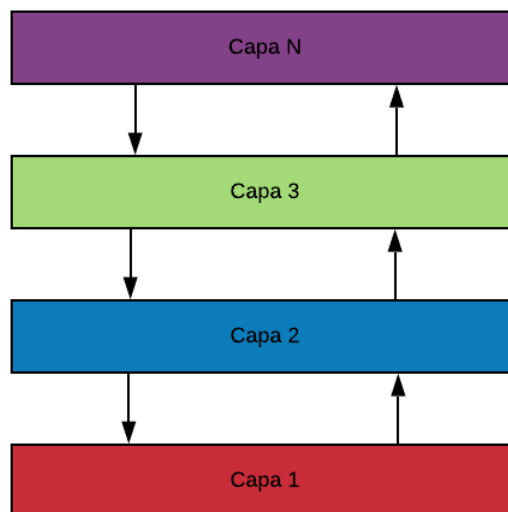


Figura 10. Interacciones del patrón N-Capas

Este modelo debido al acoplamiento de las capas, la implementación de cambios y depuración recae sobre una sola parte de la solución minimizando el impacto hacia otras capas. Esta metodología incrementa la escalabilidad del software, en contraparte el paso lineal de las capas intermedias aumenta el tiempo de respuestas de las peticiones afectando el desempeño de la aplicación.

6.1.3. Patrón de diseño de la solución

Para la implementación de la solución se utilizara el patrón de diseño de software de N-Capas adaptada a las necesidades del software, debido a la fácil escalabilidad ante la

incorporación de nuevas funcionalidades y módulos a futuro, Esta arquitectura orientada a objetos no presenta obstáculos para adaptar tanto el patrón de modelo de dominio en la capa de lógica de negocio como el patrón de repositorio en la capa de acceso a datos, cumpliendo así con los lineamientos base de diseño indicados en la fase de análisis.

- **Capa de presentación:** Esta capa integra todos los elementos de la interfaz de usuario (UI) y la lógica del comportamiento de los componentes que esta contiene, involucrando lenguaje de marcado (HTML), hojas de estilo (CSS) y lenguaje de programación (JAVASCRIPT).
- **Capa de aplicación:** Esta capa tiene como finalidad delegar las peticiones del cliente hacia los módulos de la capa de lógica de negocios.
- **Capa de lógica:** Esta capa tiene como función implementar la lógica de negocio para satisfacer los requerimientos del cliente, esta capa es la que interactúa con la capa de acceso a datos de acuerdo con el tratamiento deseado de la información intercambiada.
- **Capa de acceso a datos:** En esta capa se encuentran las conexiones encargadas de las operaciones de acceso a datos y sentencia SQL a nivel de base de datos.

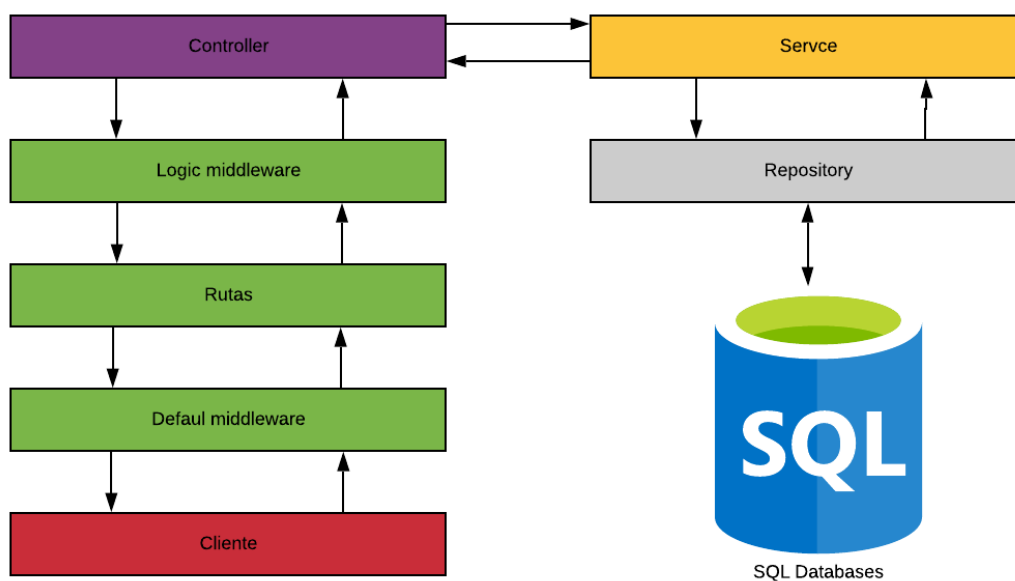


Figura 11. Flujo de arquitectura

6.2. Diseño de la Estructura de Datos

6.2.1. Diagrama de base de datos

Para este proyecto se necesita crear una base de datos para almacenar el banco de preguntas creado por los docentes, el contenido temático para fortalecer las competencias de los estudiantes, así como también los datos resultantes del entrenamiento y de la evaluación realizada a los estudiantes.

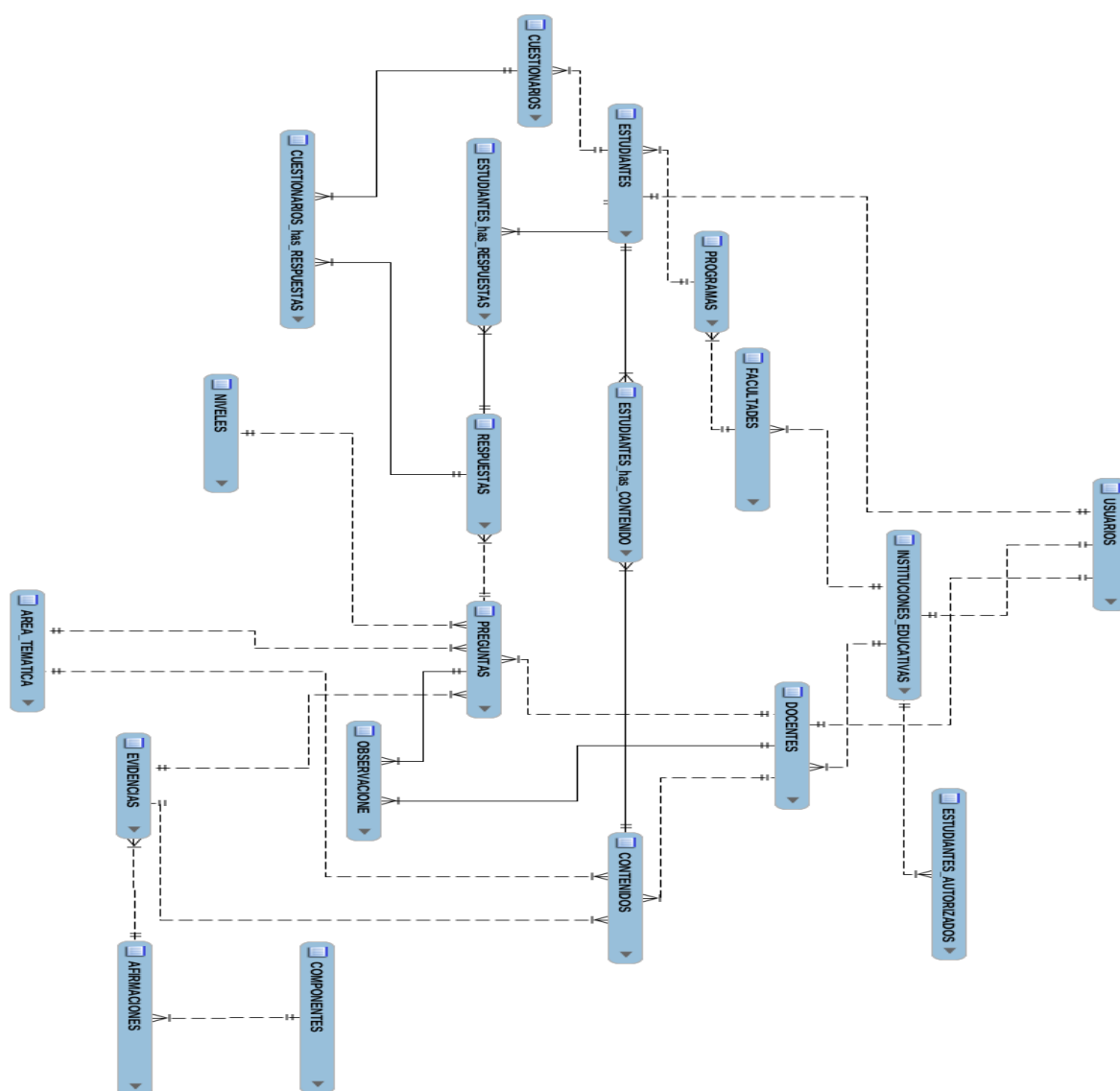


Figura 12. Diagrama de base de datos del sistema

Con base en los requerimientos presentados en el capítulo de análisis (ver Tabla 3 Casos de usos) se definieron las tablas y atributos necesarios para satisfacer la demanda de persistencia de los datos para el sistema y se muestran a continuación organizadas alfabéticamente.

Adicionalmente todas las tablas de la base de datos deben contar con los siguientes campos de forma obligatoria:

- CreatedDate de tipo DATETIME para la fecha de creación.
- UpdatedDate de tipo DATETIME para la fecha de actualización.

A continuación se muestran en detalle las tablas principales del sistema con mayor carga el modelo relacional detallado completo se encuentra en el Anexo B.

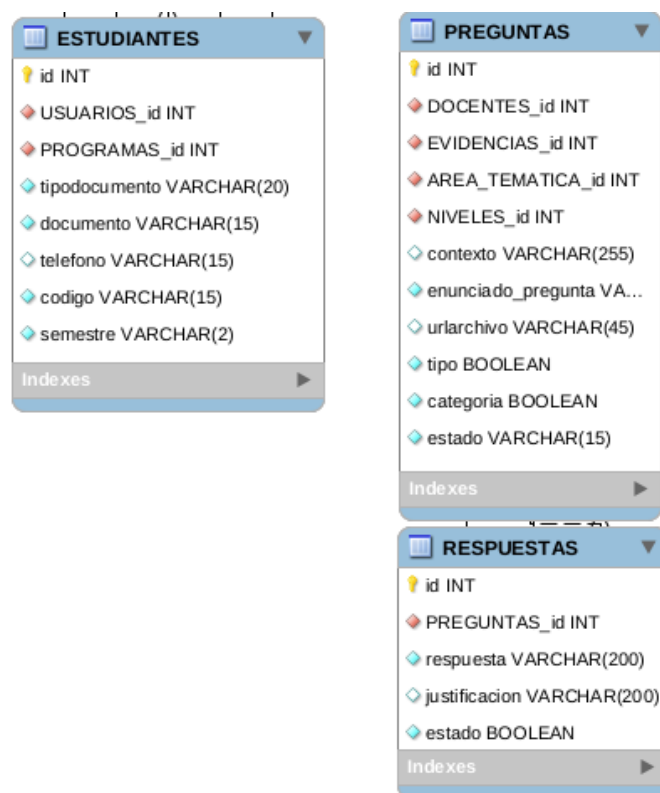


Figura 13. Tablas ESTUDIANTES-PREGUNTAS-RESPUESTAS

Tabla 3. Afirmaciones

Nombre de tabla:	Afirmaciones			
Descripción:	Son las subdivisiones de los componentes			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de afirmación
COMPONENTE_id	INT	FK	✓	Llave foránea de componente
enunciado_afirmacon	VARCHAR		✓	Enunciado de la afirmación

Tabla 4. Áreas temáticas

Nombre de tabla:	Áreas temáticas			
Descripción:	Temas que abarcan las preguntas y contenidos			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de área temática
nombre_area	VARCHAR		✓	Nombre del área temática

Tabla 5. Componentes

Nombre de tabla:	Componentes			
Descripción:	Competencias y habilidades evaluadas por el ICFES			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de componente
nombre_componente	VARCHAR		✓	Nombre de la competencia o habilidad
Porcentaje	FLOAT		✓	Porcentaje de preguntas por componente

Tabla 6. Contenidos

Nombre de tabla:	Contenidos			
Descripción:	Material de apoyo para entrenamiento (Documentos o videos)			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de contenido
DOCENTES_id	INT	FK	✓	Llave foránea de docente
AREA_TEMATICA_id	INT	FK	✓	Llave foránea de área temática
EVIDENCIAS_id	INT	FK	✓	Llave foránea de evidencia

título_contenido	VARCHAR		✓	Título del contenido
descripción	VARCHAR			Descripción del contenido
urlcontenido	VARCHAR		✓	Link del contenido
tipo	BOOLEAN		✓	Tipo de contenido (Documentos o videos)

Tabla 7. Cuestionarios

Nombre de tabla:	Cuestionarios			
Descripción:	evaluaciones realizadas a los estudiantes			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de cuestionario
ESTUDIANTES_id	INT	FK	✓	Llave foránea de estudiante
tipo	VARCHAR		✓	Tipo de cuestionario (evaluación o evaluación diagnóstica)

Tabla 8. Cuestionarios_has_Respuestas

Nombre de tabla:	Cuestionarios_has_Respuestas			
Descripción:	Respuesta de cada una de las preguntas del cuestionario			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
CUESTIONARIOS_id	INT	PK	✓	Llave primaria de cuestionario
RESPUESTAS_id	INT	PK	✓	Llave primaria de respuestas
tiempo	TIME		✓	tiempo en el que se respondió la pregunta

Tabla 9. Docentes

Nombre de tabla:	Docentes			
Descripción:	Profesores de las instituciones o entidades educativas			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de docente
INSTITUCIONES_EDUCATIVAS_id	INT	FK	✓	Llave foránea de institución educativa
USUARIO_id	INT	FK	✓	Llave foránea de usuario
autorizado	BOOLEAN		✓	Autorización para revisar preguntas

tipodocumento	VARCHAR		✓	tipo de documento de identidad
documento	VARCHAR		✓	Número de documento de identidad
teléfono	VARCHAR			Número de teléfono del docente

Tabla 10. Estudiantes

Nombre de tabla:	Estudiantes			
Descripción:	Estudiantes de las instituciones o entidades educativas			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de estudiante
USUARIO_id	INT	FK	✓	Llave foránea de usuario
PROGRAMAS_id	INT	FK	✓	Llave foránea de programa
tipodocumento	VARCHAR		✓	tipo de documento de identidad
documento	VARCHAR		✓	Número de documento de identidad
teléfono	VARCHAR			Número de teléfono del estudiante
código	VARCHAR		✓	Código estudiantil
semestre	VARCHAR		✓	Semestre actual cursado por el estudiante

Tabla 11. Estudiantes Autorizados

Nombre de tabla:	Estudiantes Autorizados			
Descripción:	Estudiantes habilitados para registrarse en el sistema			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de estudiantes autorizados
INSTITUCIONES_EDUCATIVAS_id	INT	FK	✓	Llaves primarias de institución educativa
documento	VARCHAR		✓	Número de documento del estudiante
código	VARCHAR		✓	Código estudiantil

Tabla 12. Estudiantes_has_Contenidos

Nombre de tabla:	Estudiantes_has_Contenidos			
Descripción:	Historial de contenidos revisados por los estudiantes			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
ESTUDIANTES_id	INT	PK	✓	Llave primaria de estudiante
CONTENIDOS_id	INT	PK	✓	Llaves primarias de contenido

Tabla 13. *Estudiantes_has_Respuestas*

Nombre de tabla:	Estudiantes_has_Respuestas			
Descripción:	Historial de preguntas de respondidas de tipo entrenamiento			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
ESTUDIANTES_id	INT	PK	✓	Llave primaria de estudiante
RESPUESTAS_id	INT	PK	✓	Llaves primarias de pregunta
tiempo	TIME		✓	Tiempo que demora el estudiante en responder correctamente
intentos	INT		✓	Número de intentos que hizo el estudiante antes de responder correctamente

Tabla 14. *Evidencias*

Nombre de tabla:	Evidencias			
Descripción:	Son las subdivisiones de las afirmaciones			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de evidencia
AFIRMACIONES_id	INT	FK	✓	Llave foránea de afirmación

enunciado_evidencias	VARCHAR		✓	enunciado de la evidencia
----------------------	---------	--	---	---------------------------

