



## Facultad de Ciencias y Tecnologías Computacionales

# Sistema para gestionar los indicadores de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia

*Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas*

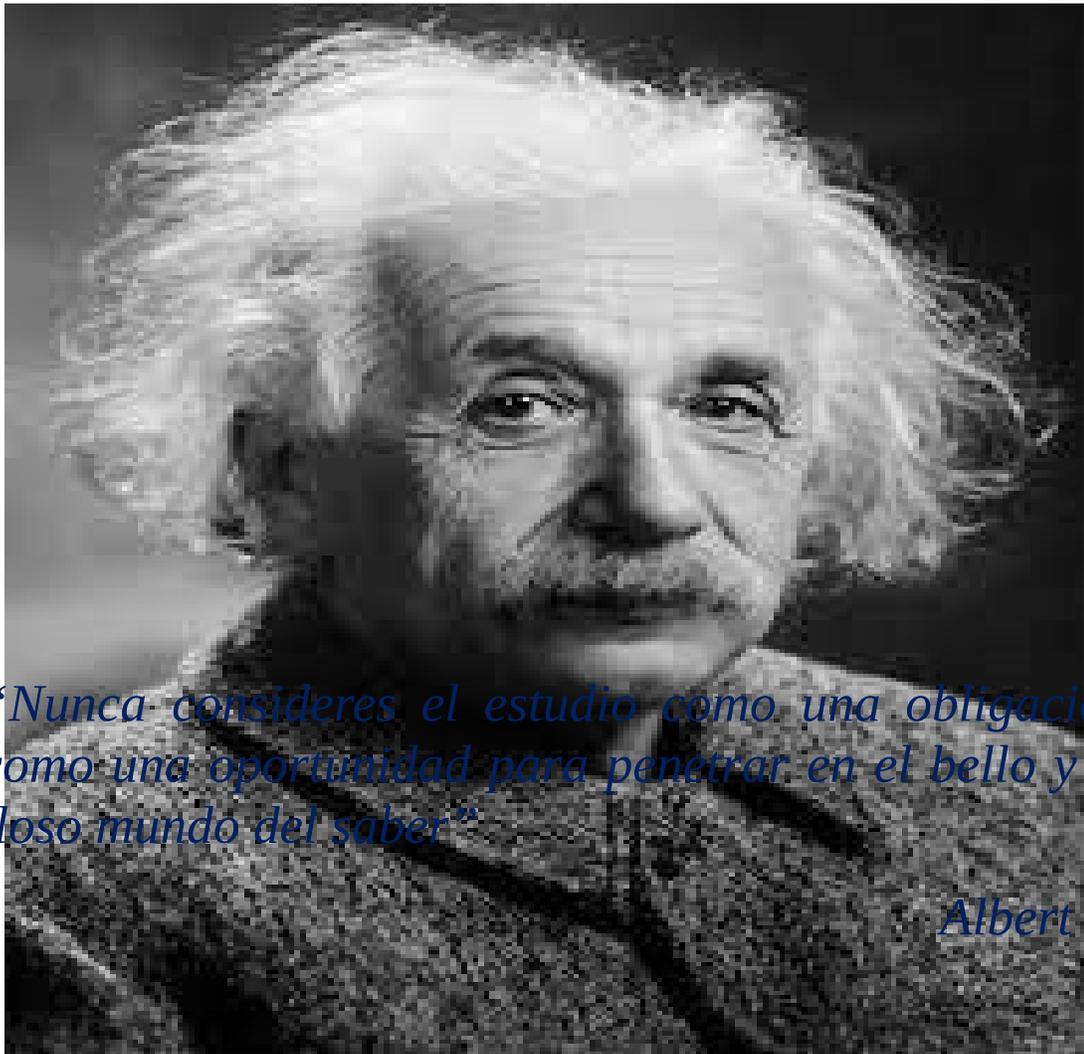
**Autores:** Adachely Miyares Magaña  
Jesús Alarcón Aguilar

**Tutores:** MSc. Porfilio Lorenzo Puig Estrada  
Inst. Ing. Reinier Fernández Coello

**Cotutor:** Ing. Yurier Jesús Herrera Lugo

La Habana, diciembre de 2023

“Año 65 de la Revolución.”



*“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”*

*Albert Einstein*

## DECLARACIÓN DE AUTORÍA

Declaramos por este medio que Adachely Miyares Magaña y Jesús Alarcón Aguilar, somos los autores del Trabajo de Diploma: “**Sistema para gestionar los indicadores de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia**” y que autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo. Y para que así conste, firmamos la presente declaración de autoría en La Habana a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2023.

---

**Firma del Autor**

Adachely Miyares Magaña

---

**Firma del Autor**

Jesús Alarcón Aguilar

---

**Firma del Tutor**

Inst. Ing. Reinier Fernández Coello

---

**Firma del Tutor**

MSc. Porfilio Lorenzo Puig Estrada

---

**Firma del Cotutor**

Ing. Yurier Jesús Herrera Lugo

## AGRADECIMIENTOS

*Mi más sincero agradecimiento a mis padres, Josefa y Andrés por su amor y sacrificio. Gracias por inculcarme valores sólidos y por ser un ejemplo de perseverancia y dedicación en la vida familiar.*

*A mi pareja, Yonder, por su apoyo incondicional durante todo el proceso de investigación y redacción de esta tesis. Gracias por comprender mis largas horas de trabajo y por brindarme tu amor y aliento en cada etapa.*

*A mi familia, en específico a mis tíos Deunis y Adis Yeline por su amor incondicional y por creer en mí en todo momento. Gracias a ustedes, he tenido el respaldo emocional y la motivación necesaria para enfrentar los desafíos que este proyecto implicaba.*

*A mis hermanas, Arasay y Anne, por su apoyo inquebrantable y por ser mi fuente constante de alegría y motivación. Han sido mi ejemplo a seguir y siempre han estado ahí para celebrar mis éxitos y levantarme en momentos de dificultad.*

*No puedo dejar de agradecer a mis tutores de tesis, Reinier Fernández Coello y Yurier Jesús Herrera, por su inestimable guía, apoyo y dedicación a lo largo de todo el proceso de investigación y redacción de esta tesis. Su experiencia, conocimiento y orientación han sido fundamentales para el éxito de este trabajo. Agradezco sinceramente su tiempo, paciencia y disposición para responder a mis preguntas y dudas, así como por sus valiosos comentarios y sugerencias que han contribuido a mejorar significativamente este estudio. Gracias por brindarme su tiempo, por sus consejos y por motivarme constantemente a superar los desafíos que se presentaron en el camino.*