Tabla 15. Facultades

Nombre de tabla:	Facultades			
Descripción:	Subdivisión de la universidad			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de facultad
INSTITUCIONES_EDUCATIVAS_id	INT	FK	✓	Llave foránea de institución educativa
nombre_facultad	VARCHAR		✓	Nombre de la facultad

Tabla 16. Instituciones_Educativas

Nombre de tabla:	Instituciones_Educativas			
Descripción:	instituciones o entidades educativas registradas en el sistema			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de institución educativa
USUARIOS_id	INT	FK	✓	Llave foránea de usuario

numeroregistro	VARCHAR		✓	Número único de registro de la institución
dirección	VARCHAR		✓	Dirección de la institución
ciudad	VARCHAR		✓	Ciudad donde se encuentra ubicada la institución
teléfono				Número de teléfono de la institución

Tabla 17. Niveles

Nombre de tabla:	Niveles			
Descripción:	nivel de dificultad de respuesta de las preguntas			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de complejidad
nombre_nivel	VARCHAR		✓	nivel de complejidad de las preguntas que va de 1 a 4
tiempo	INT		✓	Tiempo estimado de respuesta por nivel, definido en minutos

Tabla 18. Observaciones

Nombre de tabla:	Observaciones			
Descripción:	Correcciones o anotaciones hechas a las preguntas antes de ser publicadas			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
DOCENTES_id	INT	PK	✓	Llave foránea de docente que realiza la corrección
PREGUNTAS_id	INT	PK	✓	Llave foránea de pregunta
descripcion	VARCHAR		✓	Descripción detallada de los errores u observaciones de la pregunta
acción	BOOLEAN		✓	acción que desea que se realice después de corregida (volver a revisar (true), no volver a revisar(false))

Tabla 19. Preguntas

Nombre de tabla:	Preguntas			
Descripción:	Preguntas de opción múltiple de tipo entrenamiento o evaluación			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de pregunta
DOCENTES_id	INT	FK	✓	Llave foránea de docente
EVIDENCIAS_id	INT	FK	✓	Llave foránea de evidencia

AREA_TEMATICA_id	INT	FK	✓	Llave foránea de área temática
NIVELES_id	INT	FK	✓	Llave foránea del nivel
contexto	VARCHAR			El caso o lectura previa a la pregunta
enunciado_pregunta	VARCHAR		✓	Enunciado de la pregunta
urlarchivo	VARCHAR			link Imagen, grafica o video previo a la pregunta
tipo	BOOLEAN		✓	Tipo de contexto (texto o multimedia)
categoría	BOOLEAN		✓	categoría de la pregunta (entrenamiento o evaluación)
estado	VARCHAR		✓	estado de revisión de la pregunta (en revisión, revisada, corregida, publicada)

Tabla 20. Programas

Nombre de tabla:	Programas			
Descripción:	Subdivisión de las facultades			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de programa
FACULTADES_id	INT	FK	✓	Llave foránea de facultad
nombre_programa	VARCHAR		✓	Nombre del programa

Tabla 21. Respuestas

Nombre de tabla:	Respuestas			
Descripción:	Opciones de respuesta de las preguntas			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de respuesta
PREGUNTAS_id	INT	FK	✓	Llave foránea de pregunta
respuesta	VARCHAR		✓	Enunciado de las respuestas
justificación	VARCHAR			Enunciado de la justificación
estado	BOOLEAN		✓	Indica si la pregunta es correcta o incorrecta

Tabla 22. Usuarios

Nombre de tabla:	Usuarios			
Descripción:	Credenciales de acceso datos comunes de los usuarios			
Nombre del campo	Tipo de dato	Llave	No nulo	Descripción
id	INT	PK	✓	Llave primaria de usuario
correo	VARCHAR		✓	Correo electrónico de usuario

contraseña	VARCHAR		✓	Contraseña de usuario
nombre_usuario	VARCHAR		✓	Nombre del usuario

6.2.2. Entorno operativo

El software que se pretende desarrollar es de tipo educativo, por tal motivo se requiere que este cumpla como facilitador del aprendizaje, es decir, definir el entorno y el tipo de software que se requiere no es una tarea al azar o de gustos. La toma de esta decisión está apoyada en análisis multicriterio que se realizó en la fase de análisis.

Teniendo en cuenta lo anterior el sistema se desarrollará bajo la alternativa Hipertexto/Multimedia-Práctica y tomando como entorno operativo un sistema web para el cual se tendrá en cuenta las tendencias de desarrollo dadas en [15].

A continuación, se describen algunas de las opciones para frontend, backend y database, tendencias en el mercado según Stack OverFlow.

6.2.2.1. Backend

Programming, Scripting, and Markup Languages

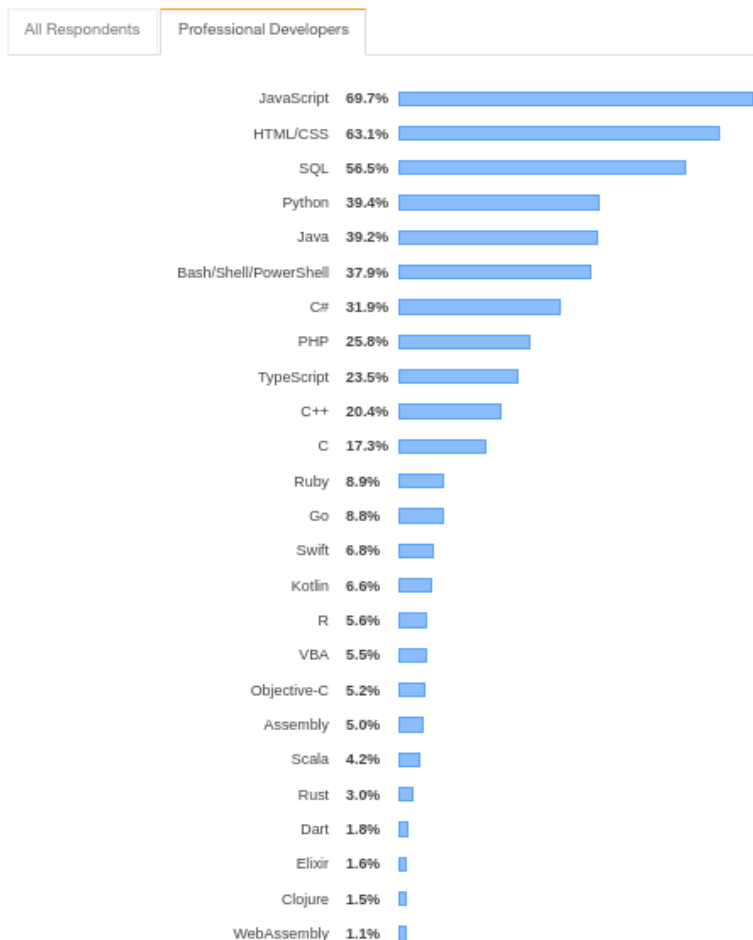


Figura 14. Lenguajes de programación más usados en 2019

- **NodeJs:**

NodeJs es ideal para manejar aplicaciones de alta concurrencia de usuarios y eventos que permite crear aplicaciones altamente escalables ya que puede soportar decenas de miles de conexiones concurrentes disminuyendo los costes de infraestructura, esto se debe a que Node utiliza el motor V8 de Google. Su parecido con JavaScript hace que la curva de aprendizaje del lenguaje sea mucho más sencilla reduciendo los tiempos de desarrollos y de trabajo.

- **Php:**

Php fue creado plenamente para el desarrollo de páginas web y plataformas web lo que facilita ampliamente el uso de acciones de formas nativas como también facilita su alojamiento debido a que actualmente el 99% de los hostings tienen soporte para proyectos php a diferencias de otros lenguajes de programación.

- **Python:**

La curva de aprendizaje de Python es adecuada, este hace énfasis en la limpieza y legibilidad del código lo que lo hace fácil de aprender. Python cuenta con varios paquetes o plugin enfocados en el procesamiento y visualización de los datos con muy poco código.

6.2.2.2. Frontend

Most Loved, Dreaded, and Wanted Web Frameworks

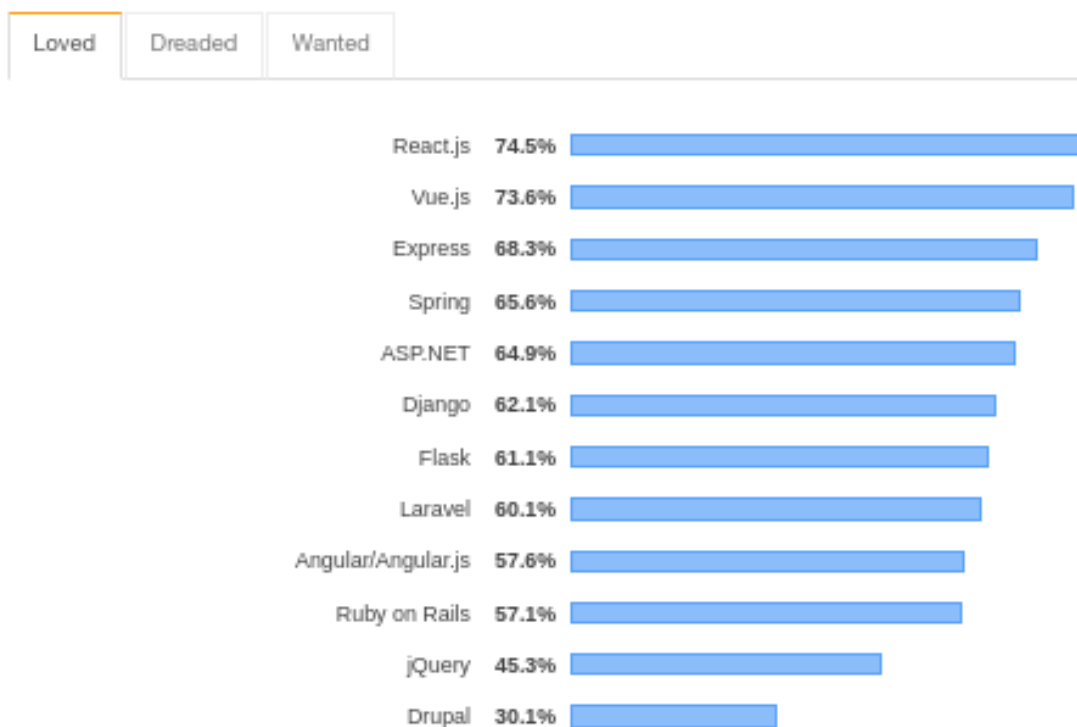


Figura 15. Web Frameworks más usados en 2019

- **Angular:**
Angular posee un marco de desarrollo robusto lo que lo hace adecuado para aplicaciones de gran tamaño, pero no es muy flexible lo cual puede ser bueno o malo según donde se implemente, angular maneja el DOM real de la aplicación lo que lo hace un poco más lento que otros framework.
- **ReactJs:**
react utiliza el DOM virtual, lo que le permite un mayor rendimiento comparado con otros framework, además puede renderizar desde el servidor utilizando Node siendo así uno de los más flexibles, pues no es un marco de desarrollo tan robusto como angular por lo que necesita librerías para tareas como el enrutamiento.

- **VueJs:**

Vue es una recopilación de los mejores conceptos de los framework lanzados antes con el mismo concepto, Vue utiliza el DOM virtual como un concepto adoptado de react. Esto garantiza un mayor rendimiento.

Esta adecuado para el desarrollo de aplicaciones ligeras, rápidas y dinámicas con una curva de aprendizaje óptima y a l igual que react puede renderizar del lado del servidor, no es tan robusto como angular, pero tiene mucho que ofrecer en si sitio web oficial como Vue Router para el enrutamiento y Vuex para el manejo de estado.

6.2.2.3. DataBase

Databases

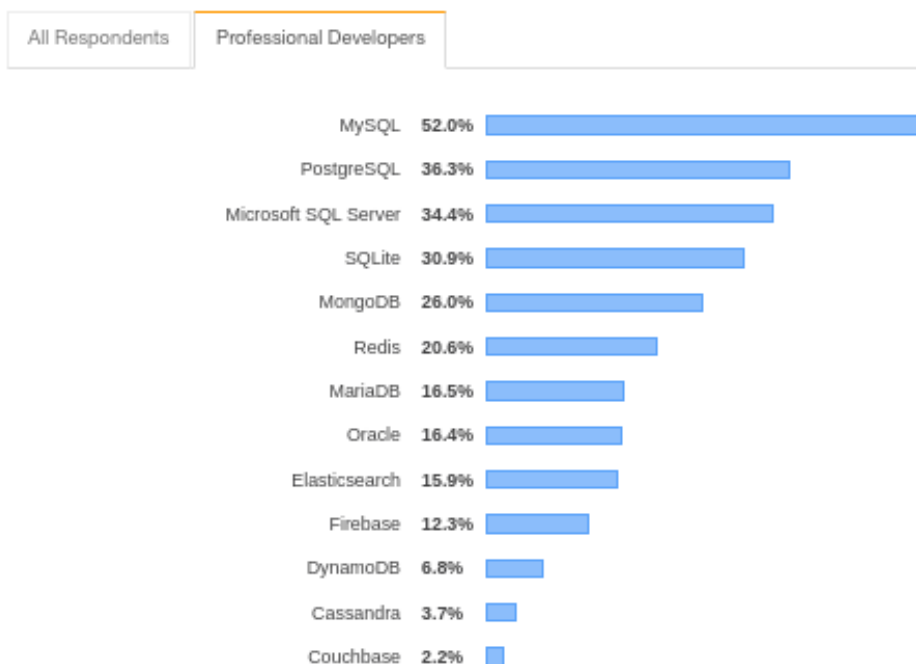


Figura 16. Motores de base de datos más usados en 2019

- **MySQL:**

MySQL incluye el conjunto más completo de funciones avanzadas, herramientas de administración y soporte técnico para lograr los niveles más altos de escalabilidad, seguridad, confiabilidad y tiempo de actividad. Reduce el riesgo, el costo y la complejidad en el desarrollo, implementación y administración.

- **PosgreSQL:**

PosgreSQL se puede configurar en cada equipo según el hardware. Por lo que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima. Con lo que se logra una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta. Está diseñado para ambientes de altos volúmenes de datos, lo que lo hace relativamente lento en inserción y actualización de tablas en comparación con bases de datos de pequeño tamaño.

- **SQL Server:**

Microsoft SQL Server es un gestor de base de datos relacional, desarrollado y mantenido por la empresa Microsoft. Utiliza el Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas.

6.2.3. Interfaz de usuario

Esta sesión mostrara las principales pantallas de las funcionalidades más importantes del sistema basados en el análisis del capítulo 1, teniendo en cuenta que no es un diseño final y está sujeto a cambios con el fin de no limitar la creatividad del diseñador final del prototipo y darle mayor libertad según las herramientas seleccionadas para el desarrollo de este.

Las interfaces se dividieron por usuarios de la siguiente manera:

6.2.3.1. Administración de usuarios.

Los diferentes parámetros que se pueden configurar en esta pantalla (ver FIGURA 7) poseen las siguientes características:

- Permite el registro de componentes según sea necesario.
- Luego de registrado el componente podrá editarlo o eliminarlo de ser necesario, teniendo en cuenta que estos parámetros los define el ICFES.
- Las mismas funcionalidades aplican para las afirmaciones y evidencias.
- Y otros parámetros propios del sistema como áreas temáticas y los niveles de dificultad de las preguntas.



Figura 17. Pantalla de administración general

6.2.3.2. Instituciones o entidades educativas.

Cada institución puede visualizar los docentes de los que dispone y gestionarlos de la forma más conveniente para dicha institución (ver FIGURA 11), como también autorizar a los estudiantes que van a poder registrarse.

DASHBOARD																																		
DOCENTES	ALUMNOS	ALUMNOS AUTORIZADOS	CONTENIDOS Y PREGUNTAS	ESTADÍSTICAS																														
<div>+</div> <h3>Lista de Docentes</h3> <table> <tr> <th>Nombre</th><th>Revisor</th><th>Telefono</th><th>Email</th><th>Accion</th></tr> <tr> <td>Luis Garrido</td><td></td><td>+5713819639</td><td>luis@ejemplo.com</td><td> </td></tr> <tr> <td>Emperatriz Zapata</td><td><input checked="" type="checkbox"/></td><td>+5713819665</td><td>emperatriz@ejemplo.com</td><td> </td></tr> <tr> <td>Elizabeth Zambrano</td><td></td><td>+5713819687</td><td>elizabeth@ejemplo.com</td><td> </td></tr> <tr> <td>Manuel Zapata</td><td><input checked="" type="checkbox"/></td><td>+5713812565</td><td>manuel@ejemplo.com</td><td> </td></tr> <tr> <td>Pedro Perez</td><td></td><td>+5713189665</td><td>pedro@ejemplo.com</td><td> </td></tr> </table>					Nombre	Revisor	Telefono	Email	Accion	Luis Garrido		+5713819639	luis@ejemplo.com		Emperatriz Zapata	<input checked="" type="checkbox"/>	+5713819665	emperatriz@ejemplo.com		Elizabeth Zambrano		+5713819687	elizabeth@ejemplo.com		Manuel Zapata	<input checked="" type="checkbox"/>	+5713812565	manuel@ejemplo.com		Pedro Perez		+5713189665	pedro@ejemplo.com	
Nombre	Revisor	Telefono	Email	Accion																														
Luis Garrido		+5713819639	luis@ejemplo.com																															
Emperatriz Zapata	<input checked="" type="checkbox"/>	+5713819665	emperatriz@ejemplo.com																															
Elizabeth Zambrano		+5713819687	elizabeth@ejemplo.com																															
Manuel Zapata	<input checked="" type="checkbox"/>	+5713812565	manuel@ejemplo.com																															
Pedro Perez		+5713189665	pedro@ejemplo.com																															

Figura 18. Pantalla de administración de las instituciones educativas.

6.2.3.3. Docentes y revisores.

La creación de contenido y preguntas de diferentes tipos requiere como mínimo que exista una coherencia en los escritos para que estos tengan sentido para el lector y sean de cierta manera fáciles de comprender, por ellos además de la creación algunos docentes están encargado de revisar y corregir estos contenidos.

Cada docente podrá administrar sus preguntas de la forma que mejor le parezca siempre y cuando esta no se encuentre publicada (ver FIGURA 12), además de hacer las sugerencias de corrección de las preguntas que le sean asignadas para revisión (ver FIGURA 13).



Figura 19. Pantalla de registro y administración de preguntas.

DASHBOARD

CONTENIDOS

PREGUNTAS

PREGUNTAS ASIGNADAS

Lista de Preguntas Asignadas

PREGUNTA	FECHA ASIGNACIÓN	ESTADO
Pregunta creada numero 1	xx/xx/xx	Revisar
Pregunta creada numero 2	xx/xx/xx	Revisar
Pregunta creada numero 3	xx/xx/xx	Revisar
Pregunta creada numero 4	xx/xx/xx	Revisar
Pregunta creada numero 5	xx/xx/xx	Revisar
Pregunta creada numero 6	xx/xx/xx	Revisar
Pregunta creada numero 7	xx/xx/xx	Revisar
Pregunta creada numero 8	xx/xx/xx	Revisada
Pregunta creada numero 9	xx/xx/xx	Revisada
Pregunta creada numero 10	xx/xx/xx	Revisada
Pregunta creada numero 11	xx/xx/xx	Revisada
Pregunta creada numero 12	xx/xx/xx	Revisada

Figura 20. Pantalla de corrección de preguntas.

Además de lo que se puede ver en el formulario (ver FIGURA 13) para cada pregunta se guarda también las fechas de creación y de actualización, el docente que crea la pregunta y el estado por defecto cuando se crea la pregunta.

DASHBOARD

CONTENIDOSPREGUNTAS

REGISTRAR PREGUNTA

☐ Evaluación☒ Entrenamiento

Seleccione un componente


Seleccione una afirmación

Seleccione una evidencia

Seleccione un área temática

Seleccione el nivel de complejidad

☐ Imagen☐ Texto☒ Video

 URL del vídeo

Enunciado

Respuesta Correcta

Justificación

Otras opciones de respuesta

Respuesta

Justificación

Respuesta

Justificación

Respuesta

Justificación

GUARDAR

Figura 21. Pantalla de registro de preguntas.

6.2.3.4. Estudiantes.

La pantalla inicial de los estudiantes (ver FIGURA 16) posee las siguientes características:

- Contenidos sugeridos al azar si estudiante aún no ha respondido la prueba diagnóstica.
- Contenidos sugeridos según las falencias del estudiante si ya respondió la prueba diagnóstica.
- Áreas temáticas por tratar en las preguntas o contenidos que se quieren ver
- Responder preguntas de tipo entrenamiento de la siguiente manera
 - Se tendrá en cuenta el primer intento de respuesta para calificar la pregunta
 - En cada pregunta se tendrá las opciones de continuar con la siguiente pregunta, cambiar de área temática, revisar el contenido relacionado con esa pregunta y finalizar el entrenamiento.

DASHBOARD

Prueba DiagnosticaSimulacro

Contenidos Sugeridos

Contenido Numero 1
Descripción Numero 1

Contenido Numero 2
Descripción Numero 2

Contenido Numero 3
Descripción Numero 3

Áreas Temáticas

Temática 1

Temática 2

Temática 3

Temática 4

Temática 5

Temática 6

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

¿Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis?

☐ Opción de respuestas A

☒ Opción de respuestas B

☐ Opción de respuestas C

☐ Opción de respuestas D

Finalizar

Contenido Sugerido

Continuar

Figura 22. Pantalla de inicio de los estudiantes.

7. Conclusiones

7.1. Conclusiones Generales.

Durante este proceso se plantearon diferentes objetivos que pudieron trazar el camino a seguir para realizar el análisis y el diseño de este proyecto de software, estos arrojaron una serie de diferentes elementos, como los requisitos tanto funcionales como no funcionales, la elaboración de casos de uso y los diferentes diagramas, tanto los BPM, los diagramas E-R y los diagramas de casos de uso.

Los procesos ejecutados para lograr el objetivo de la pasantía dan como resultado los elementos conceptuales, estructurales y de diseño necesarios para el desarrollo de las posteriores fases hasta lograr la puesta en funcionamiento de la herramienta que se construya.

7.2. Conclusiones Especificas.

- Las plataformas estudiadas durante la construcción del estado del arte permitieron maximizar las funcionalidades iniciales del proyecto ampliando los requerimientos funcionales del usuario otorgándole un valor agregado el futuro desarrollo.
- Es importante que los nuevos pasantes que desarrollarán la fase de programación tengan en cuenta las recomendaciones realizadas sobre el software de Backend, Frontend y Database. Adicionalmente puedan construir un estado del arte y un marco teórico sobre las herramientas escogidas que les permitan un mejor desempeño en esta etapa.
- Esta pasantía permitió aplicar los conocimientos adquiridos durante nuestra formación académica y lograr así habilidades que solo es posible conseguir a través de la práctica.
- Como trabajo futuro se recomienda contemplar la ampliación del sistema para abarcar las demás áreas evaluadas por el ICFES como lo son comunicación escrita, razonamiento cuantitativo, lectura crítica e inglés.

A. Anexo: Detalles de casos de uso

Anexo 1. Inicio de sesión de usuarios

Id Caso de Uso:	CUS(01)	Fecha:	N/A
Nombre:	Inicio de Sesión de usuarios		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad	Alta	Código de Requerimiento:	REQ(01), REQ(02), REQ(03) y REQ(04)
Objetivo en Contexto (Resumen):	El caso de uso describe el momento en que el usuario ingresa las credenciales de acceso al sistema.		
Actores Participantes	Administrador, instituciones o entidades educativas, Docentes y Estudiantes		
Entradas	El nombre de usuario y contraseña.		
Salidas	La posibilidad de acceso al Sistema.		

Pre-Condiciones		Estar registrado y en estado activo en el sistema.		
	Post-Condiciones		Condición final de éxito:	Acceso al sistema según el rol del usuario
			Condición final de fallo:	Correo o contraseña incorrectos
	Flujo básico de éxito			
	N o.	Actor	N o	Sistema
	1	Ingresa nombre de usuario y contraseña		
			2	Valida si el usuario y contraseña coinciden con las registradas en el sistema.
			3	Permite el acceso
	4	Finaliza la acción del caso de uso		

Anexo 2. Cierre de sesión de usuarios

	Id Caso de Uso:	CUS(02)	Fecha:	N/A
	Nombre:	cierre de Sesión de usuarios		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar		

		el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(05), REQ(06), REQ(07) y REQ(08)
Objetivo en Contexto (Resumen):		El caso debe permitir al usuario registrado en el sistema cerrar la sesión activa en el sistema .		
Actores Participantes		Administrador, instituciones o entidades educativas, Docentes y Estudiantes		
Entradas		Sesión activa		
Salidas		Página de inicio		
Pre-Condiciones		El usuario debe estar registrado en el sistema El usuario debe haber iniciado sesión en el sistema		
Post-Condiciones		Condición final de éxito:	la sesión iniciada es destruida	
		Condición final de fallo:		
Flujo básico de éxito				
N o.	Actor	N o	Sistema	
1	Selecciona el botón de cierre de sesión			
		2	Destruye la sesión iniciada	
3	Finaliza la acción del caso de uso			

Anexo 3. Registrar estudiantes

	Id Caso de Uso:	CUS(03)	Fecha:	N/A
	Nombre:	Registrar estudiantes		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(09)
	Objetivo en Contexto (Resumen):	El caso de uso le permite al estudiante registrarse como usuario del sistema		
	Actores Participantes	Estudiante		
	Entradas	Cada uno de los siguientes datos: nombre completo, tipo de documento, número de documento, teléfono, institución educativa o entidad, facultad, programa, código estudiantil, semestre, correo electrónico, contraseña.		
	Salidas	El Registro en el sistema de los datos del estudiante		

	Pre-Condiciones	Estar matriculado en una institución educativa o entidad registrada en el sistema	
Post-Condiciones		Condición final de éxito:	El sistema envía una alerta de éxito y El usuario queda almacenado correctamente en el sistema con los datos solicitados y requeridos en el formulario.
		Condición final de fallo:	Campos requeridos incompletos o mal formulados
Flujo básico de éxito			
No.	Actor	N o	Sistema
1	Selecciona la opción Registrarme		
		2	Despliega el formulario modal en pantalla para ingresar los datos requeridos
3	Digita cada uno de los campos de formulario		
		4	Valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo.
		5	Almacena los datos del registro en la base de datos y envía una alerta de éxito.
6	Finaliza la acción del caso de uso		

Anexo 4. Registrar instituciones o entidades educativas.

Id Caso de Uso:		CUS(04)	Fecha:	N/A
Nombre:		Registrar instituciones o entidades educativas		
Proyecto:		Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(10)
	Objetivo en Contexto (Resumen):		El caso de uso le permite al administrador registrar una institución o entidad educativa	
	Actores Participantes		Administrador	
	Entradas		Cada uno de los siguientes datos: nombre de la institución o entidad educativa, número de registro, dirección, ciudad, correo electrónico, número de contacto, contraseña.	
	Salidas		El Registro en el sistema de los datos de la institución o entidad educativa	
	Pre-Condiciones		Estar registrado en el sistema como administrador	
	Post-Condiciones		Condición final de éxito:	La institución o entidad educativa queda almacenada correctamente en el sistema con los datos solicitados y requeridos en el formulario.
			Condición final de fallo:	Campos requeridos incompletos o mal formulado
	Flujo básico de éxito			

	N o.	Actor	N o	Sistema
	1	Selecciona la opción administrar instituciones educativas		
			2	Despliega un listado de las instituciones o entidades educativas registradas en el sistema y un botón de agregar para registrar una nueva institución o entidad educativa
	3	Selecciona la opción de agregar		
			4	Despliega el formulario modal en pantalla para ingresar los datos requeridos
5		Digita cada uno de los campos de formulario		
8		Finaliza la acción del caso de uso	6	Si valida que los datos son correctos, el sistema realiza el ítem Número 7 del flujo
			7	Almacena los datos del registro en la base de datos y envía una alerta de éxito.

Anexo 5.Registrar docentes

Id Caso de Uso:	CUS(05)	Fecha:	N/A
Nombre:	Registrar docentes		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		

Autor:		Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad		Alta	Código de Requerimiento:	REQ(11)
Objetivo en Contexto (Resumen):		El caso de uso le permite a la institución o entidad educativa registrar un docente		
Actores Participantes		Institución o entidad educativa		
	Entradas		Cada uno de los siguientes datos: nombre completo, tipo de documento, número de documento, institución educativa(se necesita, pero no guardarla, está por defecto), correo electrónico, teléfono, rol(docente o revisor), contraseña.	
	Salidas		El Registro en el sistema de los datos del docente	
	Pre-Condiciones		Estar registrado en el sistema como institución o entidad educativa	
	Post-Condiciones		Condición final de éxito:	El docente queda almacenado correctamente en el sistema con los datos solicitados y requeridos en el formulario.
			Condición final de fallo:	Campos requeridos incompletos o mal formulado
	Flujo básico de éxito			
	N o.	Actor	N o	Sistema

	1	Selecciona la opción de administrar docentes		
			2	Despliega un listado de los docentes registrados en el sistema y un botón de agregar para registrar un nuevo docente
	3	Selecciona la opción de agregar		
			4	Despliega el formulario modal en pantalla para ingresar los datos requeridos
	5	Digita cada uno de los campos de formulario		
			6	Si valida que los datos son correctos, el sistema realiza el ítem Número 7 del flujo.
			7	Almacena los datos del registro en la base de datos y envía una alerta de éxito.
8		Finaliza la acción del caso de uso		

Anexo 6. Configuración de perfiles y gestión de usuarios

Id Caso de Uso:	CUS(06)	Fecha:	
Nombre:	Configuración de perfiles y gestión de usuarios		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0

Prioridad		Alta	Código de Requerimiento:	REQ(12), REQ(13), REQ(14), REQ(15)
Objetivo en Contexto (Resumen):		El caso de uso le permite al administrador visualizar y actualizar y eliminar la información de los usuarios registrados en el sistema		
Actores Participantes		Administrador		
Entradas		ID del usuario		
Salidas		Ver la información del usuario registrado en el sistema		
Pre-Condiciones		Estar registrado en el sistema como administrador		
	Post-Condiciones		Condición final de éxito:	El usuario es actualizado o eliminado exitosamente
			Condición final de fallo:	No se puede cargar el formulario con la información del cliente, no se ingresaron los campos requeridos, los datos ingresados son incorrectos
	Flujo básico de éxito			
	N o.	Actor	N o	Sistema
	1	Selecciona la opción administrar usuarios		
			2	Despliega un listado de los usuarios registrados en el sistema con filtros por tipos de usuarios, cada usuario cuenta con los siguientes botones: - Detalles - Editar

				- Eliminar
	3	Busca y escoge el usuario		
	4	<p>Puede seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> - Detalles que permite realizar el ítem Número 5 - Editar que permite realizar el ítem Número 6 - Eliminar que permite realizar el ítem Número 10 		
			5	Permite visualizar a través de un modal todos los detalles del usuario y permite realizar los ítems Número 6 y Número 7
			6	Despliega un formulario con la información del usuario con la opción de poder actualizar y guardar
7		Digita los cambios en el formulario y selecciona guardar		
			8	Valida que los datos ingresados puedan registrarse y actualizarse o no en la base datos, envía un mensaje de éxito
9		Selecciona la opción eliminar		
			10	<p>Muestra una alerta de confirmación con las opciones:</p> <ul style="list-style-type: none"> - Eliminar - Cancelar

11	Finaliza la acción del caso de uso		
----	------------------------------------	--	--

Anexo 7. Administrar docentes

Id Caso de Uso:	CUS(07)	Fecha:	
Nombre:	Administrar docentes		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad	Alta	Código de Requerimiento:	REQ(16), REQ(17)
	Objetivo en Contexto (Resumen):	El caso de uso le permite al personal de administración visualizar y actualizar y eliminar la información de los docentes registrados en el sistema	
	Actores Participantes	Administrador, instituciones o entidades educativas	
	Entradas	ID del usuario	
	Salidas	Ver la información del usuario registrado en el sistema	
	Pre-Condiciones	Estar registrado en el sistema como administrador o institución educativa	