*Quiero expresar mi más profundo agradecimiento al tribunal que evaluó y examinó mi tesis. Sus valiosas aportaciones, comentarios y críticas constructivas fueron fundamentales para el desarrollo y mejora de mi trabajo*

*También quiero extender mi gratitud a mi oponente, cuya participación en el proceso de*

*evaluación de mi tesis fue de gran importancia. Sus interrogantes y observaciones me permitieron reflexionar sobre mi investigación desde diferentes perspectivas y enriquecieron mi trabajo de manera significativa. Agradezco su disposición para debatir y discutir los aspectos relevantes de mi investigación.*

*Agradezco a mis amigos cercanos Leandro, Jose Orlando, Heiser, Anais, Daimi por su apoyo emocional y por ser una fuente constante de motivación. Sus palabras de aliento y su disposición a escucharme en momentos de duda fueron de gran valor para mí.*

*A mis amigos de la universidad, Fernando, Ernesto, Diana, Daymara, Dayelin, Daniela, Glenda, Danilo por compartir conmigo este camino académico y por ser una fuente de apoyo mutuo. Juntos hemos enfrentado desafíos, superado obstáculos y celebrado nuestros logros.*

*Agradezco especialmente a mi amiga Vivi, por su asesoramiento y orientación en momentos clave de mi investigación. Su experiencia y sabiduría ha sido preciso para el desarrollo de mi tesis.*

*Agradezco a Erasmo Vinent, quien, aunque ya no esté presente, su legado perdurará en cada logro que alcanzo. Se que donde quiera que te encuentres, estarás orgulloso de mi. A mis profesores en especial, Leo y Alejandro, gracias por su dedicación, paciencia y sabiduría a lo largo de la carrera. Su apoyo y conocimientos han sido incalculable para el desarrollo de esta tesis.*

*A los familiares de mi pareja, quienes también se han convertido en mi familia. Su amor, apoyo y comprensión han sido una fuente constante de fortaleza y motivación. Agradezco, por abrirme sus corazones y brindarme su confianza.*

*A mi compañero de tesis, Jesús, quien ha sido mi aliado y cómplice a lo largo de este desafiante proceso. Tú dedicación y trabajo en equipo han sido fundamentales para llegar hasta aquí.*

*Agradezco especialmente a Kike, Yusleivis, Joel, Calvin, Yoe, quienes han sido una*

*fuente constante de apoyo y motivación. Sus consejos, palabras de aliento y experiencias compartidas han sido fundamentales para mi crecimiento académico y personal.*

*A mis vecinos, quienes han sido más que simples vecinos, se han convertido en amigos y una extensión de mi familia, en especial a Tatica, Gladys, Virginia, Yinet, Maribel.*

*Gracias a todos por formar parte de esta travesía y ayudarme a convertir esta tesis en una realidad. Su apoyo ha sido preciado y siempre los llevaré en mi corazón.*

***Adachely***

Quiero expresar mi más sincero agradecimiento a todas las personas que han sido parte de mi camino hacia el éxito académico. En primer lugar, agradezco a mi familia cercana, quienes han sido un pilar fundamental en mi vida y especialmente en mi carrera universitaria. Agradezco de manera especial a mi madre Caridad, mi abuela Dilcia, mi tía Enma, mi tío Gilberto, mi tío Miguel, mi primo Daniel, mi padre Luis y mi hermana Katerine. Su apoyo incondicional, amor y aliento han sido cruciales en mi camino hacia el éxito académico. También quiero mencionar a otros familiares como mi tía Norma, tía Rosa María, tío Braulio, Anet y Yamir, quienes me han brindado su apoyo y cariño en todo momento.

En segundo lugar, quiero agradecer a mis tutores Reinier Fernández Coello y Yurier Jesús Herrera. Su dedicación, paciencia y conocimientos han sido fundamentales en mi desarrollo académico. Gracias a su guía constante, he logrado superar obstáculos y alcanzar mis metas. Su apoyo incondicional me ha inspirado a esforzarme cada día más.

En tercer lugar, quiero expresar mi profundo agradecimiento a mis amigos que a pesar de no estar presentes todo el tiempo me han apoyado. A Oslieki, Leandro, Niuman, Alejandro, Felix, Enmanuel, Yoan, Dennier, Melisa, Lisbet, Keylis, Irela y Liubis. A pesar de la distancia, su amistad y apoyo han sido inestimable en mi vida.

Quiero también mencionar a mis amigos de la universidad: Alain, Alex, Lázaro Jesús, Luis Gabriel, Iván, Noelvis, Irael, Alejandro Góngora, Yariel Roig, Brian, Aleyma, Lus Clara, Yusnavi Yi, Arianna, Aranay, Darién Armenteros, Lázaro Bladir, Sergio, José Ramón, Katya, Leandro, Leyanet, Raysel, Oscar, Ernesto, Gabriel, Ushire, Yanela y mi amiga y compañera de tesis Adachely. Su compañía y apoyo ha sido fundamental.

Por último, quiero expresar mi más sincero agradecimiento a los profesores Alieski, Barbán, Fabra, Yadira, Yordanis, Neysis, Lieen, Jonathan, Yoan y Dariela. Agradezco también a la decana Ana Marys y la vicedecana Maidelis, así como la instructora

*Niurbis y algunas de las tías del comedor que han aportado para atravesar las dificultades. Gracias por su dedicación y compromiso con la educación.*

*A todas estas personas les agradezco por su presencia en mi vida y por haber sido parte fundamental en mi camino hacia el éxito académico. ¡Gracias por creer en mí y por su apoyo incondicional!*

***Jesús***

## DEDICATORIA

*A mis padres, cuyo amor incondicional y sacrificio han sido la fuerza impulsora detrás de mis logros.*

*A mi amado novio, quien ha sido mi roca y mi fuente inagotable de apoyo emocional durante este proceso.*

*A mis queridos Reinier Fernández Coello, Yurier Jesús Herrera, cuya orientación experta y paciencia infinita han sido fundamentales en la realización de este proyecto.*

*A todos los profesores y mentores que han dejado una huella indeleble en mi formación académica.*

### ***Con gratitud y cariño, Adachely.***

*A mi familia por su constante apoyo, amor y comprensión a lo largo de toda mi carrera universitaria. Desde el inicio hasta el final, han estado a mi lado, brindándome aliento y motivación en cada paso del camino.*