	Post-Condiciones	Condición final de éxito:	El docente es actualizado o eliminado exitosamente	
		Condición final de fallo:	No se puede cargar el formulario con la información del cliente, no se ingresaron los campos requeridos, los datos ingresados son incorrectos	
	Flujo básico de éxito			
	N o.	Actor	N o	Sistema
	1	Selecciona la opción administrar docentes		
			2	Despliega un listado de los Docentes registrados en el sistema, cada usuario cuenta con los siguientes botones: - Detalles - Editar - Eliminar
	3	Busca y escoge el docente		
4	Puede seleccionar una de las siguientes opciones: - Detalles que permite realizar el ítem Número 5 - Editar que permite realizar el ítem Número 6 - Eliminar que permite realizar el ítem Número 10			

		5	Permite visualizar a través de un modal todos los detalles del docente y permite realizar los ítems Número 6 y Número 7
		6	Despliega un formulario con la información del docente con la opción de poder actualizar y guardar
7	Digita los cambios en el formulario y selecciona guardar		
		8	Valida que los datos ingresados puedan registrarse y actualizarse o no en la base datos, envía un mensaje de éxito
9	Selecciona la opción eliminar		
		10	Muestra una alerta de confirmación con las opciones: <ul style="list-style-type: none"> - Eliminar - Cancelar
11	Finaliza la acción del caso de uso		

Anexo 8. Registrar componentes (competencias y habilidades)

	Id Caso de Uso:	CUS(08)	Fecha:	
	Nombre:	Registrar componentes (competencias y habilidades)		

	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(18)
	Objetivo en Contexto (Resumen):	El caso de uso le permite al administrador registrar los componentes en el sistema		
	Actores Participantes	Administrador		
	Entradas	nombre del componente, porcentaje		
	Salidas	Visualización de los componentes registrados en el sistema con sus respectivos porcentajes		
	Pre-Condiciones	Estar Logueado en el sistema como administrador		
	Post-Condiciones	Condición final de éxito:	El componente queda registrado satisfactoriamente	
		Condición final de fallo:	Campos requeridos incompletos o mal formulado	
	Flujo básico de éxito			
	N o.	Actor	N o	Sistema
	1	Selecciona la opción registrar componente		
		2	Permite visualizar un el formulario modal con los campos requeridos	

3	Llena los campos del formulario de forma correcta		
		4	Si valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo,
		5	Almacena los datos del registro en la base de datos y envía una alerta de éxito.
6	Finaliza el caso de uso		

Anexo 9. Administrar componente

Id Caso de Uso:	CUS(09)	Fecha:	
Nombre:	Administrar componente		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad	Alta	Código de Requerimiento:	REQ(19),REQ(20)
Objetivo en Contexto (Resumen):	El caso de uso permite administrar los componentes		
Actores Participantes	Administrador		
Entradas	id del componente		
Salidas	Visualización de los componentes con sus respectivos porcentajes		
	Pre-Condiciones	Estar Logueado en el sistema como administrador	

	Post-Condiciones		Condición final de éxito:	El componente es actualizado o eliminado exitosamente
			Condición final de fallo:	Campos requeridos incompletos o mal formulado
	Flujo básico de éxito			
	N o.	Actor	N o.	Sistema
	1	El administrador selecciona la opción administrar componente		
			2	El sistema despliega un listado de los componentes registrados en el sistema, cada componente cuenta con los siguientes botones: <ul style="list-style-type: none"> - Detalles - Editar - Eliminar
	3	El usuario busca y escoge el componente		
	4	El usuario puede seleccionar una de las siguientes opciones: <ul style="list-style-type: none"> - Detalles que permite realizar el ítem Número 5 - Editar que permite realizar el ítem Número 6 - Eliminar que permite realizar el ítem Número 10 		

		5	El sistema permite visualizar a través de un modal todos los detalles del componente y permite realizar los ítems Número 6 y Número 7
		6	El sistema despliega un formulario con la información del componente con la opción de poder actualizar y guardar
7	El usuario digita los cambios en el formulario y selecciona guardar		
		8	El sistema valida que los datos ingresados puedan registrarse y actualizarse o no en la base datos, envía un mensaje de éxito
9	El usuario selecciona la opción eliminar		
		10	El sistema muestra una alerta de confirmación con las opciones: <ul style="list-style-type: none"> - Eliminar - Cancelar
11	El usuario finaliza la acción del caso de uso		

Anexo 10. Registrar afirmaciones

Id Caso de Uso:	CUS(10)	Fecha:	
Nombre:	Registrar afirmaciones		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto	Versión:	1.0.0.0

		Jose quintero			
	Prioridad	Alta	Código de Requerimiento:		REQ(21)
	Objetivo en Contexto (Resumen):		El caso de uso le permite al administrador registrar las afirmaciones en el sistema		
	Actores Participantes		Administrador		
	Entradas		nombre de la afirmación, id de la afirmación		
	Salidas		Visualización de las afirmaciones registradas en el sistema		
	Pre-Condiciones		Estar Logueado en el sistema como administrador		
	Post-Condiciones		Condición final de éxito:	La afirmación queda registrada satisfactoriamente	
			Condición final de fallo:	Campos requeridos incompletos o mal formulado	
	Flujo básico de éxito				
	No.	Actor	No	Sistema	
	1	El usuario selecciona la opción registrar afirmación			
			2	El sistema permite visualizar un el formulario modal con los campos requeridos	
	3	El usuario llena los campos del formulario de forma correcta			
			4	Si el sistema valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo	

			5	El sistema almacena los datos del registro en la base de datos y envía una alerta de éxito.
	6	El usuario finaliza el caso de uso		

Anexo 11. Administrar afirmaciones

	CUS(11)	Fecha:	
Nombre:	Administrar afirmaciones		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad	Alta	Código de Requerimiento:	REQ(22),REQ(23)
Objetivo en Contexto (Resumen):	El caso de uso permite administrar las afirmaciones		
Actores Participantes	Administrador		
Entradas	id del componente		
Salidas	Visualización de las afirmaciones registradas en el sistema		
Pre-Condiciones	Estar Logueado en el sistema como administrador		
	Condición final de éxito:	La afirmación es actualizada o eliminada exitosamente	

Post-Condiciones		Condición final de fallo:	Campos requeridos incompletos o mal formulado	
Flujo básico de éxito				
N o.	Actor		N o	Sistema
1	El administrador selecciona la opción administrar afirmaciones			
			2	El sistema despliega un listado de las afirmaciones registrados en el sistema, cada afirmación cuenta con los siguientes botones: - Detalles - Editar - Eliminar
	3	El usuario busca y escoge la afirmación		
	4	El usuario puede seleccionar una de las siguientes opciones: - Detalles que permite realizar el ítem Número 5 - Editar que permite realizar el ítem Número 6 - Eliminar que permite realizar el ítem Número 10		
			5	El sistema permite visualizar a través de un modal todos los detalles de la afirmación y permite realizar los ítems Número 6 y Número 7

			6	El sistema despliega un formulario con la información de la afirmación con la opción de poder actualizar y guardar
	7	El usuario digita los cambios en el formulario y selecciona guardar		
			8	El sistema valida que los datos ingresados puedan registrarse y actualizarse o no en la base datos, envía un mensaje de éxito
	9	El usuario selecciona la opción eliminar		
			10	El sistema muestra una alerta de confirmación con las opciones: <ul style="list-style-type: none"> - Eliminar - Cancelar
11		El usuario finaliza la acción del caso de uso		

Anexo 12. Registrar evidencias

Id Caso de Uso:	CUS(12)	Fecha:	
Nombre:	Registrar evidencias		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		

Autor:		Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad		Alta	Código de Requerimiento:	REQ(24)
Objetivo en Contexto (Resumen):		El caso de uso le permite al administrador registrar las evidencias en el sistema		
Actores Participantes		Administrador		
Entradas		nombre de la evidencia, id de la afirmación		
Salidas		Visualización de las evidencias registradas en el sistema		
Pre-Condiciones		Estar Logueado en el sistema como administrador		
	Post-Condiciones		Condición final de éxito:	La evidencia queda registrada satisfactoriamente
			Condición final de fallo:	Campos requeridos incompletos o mal formulado
	Flujo básico de éxito			
	N o.	Actor	N o.	Sistema
	1	El usuario selecciona la opción registrar evidencia		
			2	El sistema permite visualizar un el formulario modal con los campos requeridos
	3	El usuario llena los campos del formulario de forma correcta		

			4	Si el sistema valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo
			5	El sistema almacena los datos del registro en la base de datos y envía una alerta de éxito.
	6	El usuario finaliza el caso de uso		

Anexo 13. Administrar evidencias

		CUS(13)	Fecha:	
	Nombre:	Administrar evidencias		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad	Alta	Código de Requerimiento:	REQ(25),REQ(26)	
Objetivo en Contexto (Resumen):	El caso de uso permite administrar las evidencias			
Actores Participantes	Administrador			
Entradas	id de la evidencia			

Salidas		Visualización de las evidencias registradas en el sistema	
Pre-Condiciones		Estar Logueado en el sistema como administrador	
Post-Condiciones		Condición final de éxito:	La evidencia es actualizada o eliminada exitosamente
		Condición final de fallo:	Campos requeridos incompletos o mal formulado
Flujo básico de éxito			
N o .	Actor	N o	Sistema
1	El administrador selecciona la opción administrar evidencias		
		2	El sistema despliega un listado de las evidencias registrados en el sistema, cada evidencia cuenta con los siguientes botones: <ul style="list-style-type: none"> - Detalles - Editar - Eliminar
3	El usuario busca y escoge la evidencia		
	4	El usuario puede seleccionar una de las siguientes opciones: <ul style="list-style-type: none"> - Detalles que permite realizar el ítem Número 5 - Editar que permite realizar el ítem Número 6 - Eliminar que permite realizar el ítem Número 10 	

			5	El sistema permite visualizar a través de un modal todos los detalles de la evidencia y permite realizar los ítems Número 6 y Número 7
			6	El sistema despliega un formulario con la información de la evidencia con la opción de poder actualizar y guardar
	7	El usuario digita los cambios en el formulario y selecciona guardar		
			8	El sistema valida que los datos ingresados puedan registrarse y actualizarse o no en la base datos, envía un mensaje de éxito
	9	El usuario selecciona la opción eliminar		
			10	El sistema muestra una alerta de confirmación con las opciones: <ul style="list-style-type: none"> - Eliminar - Cancelar
	11	El usuario finaliza la acción del caso de uso		

Anexo 14. Registrar área temática

Id Caso de Uso:	CUS(14)	Fecha:	
Nombre:	Registrar área temática		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto	Versión:	1.0.0.0

	Jose quintero			
Prioridad	Alta		Código de Requerimiento:	REQ(27)
Objetivo en Contexto (Resumen):		El caso de uso le permite al administrador registrar las áreas temáticas en el sistema		
Actores Participantes		Administrador		
Entradas		nombre del área temática		
Salidas		Visualización de las áreas temáticas registradas en el sistema		
Pre-Condiciones		Estar Logueado en el sistema como administrador		
Post-Condiciones		Condición final de éxito:	El área temática queda registrada satisfactoriamente	
		Condición final de fallo:	Campos requeridos incompletos o mal formulado	
	Flujo básico de éxito			
	No.	Actor	No.	Sistema
	1	El usuario selecciona la opción registrar área temática		

			2	El sistema permite visualizar un el formulario modal con los campos requeridos
	3	El usuario llena los campos del formulario de forma correcta		
			4	Si el sistema valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo
			5	El sistema almacena los datos del registro en la base de datos y envía una alerta de éxito.
	6	El usuario finaliza el caso de uso		

Anexo 15. Administrar áreas temáticas

		CUS(15)	Fecha:	
	Nombre:	Administrar áreas temáticas		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(28),REQ(29)
Objetivo en Contexto (Resumen):		El caso de uso permite administrar las áreas temáticas		
Actores Participantes		Administrador		

Entradas		id del área temática	
Salidas		Visualización de las evidencias registradas en el sistema	
Pre-Condiciones		Estar Logueado en el sistema como administrador	
Post-Condiciones		Condición final de éxito:	El área temática es actualizada o eliminada exitosamente
		Condición final de fallo:	Campos requeridos incompletos o mal formulado
Flujo básico de éxito			
N o.	Actor	N o	Sistema
1	El administrador selecciona la opción administrar áreas temáticas		
		2	El sistema despliega un listado de las áreas temáticas registradas en el sistema, cada área cuenta con los siguientes botones: <ul style="list-style-type: none"> - Detalles - Editar - Eliminar
3	El usuario busca y escoge la evidencia		
	4	El usuario puede seleccionar una de las siguientes opciones: <ul style="list-style-type: none"> - Detalles que permite realizar el ítem Número 5 - Editar que permite realizar 	

		el ítem Número 6 - Eliminar que permite realizar el ítem Número 10		
			5	El sistema permite visualizar a través de un modal todos los detalles de la evidencia y permite realizar los ítems Número 6 y Número 7
			6	El sistema despliega un formulario con la información del área con la opción de poder actualizar y guardar
	7	El usuario digita los cambios en el formulario y selecciona guardar		
			8	El sistema valida que los datos ingresados puedan registrarse y actualizarse o no en la base datos, envía un mensaje de éxito
	9	El usuario selecciona la opción eliminar		
			10	El sistema muestra una alerta de confirmación con las opciones: <ul style="list-style-type: none"> - Eliminar - Cancelar
	11	El usuario finaliza la acción del caso de uso		

Anexo 16. Registrar nivel de complejidad

Id Caso de Uso:	CUS(16)	Fecha:	
Nombre:	Registrar nivel de complejidad		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
Prioridad	Alta	Código de Requerimiento:	REQ(30)
Objetivo en Contexto (Resumen):	El caso de uso le permite al administrador registrar los niveles de complejidad de las preguntas		
Actores Participantes	Administrador		
Entradas	nombre o nivel de complejidad, tiempo promedio para responder		
Salidas	Visualización de los niveles registrados en el sistema		
Pre-Condiciones	Estar Logueado en el sistema como administrador		
Post-Condiciones	Condición final de éxito:	El nivel queda registrado satisfactoriamente	
	Condición final de fallo:	Campos requeridos incompletos o mal formulado	
Flujo básico de éxito			
N o.	Actor	N o	Sistema

1	El usuario selecciona la opción registrar nivel		
			2 El sistema permite visualizar un el formulario modal con los campos requeridos
	3	El usuario llena los campos del formulario de forma correcta	
			4 Si el sistema valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo
			5 El sistema almacena los datos del registro en la base de datos y envía una alerta de éxito.
	6	El usuario finaliza el caso de uso	

Anexo 17. Administrar niveles de complejidad

		CUS(17)	Fecha:	
	Nombre:	Administrar niveles de complejidad		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(31), REQ(32)

	Objetivo en Contexto (Resumen):	El caso de uso permite administrar los niveles de complejidad de las preguntas	
	Actores Participantes	Administrador	
	Entradas	id del nivel de complejidad	
	Salidas	Visualización de los niveles de complejidad registrados en el sistema	
Pre-Condiciones		Estar Logueado en el sistema como administrador	
Post-Condiciones		Condición final de éxito:	El nivel es actualizado o eliminado exitosamente
		Condición final de fallo:	Campos requeridos incompletos o mal formulado
Flujo básico de éxito			
N o.	Actor	N o.	Sistema
1	El administrador selecciona la opción administrar niveles		
		2	El sistema despliega un listado de los niveles registrados en el sistema, cada nivel cuenta con los siguientes botones: <ul style="list-style-type: none"> - Detalles - Editar - Eliminar
3	El usuario busca y escoge el nivel de complejidad		
4	El usuario puede seleccionar una de las siguientes opciones: <ul style="list-style-type: none"> - Detalles que permite realizar el ítem Número 5 - Editar que permite realizar el ítem Número 6 		