*A todas las personas que, de una u otra manera, han aportado un poquito de su tiempo, apoyo y confianza en mí. Cada uno de los profesores que me alentaron diciéndome "si puedes lograrlo" cuando había perdido la fe en mí mismo. Su confianza en mis habilidades ha sido un impulso invaluable para superar los desafíos y alcanzar mis metas.*

*A cada una de estas personas, ¡muchas gracias por estar a mi lado a lo largo de mi carrera universitaria y por ser parte esencial de este logro!"*

***Jesús.***

## RESUMEN

Hoy en día Cuba adopta una nueva estrategia en donde ha ido desplazando la idea de que solo se puede aprender en un aula de clases y frente a un docente. Como parte de esta estrategia, el Centro Nacional de Educación a Distancia citado en la Universidad de Ciencias Informáticas, ofrece varios cursos virtuales como apoyo al desarrollo y la excelencia de la educación a distancia y semipresencial en el país. Estos cursos son evaluados y revisados por profesores con experiencias, donde se emite un informe por cada revisión. Dicho proceso se realiza de forma manual y con gran dependencia del papel, lo que constituye un problema dado al volumen de indicadores a evaluar. Lo anteriormente descrito trae consigo lentitud y demora en el proceso, y en algunos de los casos, que no se cumpla con la planificación realizada. Por ello, la presente investigación tiene como objetivo principal el desarrollo del Sistema para gestionar los indicadores de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia, de forma tal que se contribuya a la celeridad de dicho proceso. Para lograr este objetivo, se ha utilizado como metodología de desarrollo: Proceso Unificado Ágil, siguiendo la versión establecida por la propia universidad. Además, se ha empleado un conjunto de tecnologías en correspondencia a la aspiración de soberanía tecnológica. Los resultados obtenidos fueron validados a través de métricas y pruebas de software, que han permitido comprobar la calidad de las funcionalidades internas y externas del sistema propuesto.

**PALABRAS CLAVES:** calidad, celeridad, curso virtual, educación a distancia.

**ABSTRACT****ABSTRACT**

Nowadays, Cuba is adopting a new strategy where it has been displacing the idea that learning can only take place in a classroom and in front of a teacher. As part of this strategy, the National Center for Distance Education, cited at the University of Computer Sciences, offers various virtual courses to support the development and excellence of distance and blended education in the country. These courses are evaluated and reviewed by experienced professors, and a report is issued for each review. This process is done manually and relies heavily on paper, which poses a problem given the volume of indicators to evaluate. The aforementioned brings about slowness and delays in the process, and in some cases, the planned schedule is not met. Therefore, the main objective of this research is the development of a System to manage the quality indicators of the courses in the virtual classroom of the National Center for Distance Education, in such a way that it contributes to the speed of this process. To achieve this objective, the Agile Unified Process has been used as the development methodology, following the version established by the university itself. In addition, a set of technologies has been used in accordance with the aspiration of technological sovereignty. The obtained results were validated through metrics and software testing, which have allowed verifying the quality of the internal and external functionalities of the proposed system.

**KEYWORDS:** Quality, speed, virtual course, distance education

|  |      |
|--|------|
| RESUMEN.....   | IX   |
| ABSTRACT.....  | X    |
| ÍNDICE.....  | XI   |
| ÍNDICE DE TABLAS.....  | XIII |
| ÍNDICE DE FIGURAS.....   | XIV  |
| INTRODUCCIÓN.....  | 1    |
| CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO..... | 7    |
| I.1 Fundamentos teóricos-metodológicos asociados al objeto de estudio.....                 | 7    |
| I.2 Análisis de sistemas homólogos.....  | 8    |
| I.3 Metodología de desarrollo.....   | 9    |
| I.4 Herramientas y lenguaje modelado.....  | 11   |
| I.4.1 Lenguajes de desarrollo.....   | 12   |
| I.4.2 Herramienta de prototipado.....  | 14   |
| I.4.3 Bibliotecas y marcos de trabajo para el desarrollo.....                              | 15   |
| I.4.4 Entorno de desarrollo.....   | 16   |
| I.4.5 Sistema gestor de base de datos.....   | 17   |
| I.4.6 Servidor web.....  | 17   |
| I.4.7 Control de versiones.....  | 18   |
| CAPÍTULO II: DISEÑO PROPUESTA AL PROBLEMA CIENTÍFICO DE LA SOLUCIÓN.....                   | 20   |
| II.1 Descripción de la propuesta de solución.....  | 20   |
| II.2 Requisito de software.....  | 21   |
| II.2.1 Requisitos funcionales (RF).....  | 21   |
| II.2.2 Requisitos no funcionales (RnF).....  | 27   |
| II.3 Historias de usuario.....   | 29   |
| <b>Requisito:</b> Revisar curso.....   | 29   |
| <b>Requisito:</b> Registrar curso.....   | 30   |
| II.4 Arquitectura del sistema.....   | 31   |

|  |    |
|--|----|
| II.4.1 Patrón arquitectónico.....  | 31 |
| II.5 Diagrama de clases del diseño.....  | 33 |
| II.6 Patrones de diseño.....   | 34 |
| II.6.1 Patrones Generales de Software para Asignación de Responsabilidades.....                        | 35 |
| II.6.2 Patrones GoF.....   | 36 |
| II.7 Modelo de datos.....  | 37 |
| II.8 Diagrama de componentes.....  | 38 |
| II.9 Diagrama de despliegue.....   | 39 |
| Conclusiones del capítulo.....   | 40 |
| CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....   | 42 |
| III.1 Técnicas de validación de requisitos.....  | 42 |
| III.2 Revisión técnica formal.....   | 42 |
| III.3 Validación del diseño.....   | 45 |
| III.3.1 Métricas Tamaño Operacional de las Clases.....   | 45 |
| III.3.2 Métricas Relación entre Clases.....  | 48 |
| III.3.4 Estrategia de prueba.....  | 51 |
| III.3.4.1 Pruebas unitarias.....   | 52 |
| III.3.4.2 Pruebas funcionales.....   | 56 |
| III.4.3 Pruebas de usabilidad.....   | 59 |
| III.3.7 Validación de las variables de la investigación.....   | 63 |
| III.3.7 Conclusiones del capítulo En este capítulo se examinó la propuesta de solución,<br>donde:..... | 65 |
| CONCLUSIONES FINALES.....  | 66 |
| RECOMENDACIONES.....   | 67 |
| REFERENCIAS.....   | 68 |
| ANEXOS.....  | 75 |

ÍNDICE DE TABLAS

**Tabla 1.** Comparación de los sistemas homólogos estudiados.....9  
**Tabla 2.**Listado de Requisitos Funcionales del sistema.....21  
**Tabla 3.**HU\_14 Revisar Curso.....29  
**Tabla 4.**HU\_14 Registrar Curso.....30  
**Tabla 5.**Métrica TOC. Categoría por atributos y criterio de evaluación.....46  
**Tabla 6.** Métrica RC. Categoría por atributos y criterio de evaluación.....49  
**Tabla 7.** Validación de las variables de investigación.....62

ÍNDICE DE FIGURAS

**Figura 1.** Funcionamiento del patrón arquitectónico Modelo-Vista-Plantilla.....32  
**Figura 2.** Arquitectura del proyecto.....33  
**Figura 3.** Diagrama de clases del diseño Registrar Curso.....34  
**Figura 4.** Modelo de datos.....38  
**Figura 5.** Diagrama de componentes correspondiente a la HU\_11 Registrar curso.....39  
**Figura 6.** Diagrama de despliegue.....40  
**Figura 7.**Representación de la cantidad de clases por cantidad de procedimientos.....47  
**Figura 8.**Resultados de la métrica TOC.....47  
**Figura 9.**Representación de la cantidad de clases por cantidad de dependencias.....50  
**Figura 10.**Resultados de la métrica RC.....50  
**Figura 11.** Método utilizado como ejemplo para la técnica de ruta básica.....53  
**Figura 12.** Grafo del flujo del método Registrar Revisión.....54  
**Figura 13.**Total de NC detectadas por iteración.....58  
**Figura 14**Total, de NC por tipo de clasificación.....58



La incorporación y utilización de las Tecnologías de la Información y la Comunicación (TIC) en el ámbito de la educación ha dado paso a nuevas formas de aprender y de enseñar, provocando profundas transformaciones y demandando que la educación se adapte las continuas modificaciones que se producen. La utilización de las TIC en la educación contribuye a la modernización de los procesos de enseñanza y aprendizaje, al desarrollo de competencias y de habilidades, y trae consigo grandes retos para docentes, estudiantes e instituciones educativas[ CITATION Her21 \l 3082 ].

A medida que las TIC han avanzado, la educación también ha experimentado una evolución acelerada que ha favorecido a que la educación a distancia se convierta en una modalidad de estudio muy utilizada en la actualidad. Esta se caracteriza por la separación del profesor y los estudiantes en tiempo y espacio, así como del aprovechamiento de las oportunidades que ofrecen los medios tecnológicos para el desarrollo del proceso de enseñanza y aprendizaje. De ahí que ha evolucionado y han surgido los cursos en línea o virtuales, masivos y abiertos[ CITATION Arl21 \l 3082 ].

Promover la educación a distancia en Cuba es uno de los objetivos que el Ministerio de Educación Superior y el Centro Nacional de Educación a Distancia se han planteado en los últimos años. Esto se debe ya que, gracias al avance de las TIC, se crean nuevas se crean nuevas posibilidades en el proceso de enseñanza-aprendizaje. Ejemplo de ello es el acceso a una gran cantidad de información y recursos educativos digitales desde cualquier lugar, lo que significa que los estudiantes pueden acceder a la educación desde cualquier lugar del mundo con conexión a internet. De igual modo se ha dado paso a la existencia de herramientas y recursos que facilitan la investigación y la colaboración. Por otro lado, permiten a los estudiantes y profesores comunicarse y colaborar más fácilmente, ya sea a través de herramientas de comunicación en línea, foros de discusión o videoconferencias[CITATION red20 \l 3082 ].

Con el Modelo de Educación a Distancia de la Educación Superior Cubana, se considera que: la educación a distancia es una modalidad educativa en la que el proceso de enseñanza aprendizaje se caracteriza por la separación del profesor y el estudiante en tiempo y espacio, se apoya en diferentes recursos educativos que propician y estimulan el aprendizaje autónomo del estudiante[ CITATION Lóp17 \l 3082 ].

En 2014, se creó el Centro Nacional de Educación a Distancia (CENED), el cual contribuye al desarrollo y la excelencia de la educación a distancia y semipresencial en Cuba. Con el objetivo de desarrollar programas de enseñanza en la modalidad virtual, así como al diseño y evaluación de

cursos y recursos educativos de apoyo a ese tipo de formación. Como apoyo a la educación a distancia se diseñó el Modelo de Educación a Distancia de la Educación Superior Cubana (2016); el cual plantea que se debe potenciar la flexibilidad, la interacción y la comunicación, la convergencia e integración tecnológica. El mismo consta de cuatro componentes estrechamente relacionados entre sí: el componente recursos humanos, el componente pedagógico, el componente tecnológico y el componente organizativo.

El Centro Nacional de Educación a Distancia (CENED), con sede en la Universidad de las Ciencias Informáticas tiene como misión contribuir al desarrollo y la excelencia de la educación a distancia y semipresencial en Cuba; orientar, dirigir y controlar el trabajo metodológico en esta modalidad y el desarrollo y aplicación de las tecnologías como medio de apoyo, además de coordinar la formación de los docentes que trabajan en la educación a distancia. Actualmente, la revisión de los cursos virtuales que se ofrecen como apoyo al desarrollo y la excelencia de la educación a distancia y semipresencial en Cuba, se realiza de forma manual. Esta revisión se realiza teniendo en cuenta las pautas establecidas por el CENED para el diseño de cursos a distancia, los indicadores para la revisión de la calidad de los cursos en entornos virtuales de aprendizaje y los elementos asociados al Modelo de Educación a Distancia de la Educación Superior Cubana.

Estos indicadores se encuentran agrupados por cuatro categorías para un total de cuarenta y dos indicadores, y se evalúan a través de un Excel del paquete de Office, es válido aclarar que esta cifra puede variar según las pautas establecidas por el CENED. Esto trae consigo además del evidente gasto de tiempo de cada uno de los revisores y que el análisis del resultado de estas revisiones no sea completamente eficiente en cada una de sus etapas, debido al cúmulo de información que se debe procesar. Esto provoca que el proceso de revisión de los cursos disponibles en el aula virtual se vuelva lento y que requiera de mucho tiempo y esfuerzo.

Ante la problemática anteriormente planteada surge el siguiente **problema a resolver**: ¿Cómo gestionar los indicadores de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia de forma tal que se contribuya a la celeridad de dicho proceso?

Con el fin de dar solución al problema propuesto para el presente trabajo, se delimita como **objeto de estudio** el proceso de desarrollo de sistemas web de gestión, delimitándose como **campo de acción** la gestión de indicadores de calidad de los cursos virtuales disponibles en el aula virtual del Centro Nacional de Educación a Distancia.