	- Eliminar que permite realizar el ítem Número 10		
		5	El sistema permite visualizar a través de un modal todos los detalles del nivel y permite realizar los ítems Número 6 y Número 7
		6	El sistema despliega un formulario con la información del nivel con la opción de poder actualizar y guardar
7	El usuario digita los cambios en el formulario y selecciona guardar		
		8	El sistema valida que los datos ingresados puedan registrarse y actualizarse o no en la base datos, envía un mensaje de éxito
9	El usuario selecciona la opción eliminar		
		10	El sistema muestra una alerta de confirmación con las opciones: - Eliminar - Cancelar
11	El usuario finaliza la acción del caso de uso		

Anexo 18. Crear contenido de entrenamiento

	Id Caso de Uso:	CUS(18)	Fecha:	
	Nombre:	Crear contenido de entrenamiento		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		

	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(33), REQ(34)
	Objetivo en Contexto (Resumen):	El caso de uso permitirá crear contenidos de tipo texto o multimedia		
	Actores Participantes	Docente		
	Entradas	docente que lo crea, área temática, evidencia, título, descripción, url del contenido, tipo		
Salidas		Visualización del contenido que ha creado el usuario		
Pre-Condiciones		Estar Logueado en el sistema como docente		
Post-Condiciones		Condición final de éxito:	El contenido queda registrado satisfactoriamente	
		Condición final de fallo:	Campos requeridos incompletos o mal formulado	
Flujo básico de éxito				
N o.	Actor	N o	Sistema	
1	El usuario selecciona la opción registrar contenido			
		2	El sistema permite visualizar un el formulario modal con los campos requeridos	
3	El usuario llena los campos del formulario de forma correcta			
		4	Si el sistema valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo	

		5	El sistema almacena los datos del registro en la base de datos y envía una alerta de éxito.
6	El usuario finaliza el caso de uso		

Anexo 19. Crear contenido para el banco de preguntas

	Id Caso de Uso:	CUS(19)	Fecha:	
	Nombre:	Crear contenido para el banco de preguntas		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(35)

	Objetivo en Contexto (Resumen):	El caso de uso permitirá crear preguntas de tipo entrenamiento y de tipo evaluación	
	Actores Participantes	Docente	
	Entradas	docente que lo crea, área temática, evidencia, nivel de complejidad, contexto, enunciado, url del archivo, tipo, categoría, estado, respuesta correcta, justificación correcta, 3 opciones de respuestas, 3 justificaciones respectivas,	
	Salidas	Visualización de las preguntas que ha creado el usuario	
	Pre-Condiciones	Estar Logueado en el sistema como docente	
	Post-Condiciones	Condición final de éxito:	La pregunta queda registrada satisfactoriamente
		Condición final de fallo:	Campos requeridos incompletos o mal formulado
Flujo básico de éxito			
N o.	Actor	N o	Sistema
1	El usuario selecciona la opción crear pregunta		
		2	El sistema permite visualizar un el formulario modal con los campos requeridos
3	El usuario llena los campos del formulario de forma correcta		
		4	Si el sistema valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo y asigna por defecto el estado en revisión por defecto

		5	El sistema almacena los datos del registro en la base de datos y envía una alerta de éxito.
6	El usuario finaliza el caso de uso		

Anexo 20. Asignar responsabilidades

Id Caso de Uso:	CUS(20)	Fecha:	
Nombre:	Asignar responsabilidades		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento: REQ(36), REQ(37), REQ(38)
	Objetivo en Contexto (Resumen):	El caso de uso describe el momento en que el usuario asigna a un docente una pregunta para revisión	
	Actores Participantes	Institución o entidad educativa	
	Entradas	Pregunta para revisión, Docente	
	Salidas	La pregunta para revisar asignada al docente	
	Pre-Condiciones	Estar registrado y en estado activo en el sistema, y el docente a asignar la pregunta debe tener estar autorizado como revisor	

	Post-Condiciones	Condición final de éxito:	La asignación de la pregunta queda guardada satisfactoriamente en la base de datos	
		Condición final de fallo:	Campos requeridos incompletos	
	Flujo básico de éxito			
	N o.	Actor	N o	Sistema
	1	El usuario selecciona la opción asignar docente		
			2	El sistema permite visualizar en un modal los docentes disponibles para asignar
	3	El usuario selecciona el docente		
			5	El sistema almacena los datos del registro en la base de datos y envía una alerta de éxito.
	6	El usuario finaliza el caso de uso		

Anexo 21. Sugerir corrección de preguntas

Id Caso de Uso:	CUS(21)	Fecha:	
Nombre:	Sugerir corrección de preguntas		
Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0

Prioridad	Alta	Código de Requerimiento:	REQ(39), REQ(40)	
Objetivo en Contexto (Resumen):	El caso de uso permitirá crear observaciones de las preguntas asignadas para revisión			
Actores Participantes	Docente			
Entradas	docente que crea la observación, pregunta a la que pertenece la observación, descripción y la acción			
Salidas	Visualización de las preguntas que ha creado el usuario			
Pre-Condiciones	Estar Logueado en el sistema como docente Estar autorizado como revisor La pregunta está asignada al usuario que la corrige			
Post-Condiciones	Condición final de éxito:	La observación queda registrada satisfactoriamente		
	Condición final de fallo:	Campos requeridos incompletos o mal formulado		
	Flujo básico de éxito			
	No.	Actor	No	Sistema
	1	El usuario selecciona la opción agregar observación		
			2	El sistema permite visualizar un el formulario con los campos requeridos

	3	El usuario llena los campos del formulario de forma correcta		
			4	Si el sistema valida que los datos son correctos, el sistema realiza el ítem Número 5 del flujo y asigna por defecto el estado en revisión por defecto
			5	El sistema almacena los datos del registro en la base de datos, cambia el estado de la pregunta de asignada a revisada y envía una alerta de éxito.
	6	El usuario finaliza el caso de uso		

Anexo 22. Corregir y publicar preguntas

	Id Caso de Uso:	CUS(22)	Fecha:	
	Nombre:	Corregir y publicar preguntas		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(41), REQ(42), REQ(43), REQ(44), REQ(45)
	Objetivo en Contexto (Resumen):	El caso de uso permitirá modificar las preguntas para hacer las correcciones necesarias que indican las observaciones que se le hizo		
	Actores Participantes	Docente		
	Entradas	campos para modificar de la pregunta		

Salidas		Visualización de las preguntas actualizadas que ha creado el usuario	
Pre-Condiciones		Estar Logueado en el sistema como docente la pregunta se le debió hacer una observación previamente	
Post-Condiciones	Condición final de éxito:	La pregunta se actualiza satisfactoriamente	
	Condición final de fallo:	Campos requeridos incompletos o mal formulado	
Flujo básico de éxito			
N o.	Actor	N o.	Sistema
1	El usuario selecciona la opción corregir pregunta		
		2	El sistema permite visualizar un el formulario completo de preguntas con los campos llenos con la información correspondiente
3	El usuario modifica los campos del formulario según sea necesario		
4	El usuario tiene las siguientes posibles opciones: - guardar y publicar en caso de que la acción sea no volver a revisar ítem número 5 - actualizar en caso de que la acción se volver a revisar ítem número 6		
		5	El sistema almacena los cambios y deja la pregunta lista para ser usada por los estudiantes

			6	El sistema almacena los cambios y deja la pregunta nuevamente en revisión
	7	El usuario finaliza el caso de uso		

Anexo 23. Responder preguntas de entrenamiento

	Id Caso de Uso:	CUS(23)	Fecha:	
	Nombre:	Responder preguntas de entrenamiento		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(46), REQ(47), REQ(48), REQ(49), REQ(50)
	Objetivo en Contexto (Resumen):	El caso de uso permitirá a los estudiantes responder las preguntas de entrenamiento publicadas		
	Actores Participantes	Estudiante		
	Entradas	primer intento de respuesta, tiempo, intentos		
	Salidas	retroalimentación según la respuesta seleccionada		
	Pre-Condiciones	Estar Logueado en el sistema como estudiante Haber respondido la prueba diagnóstica		

Post-Condiciones		Condición final de éxito:	La respuesta queda almacenada satisfactoriamente
		Condición final de fallo:	
Flujo básico de éxito			
N o.	Actor	N o	Sistema
1	Selecciona del menú principal la opción de entrenamiento		
		2	Permite visualizar las áreas temáticas que están disponible
3	Selecciona el área temática de su interés		
		4	Permite visualizar las preguntas de la temática seleccionada una a una con las opciones de: <ul style="list-style-type: none"> - Cambiar de área temática ítem número 7 - Revisar el contenido relacionado ítem número 8 - continuar a la siguiente pregunta ítem número 9 - finalizar el entrenamiento ítem número 10
5	Responde la pregunta y realiza el ítem número 6 o selecciona una de las opciones del ítem número 4		
		6	Almacena la respuesta en el sistema, permite mostrar la justificación de la respuesta seleccionada y permite volver a responder en el caso de que la respuesta sea errónea
		7	permite visualizar el ítem número 2

			8	Permite visualizar los contenidos relacionados con la pregunta
			9	Muestra en pantalla la siguiente pregunta
			10	Finaliza el entrenamiento y redirige a la pantalla principal
	11	Finaliza el caso de uso		

Anexo 24. Responder cuestionarios

	Id Caso de Uso:	CUS(24)	Fecha:	
	Nombre:	Responder cuestionarios		
	Proyecto:	Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimiento:	REQ(50), REQ(52), REQ(54) REQ(51), REQ(53),
	Objetivo en Contexto (Resumen):	El caso de uso permitirá a los estudiantes responder las preguntas de los cuestionarios ya sean de evaluación o diagnósticos		
	Actores Participantes	Estudiante		
	Entradas	estudiante, tipo de cuestionario, respuesta de cada pregunta del cuestionario, tiempo de cada respuesta		
	Salidas	Resultados del cuestionario respondido		

	Pre-Condiciones		Estar Logueado en el sistema como estudiante	
Post-Condiciones		Condición final de éxito:	Las respuestas quedan almacenadas satisfactoriamente	
		Condición final de fallo:		
Flujo básico de éxito				
N o.	Actor		N o	Sistema
1	Selecciona del menú principal la opción de evaluación			
			2	Genera un cuestionario al azar de tipo diagnóstico la primera vez y de tipo evaluación las veces siguientes
3	Responde una a una el total de las preguntas			
			4	Almacena las respuestas en el sistema
			5	permite visualizar el resultado de la evaluación
6	Finaliza el caso de uso			

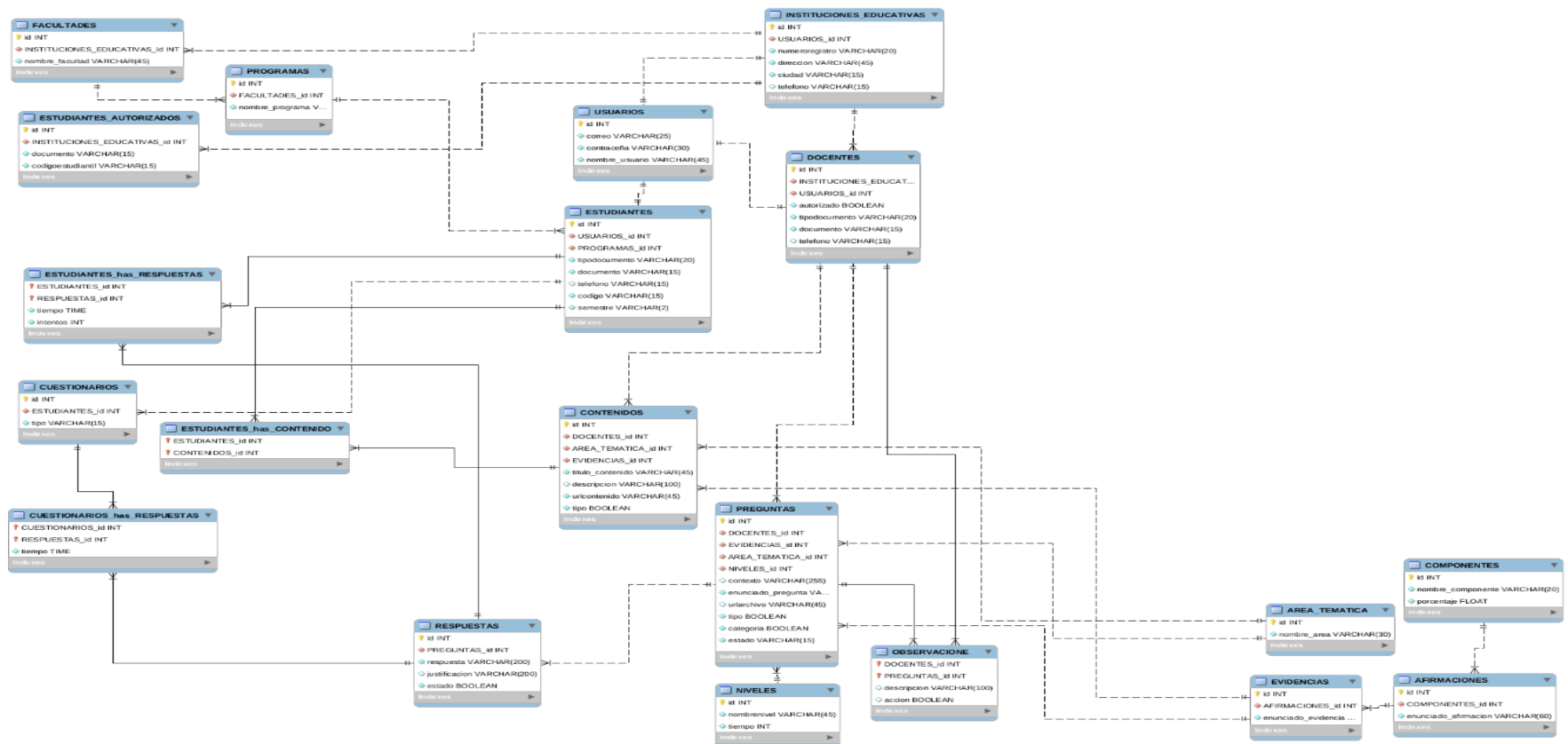
Anexo 25. Retroalimentar

Id Caso de Uso:	CUS(25)	Fecha:	
Nombre:	Retroalimentar		

Proyecto:		Análisis y Diseño de los módulos de Administración y Alimentación del Banco de Preguntas que hacen parte del software de práctica para apoyar el desarrollo de las habilidades en competencias ciudadanas de estudiantes universitarios.		
	Autor:	Eric pinto Jose quintero	Versión:	1.0.0.0
	Prioridad	Alta	Código de Requerimien to:	REQ(55)
	Objetivo en Contexto (Resumen):		El caso de uso permitirá a los estudiantes visualizar contenidos relacionado a sus falencias	
	Actores Participantes		Estudiante	
	Entradas		Estadísticas del progreso del estudiante	
	Salidas		Sugerencias de contenidos según las falencias evidenciadas	
	Pre-Condiciones		Estar Logueado en el sistema como estudiante	
	Post-Condiciones		Condición final de éxito:	Los contenidos sugeridos dejan de ser al azar y se vuelven personalizados para cada estudiante
			Condición final de fallo:	
	Flujo básico de éxito			
	N o.	Actor	N o	Sistema
	1	Navega a la página de inicio		
			2	Permite visualizar contenidos focalizados en las falencias del estudiante
	3	finaliza el caso de uso		

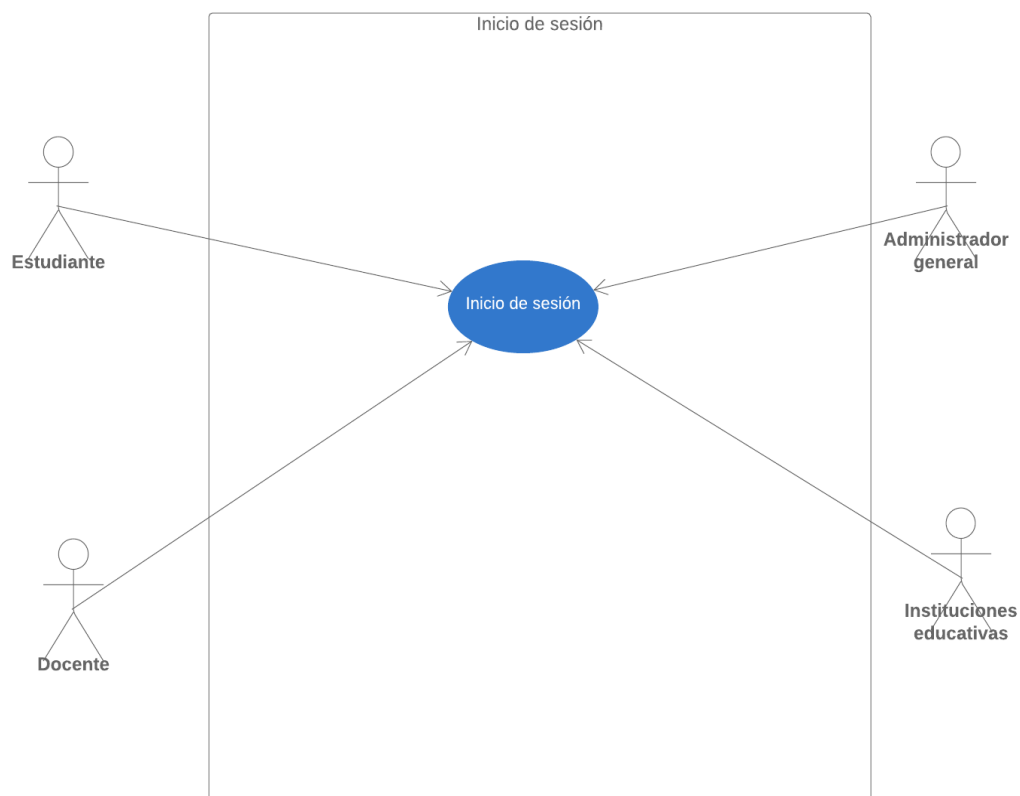
B. Anexo: Modelo relacional detallado

Anexo 26. Base de datos

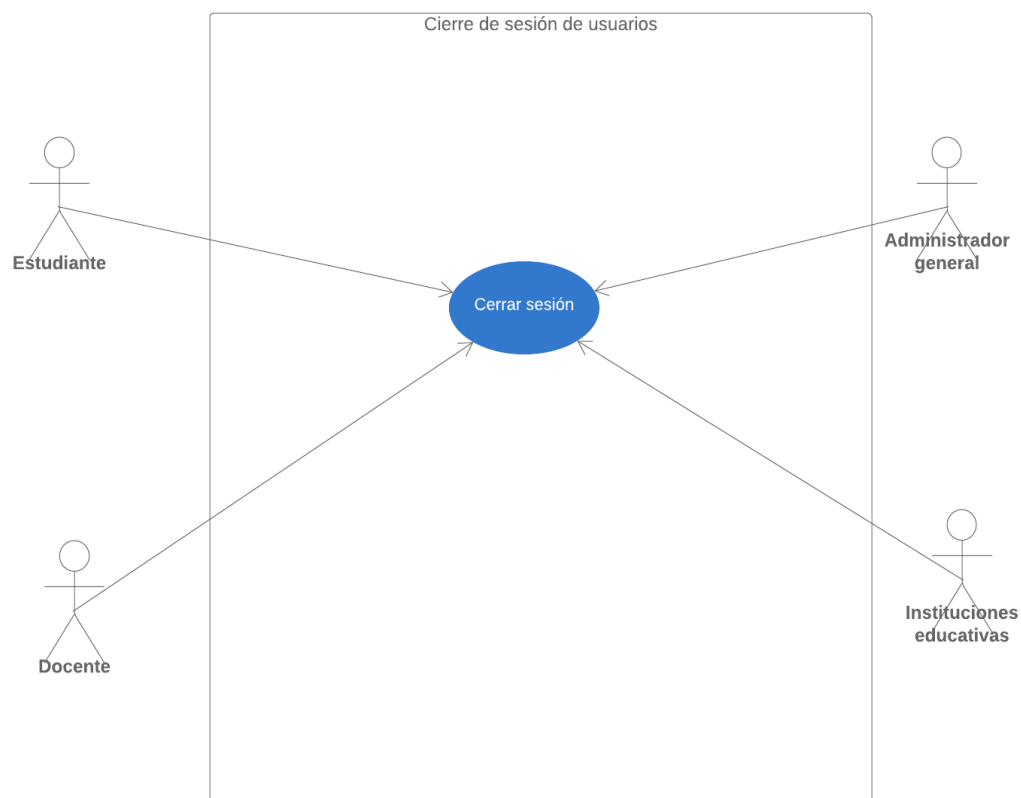


C. Anexo: Diagramas de casos de uso

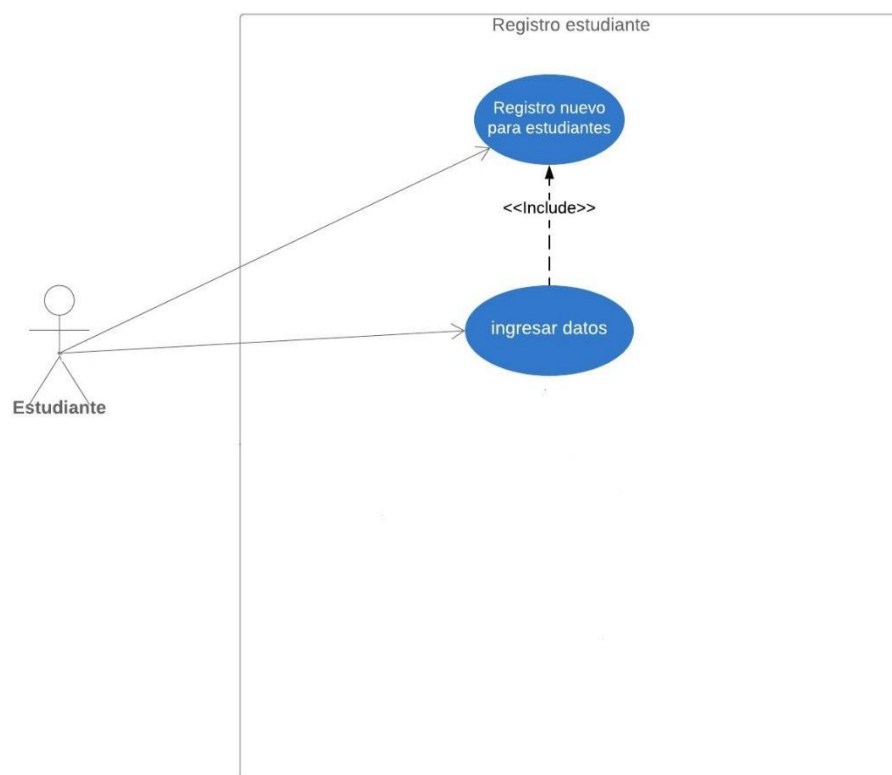
Anexo 27. Diagrama Inicio de sesión



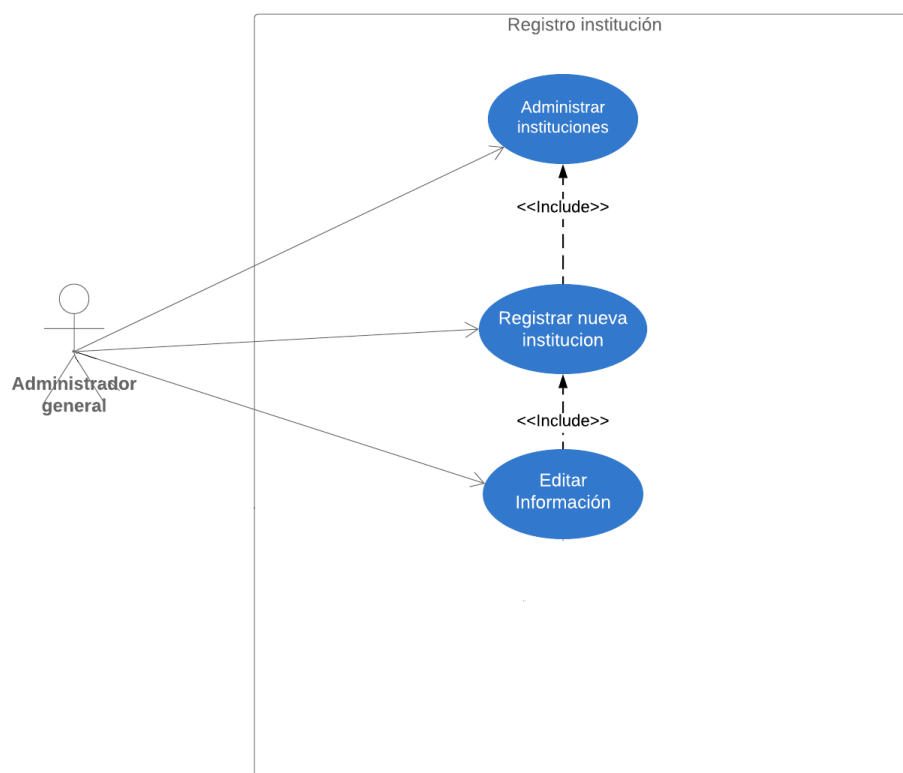
Anexo 28. Diagrama Cierre de sesión de usuarios



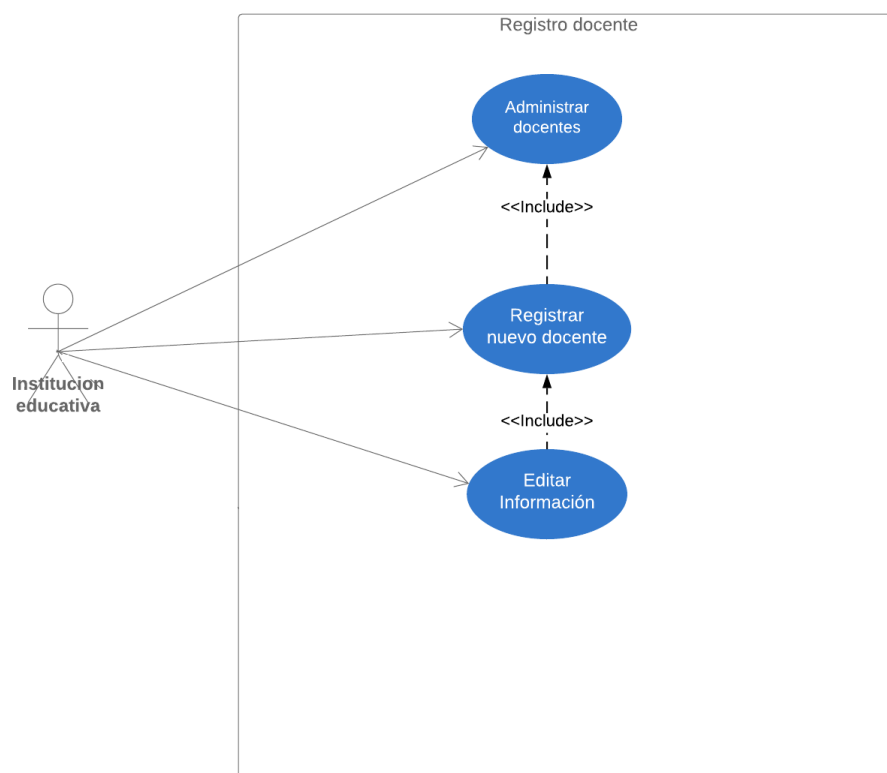
Anexo 29. Diagrama Registro estudiante



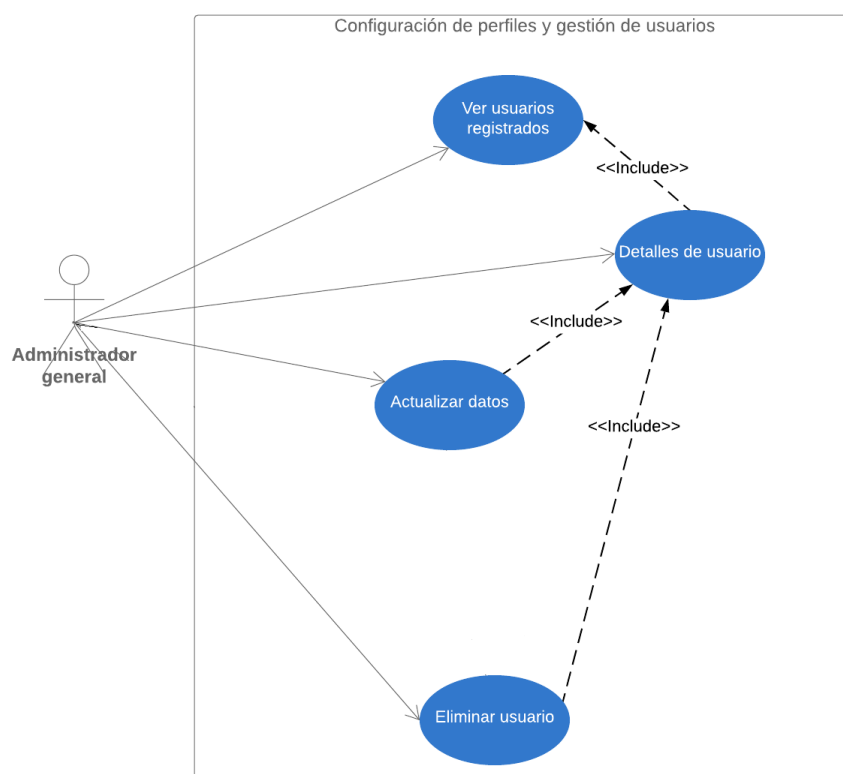
Anexo 30. Diagrama Registro institución



Anexo 31. Diagrama Registro docente



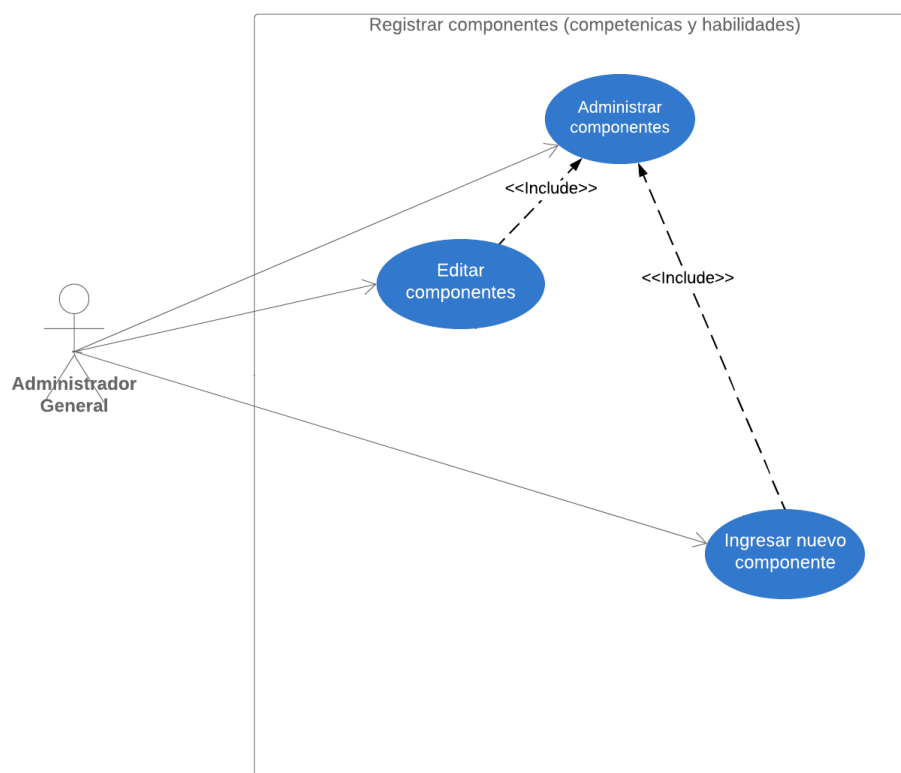
Anexo 32. Diagrama Configuración de perfiles y gestión de usuarios