Para dar solución al problema planteado, se define como **objetivo general**: desarrollar el Sistema para gestionar los indicadores de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia, de forma tal que se contribuya a la celeridad de dicho proceso.

Por lo tanto, se plantea la siguiente **idea a defender**: si se desarrolla el Sistema para gestionar los indicadores de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia, se contribuye a la celeridad de dicho proceso.

Con el fin de darle solución al objetivo general anteriormente planteado se trazaron los siguientes **objetivos específicos**:

1. Establecer los referentes teóricos en los que se sustenta la propuesta de solución sobre el desarrollo de sistemas web de gestión dirigidos a la gestión de indicadores de calidad de cursos virtuales.
2. Realizar la identificación de requisitos, análisis y diseño del sistema que se propone para gestionar los indicadores de calidad de los cursos del aula virtual del CENED, como aproximación a la implementación.
3. Implementar el Sistema para gestionar los indicadores de calidad de los cursos del aula virtual del CENED.
4. Validar el correcto funcionamiento del sistema propuesto aplicando métricas y pruebas de software para garantizar la calidad del mismo, así como las variables dependientes e independientes de la presente investigación.

Para llevar a cabo la presente investigación se tuvo en cuenta la utilización de varios **métodos científicos de investigación**, clasificados en teóricos y empíricos, tales como:

#### **Métodos del nivel teórico:**

- **Histórico-Lógico**: el método histórico caracteriza al objeto en sus aspectos más externos, a través de la evolución y desarrollo histórico del mismo; y el método lógico reproduce en el plano teórico la esencia del objeto de estudio, investigando las leyes generales y primordiales de su funcionamiento y desarrollo [ CITATION Pér17 \l 3082 ]. El empleo de este método permitió sentar las bases sobre el estudio de sistemas relacionados con la gestión de los indicadores utilizados al evaluar la calidad de los cursos virtuales. Además, permitió adquirir conocimientos

de la lógica objetiva del desarrollo respecto a cómo se realiza la revisión de calidad y de los indicadores involucrados en el proceso en el CENED.

- **Análisis-Síntesis:** el análisis permite la división mental del fenómeno en sus múltiples relaciones y la síntesis establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas [ CITATION Pér17 \l 3082 ]. La utilización de este método permitió realizar el análisis teórico e identificar los principales conceptos a incluir en la fundamentación teórica, relacionados con la gestión de indicadores de calidad de los cursos virtuales disponibles en el aula virtual del Centro Nacional de Educación a Distancia.
- **Modelación:** permite la representación material de un proceso para entender mejor su origen o funcionamiento [ CITATION Pér17 \l 3082 ]. La utilización de este método permitió realizar los diagramas necesarios en el proceso de desarrollo de software, haciendo una representación abstracta de la solución, facilitando así el desarrollo del sistema que se propone para la gestión de los indicadores de calidad de los cursos virtuales disponibles en el aula virtual del CENED.

#### **Métodos del nivel empírico:**

- **Entrevista:** la entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información [ CITATION Pér17 \l 3082 ]. Se empleó con asesores y miembros calificados del CENED para entender el proceso de negocio, comprender la estructura y funcionamiento de la organización. De igual modo permitió identificar las deficiencias existentes que permitieron definir el problema a resolver y establecer el objeto de estudio. Además, se obtuvieron referencias e información útil para elaborar los requisitos de la solución.
- **Observación:** es un método científico ampliamente utilizado en diferentes disciplinas. es relativamente fácil de llevar a cabo y no requiere de equipos costosos o de habilidades técnicas especializadas [ CITATION Pot17 \l 3082 ]. Posibilitó identificar elementos de importancia a relacionar en la solución propuesta, relacionados con la revisión de los cursos disponibles en el aula virtual del CENED.

El presente documento consta de tres capítulos, conclusiones generales, referencias bibliográficas y anexos. La **estructura capitular** se realiza según el nivel de detalle que requiere el contenido abordado en cada uno de los capítulos, definidos de la siguiente forma:

**Capítulo I. Fundamentos y referentes teórico-metodológicos:** en este capítulo se aborda lo relacionado con la fundamentación teórica con el objetivo de establecer los elementos teóricos para la realización de la investigación. De ahí que, se especifican los conceptos asociados al desarrollo del sistema que se propone para gestionar los indicadores y revisiones de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia. En segundo lugar, se hace un análisis de soluciones similares existentes en el mundo y en Cuba. Seguidamente se define la metodología a seguir y por último se caracterizan las tecnologías y herramientas que serán utilizados durante el desarrollo de dicho sistema.

**Capítulo II. Diseño de la solución propuesta al problema científico:** en este capítulo se aborda lo relacionado a la propuesta de solución, donde se lleva a cabo la disciplina de requisitos, en la cual se engloban los requisitos funcionales y no funcionales. Se obtienen de igual forma los artefactos correspondientes a la disciplina de "Análisis y Diseño" partiendo del patrón arquitectónico que da soporte a la propuesta de solución, De igual forma y aplicando los patrones de diseño definidos como buenas prácticas, se obtienen los diagramas de clase del diseño y el modelo de datos. Por último, se desarrolla la vista de implementación del sistema la cual es representada a partir del diagrama de componentes; y como complemento a este se representa el diagrama de despliegue.

**Capítulo III. Validación de la solución propuesta:** en este capítulo se evalúa el grado de calidad de los resultados obtenidos en este trabajo, donde se valida el diseño aplicando las métricas: Tamaño operacional de las clases y Relaciones entre clases. Además, se aplican pruebas de software para verificar y revelar la calidad del producto del sistema que se propone como solución a la problemática planteada. Por último, se realiza la validación de las variables de la investigación para demostrar que la solución propuesta contribuye a la celeridad del proceso de revisión de los cursos virtuales disponibles en el aula virtual, a partir de los indicadores de calidad definidos por el CENED.

En el presente capítulo se definen conceptos de interés vinculado a la gestión de indicadores y revisiones de calidad de cursos virtuales. De ahí que se parte con la conceptualización de los elementos teóricos bases para la ejecución de la investigación. Seguidamente se muestra un análisis de soluciones similares al sistema que se desea desarrollar, que servirán como punto de partida para la construcción de la solución propuesta. Además, y de acuerdo con las políticas que sigue la UCI para estandarizar el proceso de desarrollo de software, se define la metodología recomendada. Por último, teniendo en cuenta la soberanía tecnológica a la que aspira el país, se caracterizan las herramientas y lenguajes que serán utilizados para obtener los artefactos ingenieriles generados durante el proceso de desarrollo.

### 1.1 Conceptos asociados a la gestión de indicadores de calidad de cursos virtuales

Para un mejor entendimiento de la propuesta, luego de una consulta de la documentación relacionada con el objeto de estudio, los autores de la presente investigación asumen los siguientes conceptos:

**Sistema de gestión:** es una herramienta que permite controlar, planificar, organizar y automatizar las tareas administrativas de una organización. Analiza los rendimientos y los riesgos de una empresa, con el fin de otorgar un ambiente laboral más eficiente y sostenido [CITATION Eva23 \l 3082]. En el contexto de la presente investigación, el término se refiere a una solución capaz de organizar, controlar el proceso de revisión de los indicadores de calidad de los cursos disponibles en el aula virtual, donde juega un papel fundamental el uso del avance de la tecnología.

**Celeridad:** se trata de un término que hace referencia a la velocidad, la premura, la rapidez o la prisa. Desde el punto de vista de la solución el término será tratado como el tiempo de asignación de datos, tiempo para recopilar datos y tiempo de búsqueda [CITATION Por23 \l 3082]. Para el caso de la presente investigación, la celeridad hace referencia a la rapidez o al tiempo empleado para completar la revisión de los cursos virtuales disponibles en el aula virtual del CENED, antes y una vez puesta en producción la solución propuesta.

**Calidad:** término que proviene del latín *qualitas* o *qualitatis* y hace referencia a una herramienta básica e importante para una propiedad inherente de cualquier cosa que permite que la misma sea comparada con cualquier otra de su misma especie [CITATION Oxf23 \l 3082]. La palabra calidad tiene múltiples significados, por ejemplo, se considera como la capacidad que posee un objeto para satisfacer necesidades implícitas o explícitas según un parámetro, un cumplimiento de requisitos de

calidad[ CITATION ISO15 \l 3082 ]. Para el contexto de la evaluación y revisión a la cual están sujetos los cursos disponibles en el aula virtual del CENED, los autores asumen la calidad a la propiedad que posee cada curso al cumplir o no con cada uno de los indicadores definidos para su revisión, con el fin de lograr satisfacer las necesidades implícitas o explícitas deseadas por los profesores.

**Curso virtual:** hace referencia al desarrollo de la dinámica de enseñanza - aprendizaje que es realizado de forma virtual. Es decir, existe un formato educativo en donde los docentes y estudiantes pueden interactuar diferente al espacio presencial[ CITATION Gar19 \l 3082 ].

### I.2 Análisis de sistemas homólogos

Luego del análisis de los principales conceptos asociados al dominio del problema, se realiza un estudio de los sistemas homólogos. A raíz de del estudio realizado en el ámbito foráneo se concluye que no se encontró estudios previos de sistemas de gestión enfocados a la revisión y evaluación de cursos en línea, por tal razón se enfoca el estudio de sistemas homólogos en sistemas internacionales.

1. Sistema de Garantía de Calidad de la Universidad Nacional de Educación a Distancia (UNED): la UNED cuenta con un sistema de garantía de calidad que incluye la revisión y evaluación de los cursos y programas de enseñanza a distancia. El sistema se basa en la elaboración de informes de seguimiento y evaluación de los cursos, que son analizados por un comité de expertos en educación a distancia[ CITATION Une22 \l 3082 ].
2. Sistema de Evaluación de la Calidad de la Educación Superior (SECE): el SECE es un sistema de evaluación de la calidad de la educación superior en España que se encarga de evaluar los programas y cursos de las instituciones de educación superior. Este sistema se basa en la evaluación de las competencias y habilidades de los estudiantes, así como en la calidad del profesorado y los recursos educativos[CITATION Min21 \l 3082 ].
3. Sistema de Acreditación de la Calidad de los Títulos Universitarios (ANECA): ANECA es una agencia de evaluación de la calidad de la educación superior en España. Esta agencia se encarga de acreditar los títulos universitarios y los programas de enseñanza a distancia, y para ello utiliza una metodología basada en la evaluación de los planes de estudio, la calidad del profesorado y los recursos educativos[CITATION ANE21 \l 3082 ].
4. Sistema de Acreditación de la Educación a Distancia (DEAC): DEAC es una organización internacional que se encarga de acreditar los programas y cursos de educación a distancia en todo el

## CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS

mundo. Esta organización utiliza una metodología basada en la evaluación de los planes de estudio, la calidad del profesorado y los recursos educativos[CITATION Dis21 \l 3082 ].

5. Sistema de Acreditación de la Educación Superior (CHEA): CHEA es una organización internacional que se encarga de acreditar los programas y cursos de educación superior en todo el mundo. Para ello, utiliza una metodología basada en la evaluación de los planes de estudio, la calidad del profesorado y los recursos educativos[CITATION Cou21 \l 3082 ].

**Tabla 1.** Comparación de los sistemas homólogos estudiados.

| Sistemas homólogos/<br>Aspectos a evaluar   | Gratuita | Acceso al<br>código | Revisión de<br>cursos online | Indicadores<br>de calidad |
|---|----------|---------------------|------------------------------|---------------------------|
| Sistema de Garantía de Calidad de la Universidad Nacional de Educación a Distancia (UNED) | sí       | no                  | no                           | sí                        |
| Sistema de Acreditación de la Calidad de los Títulos Universitarios (ANECA)               | no       | no                  | no                           | sí                        |
| Sistema de Acreditación de la Educación a Distancia (DEAC)                                | no       | no                  | no                           | no                        |
| Sistema de Acreditación de la Educación Superior (CHEA)                                   | no       | no                  | no                           | no                        |
| Sistema de Acreditación de la Educación en Línea (SACSCOC)                                | no       | no                  | no                           | no                        |

**Fuente:** elaboración propia.

Como se muestra en la tabla anterior, y a pesar de que los sistemas UNED y ANECA se rigen por una metodología que contempla un conjunto de indicadores de calidad, ninguno de los sistemas estudiados posee un mecanismo para la revisión y evaluación de la calidad de los cursos virtuales. De igual modo, al ser desarrollados fuera del territorio nacional, los autores no tienen acceso al código fuente para adaptarlo al contexto de la presente investigación. Todo esto evidencia la necesidad de desarrollar un sistema capaz de gestionar los indicadores de calidad de los cursos virtuales disponibles en el aula virtual del CENED.