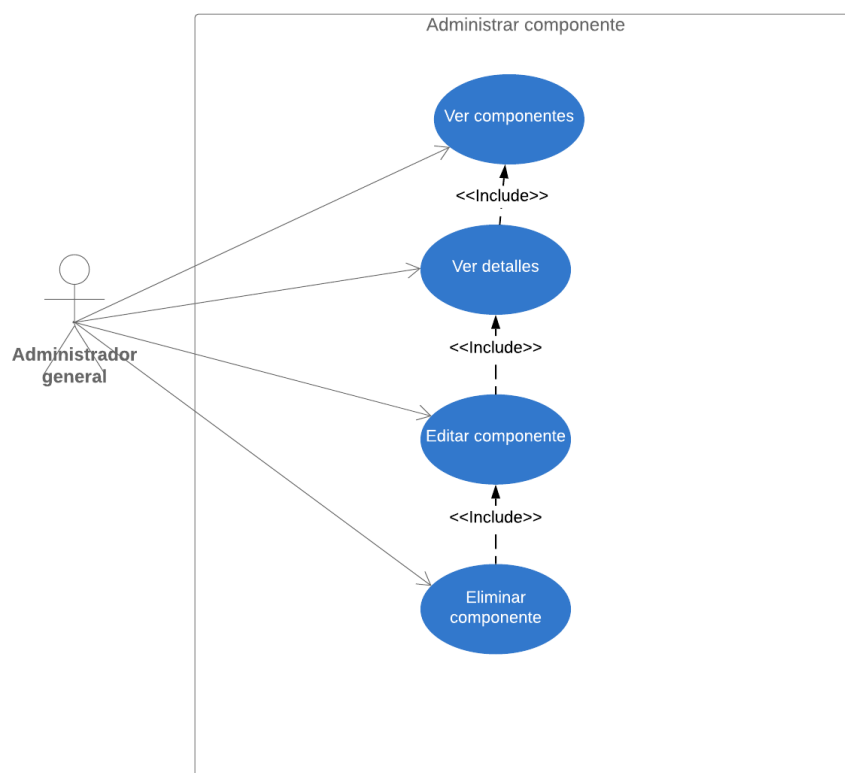
Anexo 33. Diagrama Administrar docente



Anexo 34. Diagrama Registrar componentes (competencias y habilidades)



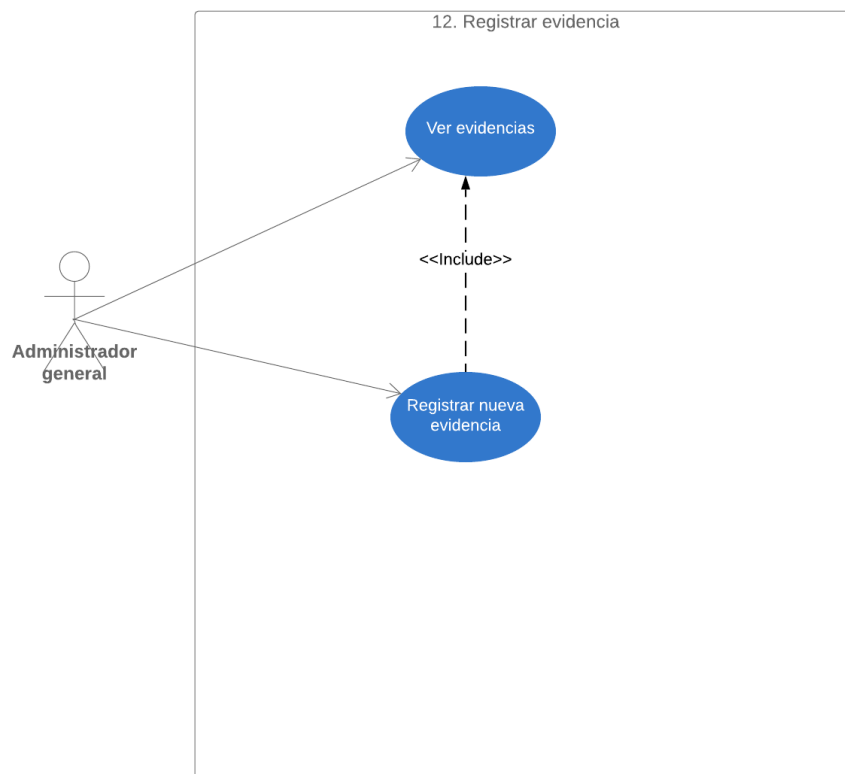
Anexo 35. Diagrama Administrar componente

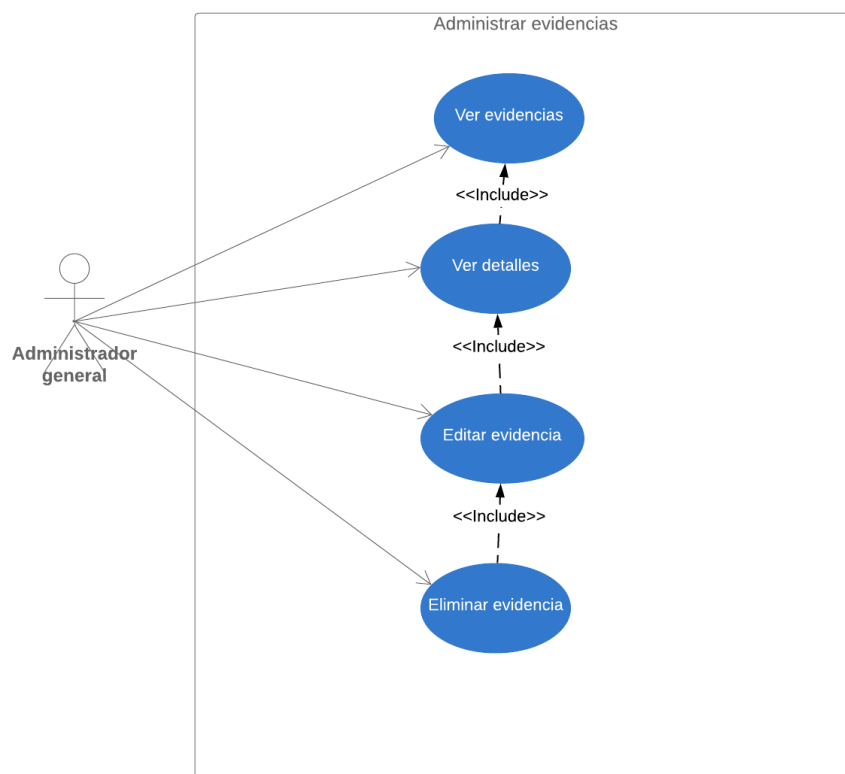


Anexo 36. Diagrama Registrar Afirmaciones



Anexo 37. Diagrama Administrar afirmaciones

Anexo 38. Diagrama Registrar evidencia

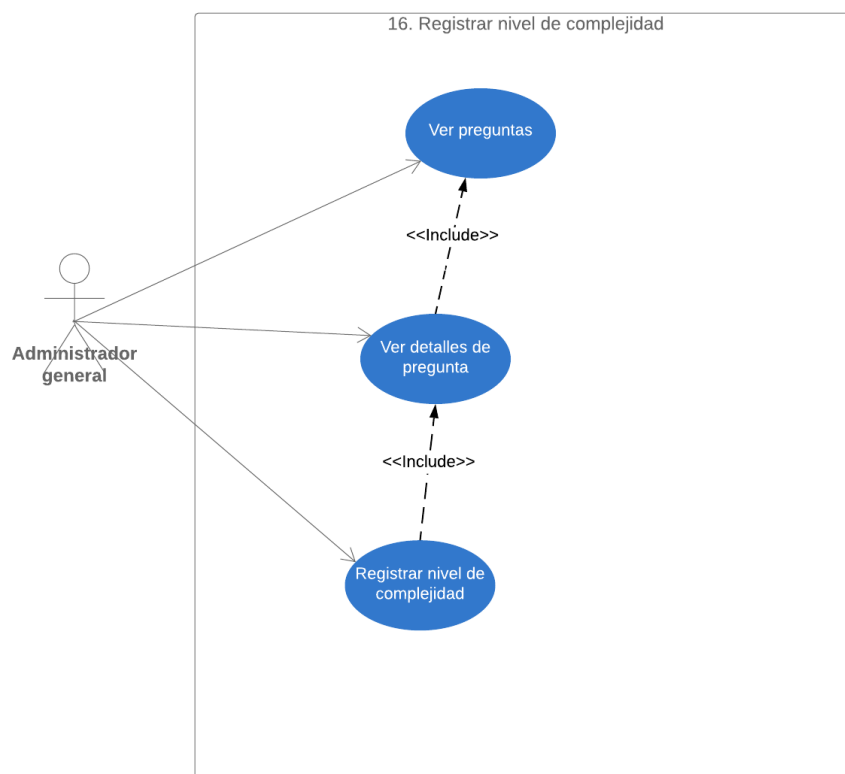
Anexo 39. Diagrama Administrar evidencias

Anexo 40. Diagrama Registrar área temática

Anexo 41. Diagrama Administrar área temática

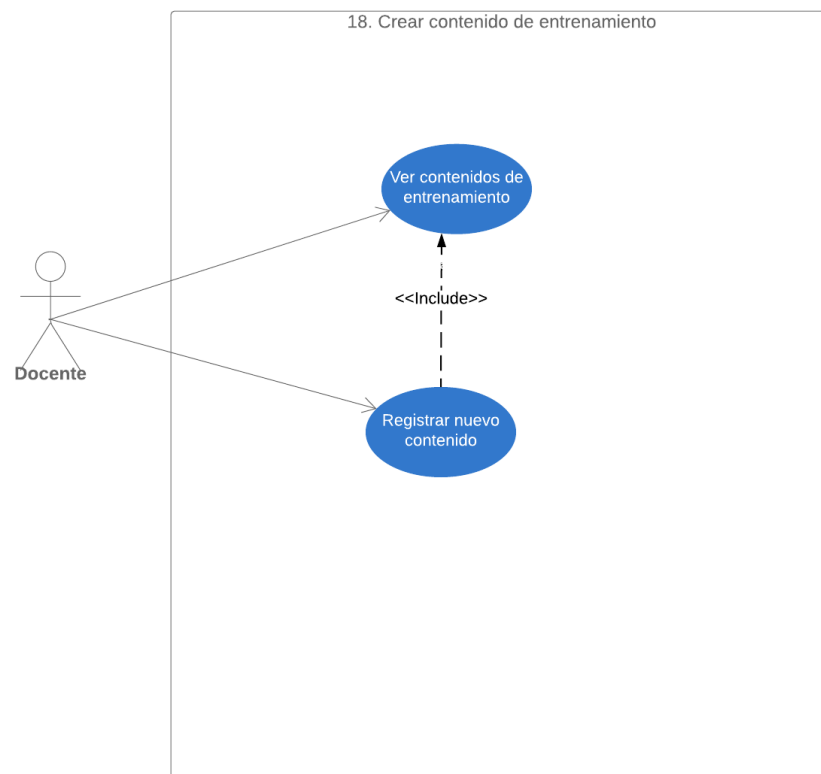


Anexo 42. Diagrama Registrar nivel de complejidad

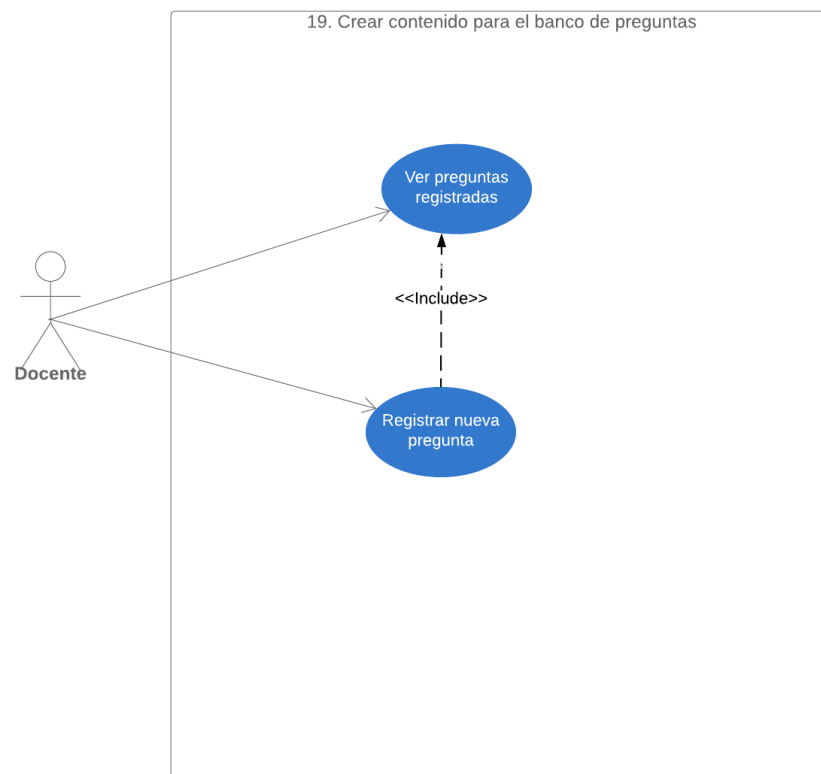


Anexo 43. Diagrama Administrar nivel de complejidad

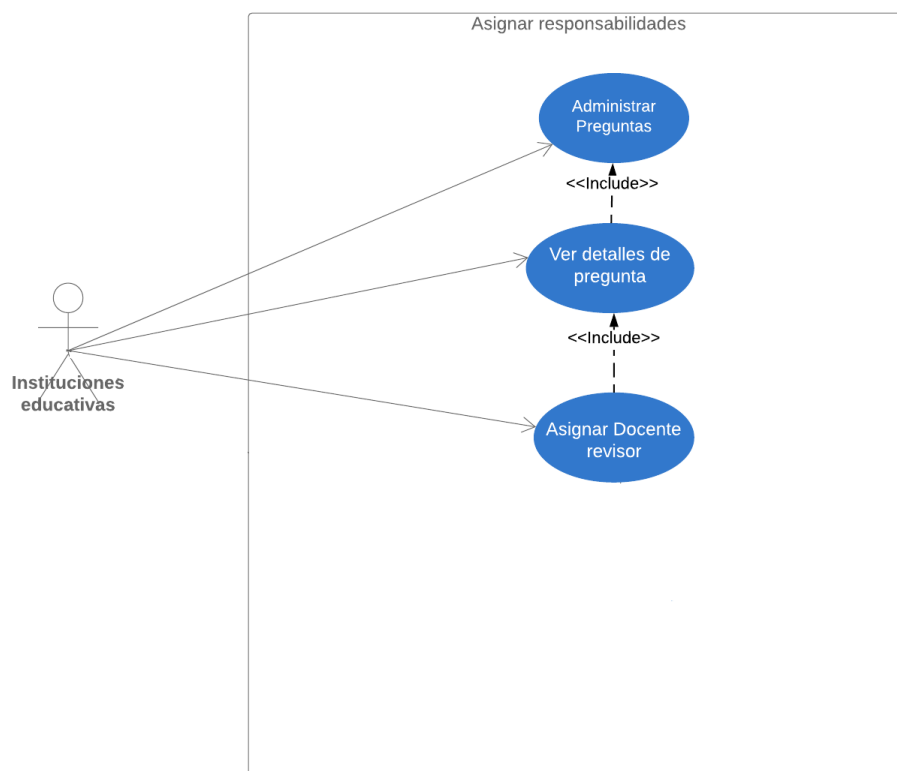
Anexo 44. Diagrama Crear contenido de entrenamiento



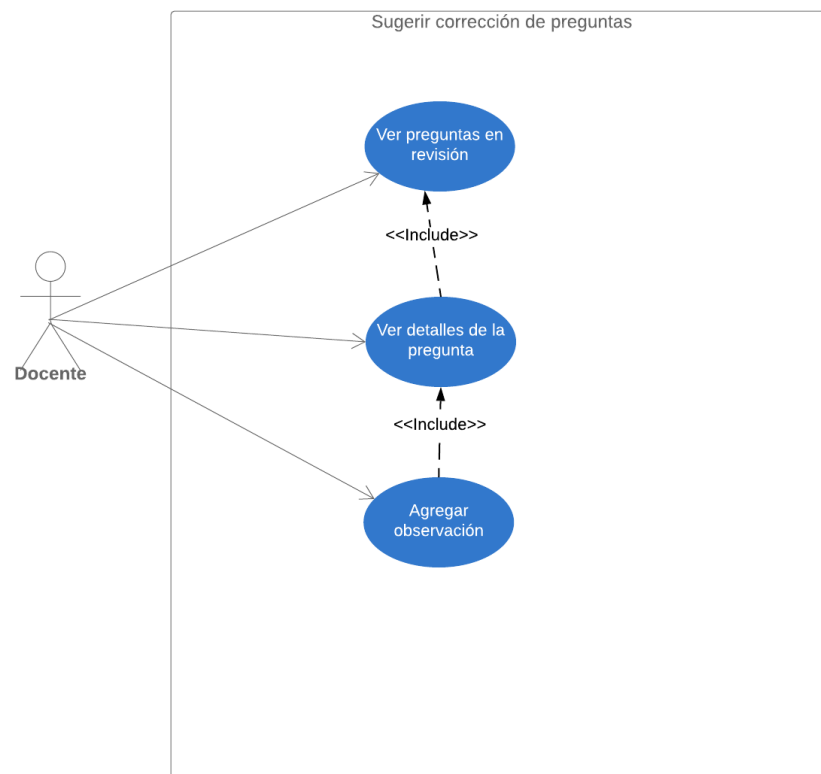
Anexo 45. Diagrama Crear contenido para el banco de preguntas



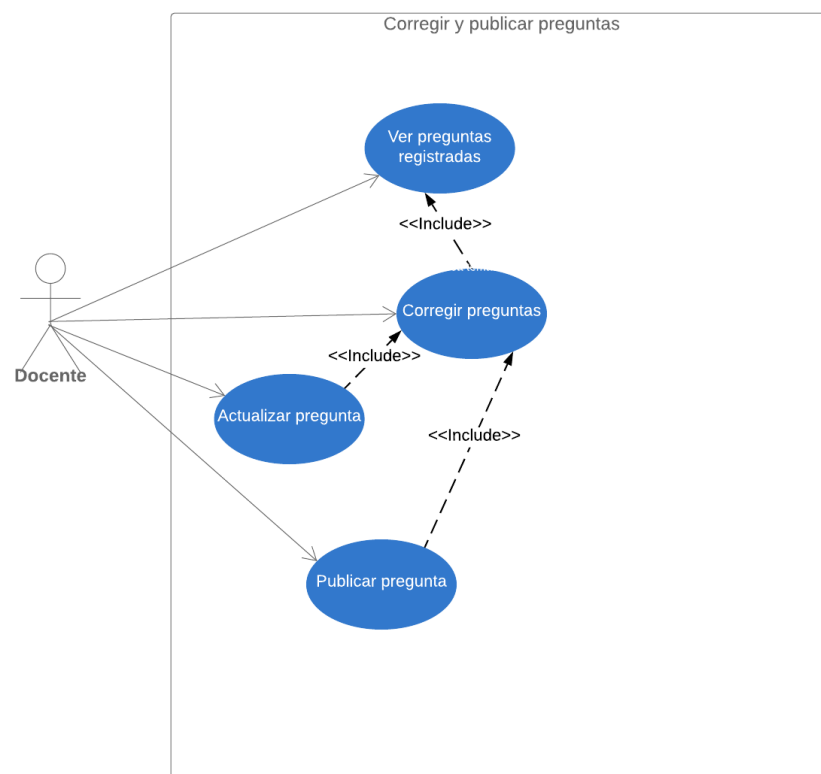
Anexo 46. Diagrama Asignar responsabilidades



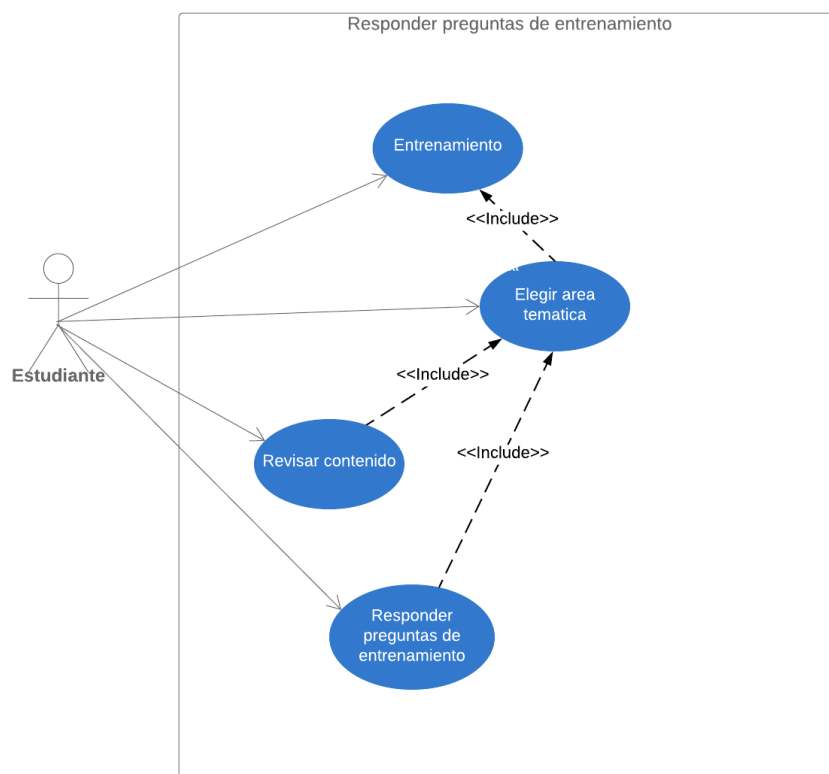
Anexo 47. Diagrama Sugerir corrección de preguntas



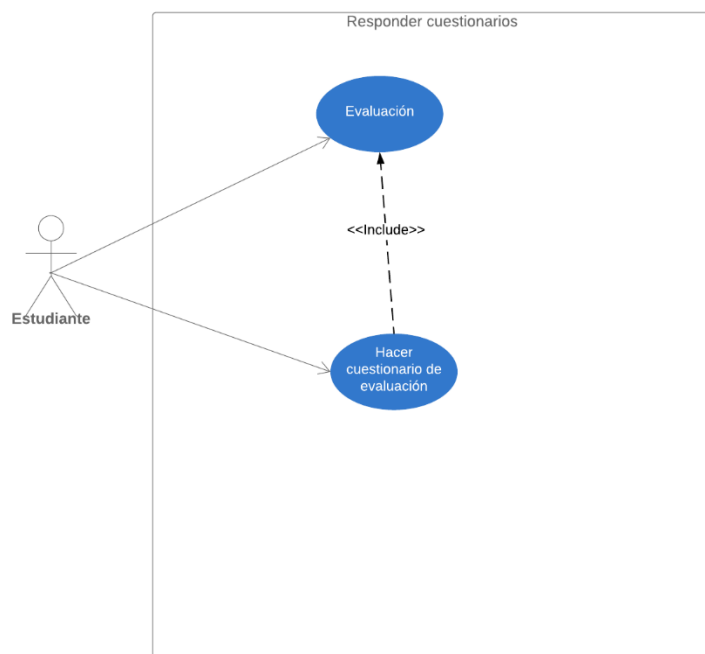
Anexo 48. Diagrama Corregir y publicar preguntas

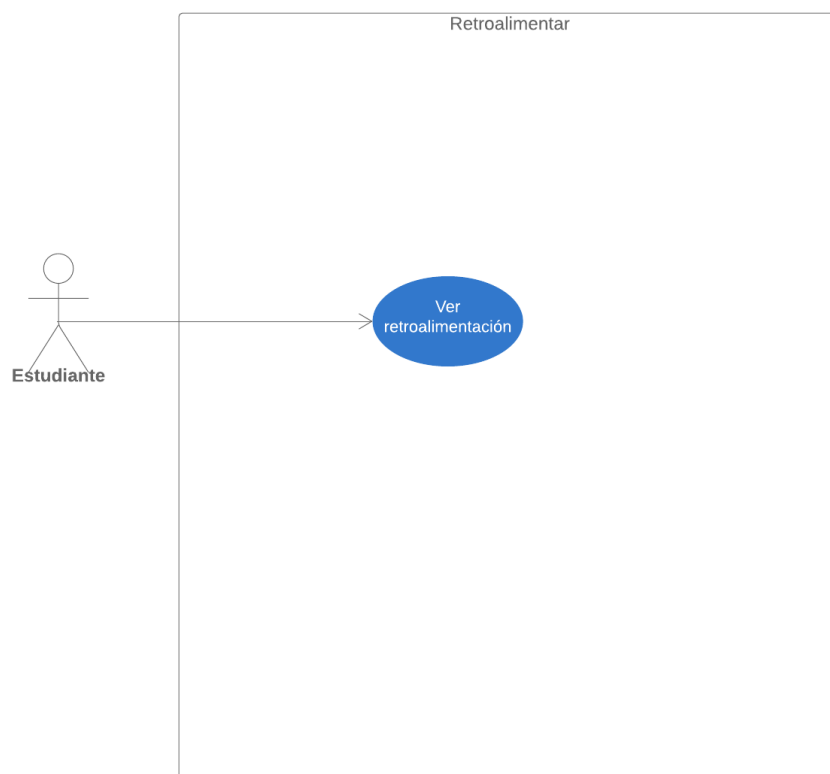


Anexo 49. Diagrama Responder preguntas de entrenamiento



Anexo 50. Diagrama Responder cuestionarios



Anexo 51. Diagrama Retroalimentar

D. Anexo: Estado del arte

Anexo 52. Aspectos funcionales

Evaluación de Software Educativos			
Tipo de Software: Entrenamiento			
Características: Hipertexto - Multimedia			
Aspectos Funcionales:			
	ESE	Grupo Geard	Amir
1. Eficacia didáctica: el programa debe facilitar el logro de los objetivos	No	Si	Si
2. Relevancia e interés de los contenidos y servicios: su valor será mayor cuanto más relevantes sean los objetivos, y más interesantes los contenidos y servicios que ofr	No	Si	No
3. Facilidad de uso:			
ü deben resultar agradables, fáciles y autoexplicativos	No	Si	Si
ü deben ofrecer mapas para saber en todo momento donde se está, y ofrecer sistemas de ayuda.	No	No	No
4. Facilidad de instalación y acceso a la aplicación (web y con un solo repositorio d información)	Si	Si	Si
5. Versatilidad didáctica: para poder responder a las distintas necesidades educativas, los programas deben poder adaptarse a diversos:			
a) entornos de uso: aula de informática, clase, casa...			
Para llevar a cabo esta adaptación, los programas deben			
ü incluir sistemas de evaluación y seguimiento	Si	Si	Si
ü promover actividades complementarias	No	Si	Si
6. Múltiples enlaces externos (en nuestro caso a videos de youtube)	Si	Si	Si
7. Funcionalidad de la documentación o guía de uso: incluyendo manual de usuario y guía didáctica	No	No	No
8. Créditos: fecha de actualización, autores, patrocinadores	Si	Si	Si
9. Ausencia de publicidad	Si	Si	Si

Anexo 53. Aspectos Técnicos

Evaluación de Software Educativos			
Tipo de Software: Entrenamiento			
Características: Hipertexto - Multimedia			
Aspectos Técnicos			
	ESE	Grupo Geard	Amir
1. Calidad del entorno audiovisual			
ü presentación atractiva y correcta (indicará la resolución óptima)	No	Si	No
ü diseño claro y atractivo de las pantallas	No	Si	No
ü calidad técnica y estética de los elementos	Si	Si	Si
2. Calidad y cantidad de elementos multimedia	No	Si	No
3. Calidad y estructura de los contenidos (bases de datos):	No	Si	No
ü información correcta en extensión y rigor, y actual	Si	Si	Si
ü buena estructura	No	Si	No
ü fragmentación adecuada	Si	Si	Si
ü textos bien contruidos gramatical, sintáctica y ortográficamente	Si	Si	Si
ü ausencia de discriminaciones y de mensajes negativos o tendenciosos	Si	Si	Si
4. Buen sistema de navegación por las actividades y mapa de navegación	Si	Si	Si
5. Hipertextos: nivel adecuado (no más de 3 niveles)	Si	Si	Si
6. Interacción:			
ü uso transparente del teclado (que se puedan corregir errores)	Si	Si	Si
7. Ejecución fiable, velocidad y visualización adecuadas	Si	Si	Si
8. Originalidad y uso de tecnología avanzada	No	Si	No

Anexo 54. Aspectos Pedagógicos

Evaluación de Software Educativos			
Tipo de Software: Entrenamiento			
Características: Hipertexto - Multimedia			
Aspectos Pedagógicos:			
	Ese	Grupo Geard	Amir
1. Especificación de los objetivos	No	no	No
2. Capacidad de motivación y atractivo	No	si	si
3. Adecuación a los destinatarios: en cuanto a			
ü contenidos (del tema a desarrollar)	Si	Si	Si
ü actividades (propuestas para entrenamiento)	No	Si	Si
ü servicios de apoyo (para resolver las actividades de entrenamiento)	No	no	Si
4. Adaptación a los usuarios y su ritmo de trabajo: Pueden seleccionar los contenidos en cualquier orden y según s	Si	Si	Si
5. Potencialidad de los recursos didácticos:			
ü organizadores previos (muestra temas que se necesiten para resolver los ejercicios y/o la prueba)	No	Si	No
ü preguntas y ejercicios de relación de conocimientos	Si	Si	Si
ü adecuada integración de medias (que no sean ni pocos ni demasiados, en cantidad)	Si	Si	No
6. Fomento de la iniciativa y el autoaprendizaje:			
ü proporcionar herramientas cognitivas para que el alumno decida las tareas, formas, y nivel, y autocontrole su tra	No	Si	No
ü facilitar el aprendizaje a partir de los errores (para nosotros en la ejercitación, mirar si se tiene, retroalimentación)	No	Si	Si

BIBLIOGRAFÍA

- [1] “Patrón Modelo-Vista-Controlador. | Díaz González | Revista Telemática.” [Online]. Available: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>. [Accessed: 05-Mar-2020].
- [2] E. Bascon, “El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing.”
- [3] R. J. Vargas, D. Valle, and J. P. Maltés Granados, “Programación en Capas.”
- [4] P. Jesus Demetrio Velázquez Camacho, “Desarrollo en Cascada (Waterfall) VS Desarrollo Agile-SCRUM.”
- [5] oliver perez, “Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM.”
- [6] EServiciosEducativos, “Saber Pro (ECAES).” [Online]. Available: <https://eservicioseducativos.com.co/inicio/9-saber-pro-ese.html>. [Accessed: 08-Feb-2020].
- [7] “Pruebas ICFES Saber Pro - Grupo Geard Colombia.” [Online]. Available: <https://grupogear.com/co/pruebas-saber/icfes-saber-pro/>. [Accessed: 10-Feb-2020].
- [8] “Saber PRO – Amir Colombia.” [Online]. Available: <https://www.amircolombia.com/cursos/saber-pro/>. [Accessed: 10-Feb-2020].
- [9] A. Bianchini, “Conceptos y definiciones de hipertexto,” Caracas , 1999.
- [10] U. Del Valle, “INTRODUCCIÓN A LA TECNOLOGÍA INFORMÁTICA Software Colaborativo.”
- [11] J. Cervantes OJeda María CarMen Gómez Fuentes, “Taxonomía de los modelos y metodologías de desarrollo de software más utilizados Resumen,” vol. 52, pp. 37–47, 2012.
- [12] Á. Fernández-López, M. J. Rodríguez-Fórtiz, M. L. Rodríguez-Almendros,

- and M. J. Martínez-Segura, “Guía de arquitectura en N capas orientadas al dominio con Net 4.0,” *Comput. Educ.*, vol. 61, pp. 77–90, Feb. 2013, doi: 10.1016/j.compedu.2012.09.014.
- [13] R. Presuman, “Ing Soft Pressman Quinta Ed | Jorge Barrantes Tejada - Academia.edu.” [Online]. Available: https://www.academia.edu/7365760/Ing_Soft_Pressman_Quinta_Ed. [Accessed: 03-Mar-2020].
- [14] S. Domingo, M. Henríquez, H. Vega Huerta, and L. Guerra Grados, “Programación en N capas.”
- [15] “Stack Overflow Developer Survey 2019.” [Online]. Available: <https://insights.stackoverflow.com/survey/2019>. [Accessed: 06-Dec-2019].