### I.3 Metodología de desarrollo

Una metodología es una disciplina de conocimiento encargada de elaborar, definir y sistematizar el conjunto de técnicas, métodos y procedimientos que se deben seguir durante el desarrollo de un proceso de investigación para la producción de conocimiento[ CITATION Fab23 \l 3082 ]. Dado que la solución se desarrolla en el marco productivo de la UCI, donde se tiene definida como metodología

una variación del Proceso Unificado Ágil (AUP del inglés *Agile Unified Process*) se decide utilizar dicha metodología para guiar el desarrollo de la solución propuesta.

### I.3.1 Metodología de desarrollo de software AUP versión UCI

La UCI desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva a la universidad. En esta versión se mantiene la fase de Inicio para el ciclo de vida de los proyectos, pero se modifican los objetivos de la misma y se unifican en la fase de Ejecución las fases restantes de la metodología AUP, y se agrega la fase de Cierre[CITATION Tam14 \l 3082 ].

Por otro lado, consta de 4 escenarios para modelar el sistema aplicando diferentes formas de encapsular los requisitos, las cuales son: Caso de Uso del Sistema, Historias de Usuario y Descripción de Requisitos por Proceso (CUS, HU y DRP por sus siglas) y variantes de modelado de negocio tales como Caso de Uso del Negocio, Descripción de Proceso de Negocio o Modelo Conceptual (CUN, DPN y MC), siendo estos[CITATION Tam14 \l 3082 ].

- Escenario No 1: proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

$$\text{CUN} + \text{MC} = \text{CUS}$$

- Escenario No 2: proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

$$\text{MC} = \text{CUS}$$

- Escenario No 3: proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

$$\text{DPN} + \text{MC} = \text{DRP}$$

- Escenario No 4: proyectos que no modelen negocio solo pueden modelar el sistema con HU.

$$\text{HU.}$$

Para encapsular los requisitos de la solución propuesta se decide utilizar el escenario #4, ya que al evaluar el negocio se obtuvo un negocio bien definido, por lo cual no es necesario modelar el negocio. Además, el cliente estará acompañando al equipo de desarrollo para convenir los detalles de los requisitos encontrados y poder implementarlos, probarlos y validarlos. De ahí que se pueden encapsular los requisitos a través de las HU.

#### I.4 Herramienta y lenguaje modelado

Las herramientas de software utilizadas cumplen con el objetivo de facilitar, optimizar y mejorar el desempeño del trabajo. Son un conjunto de programas y ayudas a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software, permitiendo la creación de todo tipo de diagramas[CITATION Raf \l 3082 ]. Para el desarrollo de la solución propuesta, se hace necesario el uso de una herramienta de Ingeniería de Software Asistida por Computadora que permita obtener una abstracción de la realidad en donde incide problemática, a través de los diferentes artefactos a generar en cada una de las disciplinas de la metodología que guía la presente investigación. Esto con el fin de representar los elementos claves del proceso de modo que sea posible una mejor comprensión.

**Visual Paradigm:** es una herramienta que utiliza UML como lenguaje del modelado. Esta herramienta está pensada para usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades en las que se destacan: Ingeniería de Software, análisis de sistemas y análisis de negocio[ CITATION Par13 \l 3082 ]. La utilización de la herramienta en cuestión ayudará a aumentar la productividad durante el desarrollo de la solución, automatizando parte del proceso, además, la documentación estará enriquecida con diagramas para lograr un mejor entendimiento del funcionamiento del sistema para futuros desarrolladores.

Se decide trabajar con el Visual Paradigm (v8.0) para UML por su robustez, usabilidad y portabilidad. Además, la universidad posee una licencia académica para su utilización y soporta el ciclo completo del proceso de desarrollo de software.

**UML:** el Lenguaje de Modelado Unificado (UML del inglés *Unified Modeling Languag*) es un lenguaje basado en diagramas para la especificación, visualización, construcción y documentación de cualquier sistema complejo, se basa en el paradigma orientado a objeto, es un modelo que simplifica la realidad que construimos para comprender mejor el sistema a desarrollar; proporciona los planos de un sistema, incluyendo tanto los que ofrecen una visión global del sistema como lo más detallado de alguna de sus partes[ CITATION Cra03 \l 3082 ]. Se selecciona UML (v2.0) ya que es el lenguaje de modelado que trae definido la herramienta CASE seleccionada, por lo que posibilita estandarizar la generación de artefactos referentes al proceso de desarrollo de software.

### I.5 Herramienta de prototipado

Es una herramienta utilizada en el proceso de diseño y desarrollo de productos o aplicaciones para crear representaciones visuales o interactivas de los mismos antes de su implementación final. Estas representaciones, conocidas como prototipos, permiten a los diseñadores y desarrolladores probar y validar ideas, obtener retroalimentación de los usuarios y realizar ajustes antes de invertir tiempo y recursos en la implementación completa[ CITATION War09 \l 3082 ].

**Figma:** se trata de una herramienta de diseño enfocado en sitios web y aplicaciones. A diferencia de otras herramientas de prototipado, Figma se aloja en la web y permite la colaboración en vivo. Tiene una versión de pago para profesionales con capacidad ilimitada de proyectos, versiones y que permite otorgar permisos individuales a determinados ficheros o usuarios, que puede ser obtenida gratuitamente justificando que el usuario que solicita es un estudiante o un docente. También hay una versión corporativa que permite crear plugins privados, compartir fuentes entre organizaciones y habilitar herramientas de análisis de los prototipos generados[ CITATION Ism21 \l 3082 ].

Se decide la utilización de Figma (v108.1) como herramienta de prototipado ya que es una herramienta gratis en su plan básico. Además, posee gran relevancia en el mercado de la industria de software y la emplean tanto diseñadores como desarrolladores, permitiéndole evaluar los prototipos con el cliente de una manera rápida y eficaz y multiplataforma.

### I.6 Lenguajes de desarrollo

Los lenguajes de programación se utilizan para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora[ CITATION Lóp20 \l 3082 ]. En este epígrafe se hace referencia sobre diversos lenguajes de programación empleados en la confección de la propuesta de solución. Estos son utilizados para dar forma al sistema de gestión y obtener una aplicación funcional que resuelva la problemática de esta investigación.

**Python:** es un lenguaje fácil de aprender. Tiene una cómoda sintaxis y un tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. Posee estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos[ CITATION Ros09 \l 3082 ].

Se decidió emplear el lenguaje de programación Python (v3.8), ya que es un lenguaje que hace referencia a su propia limpieza y legibilidad, está desarrollado bajo una licencia de código abierto, por lo que es de libre uso y distribución, incluso para uso comercial. Posee buen rendimiento, además su característica multiplataforma permite que este pueda ser usado en diferentes sistemas operativos. Posee una gran comunidad a nivel mundial que están constantemente aportando nuevas mejoras. Además, es el lenguaje de programación en el cual el equipo de desarrollo tiene mayor conocimiento, por lo que se ahorraría tiempo en capacitación sobre este lenguaje

**JavaScript:** es un lenguaje de programación ligero, interpretado, o compilado justo a tiempo. Si bien es más conocido como un lenguaje de *scripting*<sup>1</sup> para páginas web, y es usado en [entornos fuera del navegador](#)[CITATION Rut23 \l 3082 ]. JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario[ CITATION Egu20 \l 3082 ].

Es un lenguaje de programación versátil que ofrece varias ventajas, por ejemplo: permite agregar interactividad a las páginas web, mejorando la experiencia del usuario. Además, es fácil de aprender y utilizar, con una sintaxis similar a otros lenguajes populares. JavaScript es compatible con todos los navegadores modernos y cuenta con una amplia comunidad de desarrolladores y recursos disponibles. También se puede utilizar tanto en el lado del cliente como en el lado del servidor, lo que lo hace muy versátil. Se integra fácilmente con HTML y CSS, permitiendo la manipulación dinámica del contenido y la apariencia de una página web. JavaScript se actualiza constantemente, añadiendo nuevas características y mejoras de rendimiento. Por último, ofrece una ejecución rápida y eficiente de los scripts al ser interpretado en el navegador. Se decidió utilizar JavaScript (v1.9) debido a que este lenguaje permite darle dinamismo y versatilidad al sistema que se propone para la revisión de los indicadores de calidad de los cursos virtuales, logrando así una aplicación web interactiva y dinámica.

**Hojas de Estilo en Cascada:** las Hojas de Estilo en Cascada (CSS del inglés *Cascading Style Sheets*) podrían definirse como un tipo de lenguaje que permite definir y crear la presentación de un documento ya estructurado y escrito en un lenguaje de marcado como puede ser HTML. Es decir, permite generar el diseño visual de páginas web e interfaces de usuario. Cabe agregar que el lenguaje se puede aplicar en la misma hoja en la que se está desarrollando un documento HTML, pero por motivos de productividad se suele realizar en un documento aparte con la extensión \*.css. Este documen-

---

1 Secuencias de comandos

to se puede vincular a cada página HTML que conforme el sitio web, es por ello que es más útil realizar los estilos por separado. CSS funciona mediante módulos, algunos de los más comunes son “*colors*”, “*fonts*”, “*backgrounds*”. Los módulos son sólo categorías en las que se pueden dividir las modificaciones que hacen al aspecto del sitio web[ CITATION Ram17 \l 3082 ]. Se decide emplear CSS v3.0 para crear una interfaz intuitiva y amena para los usuarios a los que va dirigida la propuesta de desarrollo.

**Lenguaje de Marcas de Hipertexto:** el Lenguaje de Marcas de Hipertexto (HTML del inglés *Hyper-Text Markup Language*) es un lenguaje de etiquetas, utilizado para la estructuración y la presentación de contenido en los sitios web. Una de las características fundamentales es su universalidad, esto posibilita que prácticamente cualquier ordenador, independientemente del sistema operativo que tenga, pueda leer o interpretar una página web[CITATION JRo \l 3082 ].

Es un sistema para formatear las páginas y hacer ajustes a su aspecto. A través de este, los navegadores como Mozilla Firefox, Google Chrome, Internet Explorer y otros pueden saber cómo mostrar una página web determinada, ubicar los elementos, textos e imágenes. Se puede reducir la dependencia de complementos y amplía el horizonte del desarrollo de aplicaciones que pueden ser usadas en múltiples dispositivos[ CITATION Gau12 \l 3082 ]. Se decidió la utilización de HTML (v5.0), debido a que es el lenguaje de estructuración básica actual de los elementos que se emplea en el diseño y maquetado de las interfaces de la propuesta de desarrollo.

### 1.7 Bibliotecas y marcos de trabajo para el desarrollo

Los marcos de trabajo (*framework*) en el desarrollo de software son una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software[ CITATION Dav12 \l 3082 ].

Seguidamente se describe el marco de trabajo de desarrollo para el lenguaje de programación de Python y se caracteriza la biblioteca JavaScript que fue empleada en la confección de la solución. De igual forma se menciona el motivo de su adopción para la presente solución.

**Django:** Django es un *framework* web extremadamente popular y completamente funcional, escrito en Python. Se trata de una herramienta de código abierto y gratuita que cuenta con una comunidad amplia y que comparte recursos constantemente. Se puede usar para el desarrollo *full-stack* de aplicaciones y páginas web, así como para el desarrollo de servidores. Posee varios módulos integrados para la seguridad y permite realizar operaciones básicas con solo unas pocas líneas de

códigos, incrementando la calidad de las soluciones, la productividad y disminuyendo los errores en el código. Igualmente es un potente marco de trabajo que posee una arquitectura bien definida [ CITATION Wil20 \l 3082 ]. Se dispuso emplear el *framework* Django (v4.0) por las características anteriormente escritas unido al conocimiento sobre este marco de trabajo por los autores de la presente investigación, lo que facilita la implementación y se evita empelar tiempo para capacitar a los desarrolladores.

**jQuery:** es una biblioteca multiplataforma de JavaScript que, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, las funciones propias de esta biblioteca permiten realizar tareas de manera más eficiente al aprovechar al máximo los recursos disponibles. Esto se traduce en una reducción significativa del tiempo requerido para completar las tareas. Además, estas funciones están diseñadas para ocupar menos espacio en la memoria, lo que resulta en un uso más eficiente de los recursos del sistema [ CITATION Mig10 \l 3082 ].

Se dispuso emplear jQuery (v3.6) debido a que existen un amplio desarrollo de *plugin* para esta biblioteca que ayudan a dinamizar el contenido, el tratamiento y comunicación entre el servidor y el cliente. También este contiene un conjunto de funcionalidades para el trabajo con librerías de animación, validación y componentes personalizados que facilitan y extienden las funcionalidades de la programación con JavaScript.

### I.8 Entorno de desarrollo

Un entorno de desarrollo integrado o entorno de desarrollo interactivo, es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software, la mayoría de los entornos de desarrollo integrado (IDE) incluyen funcionalidades que van más allá de la edición de texto. Además, proporcionan una interfaz central para herramientas de desarrollo comunes, lo que hace que el proceso de desarrollo de software sea mucho más eficiente [ CITATION Red23 \l 3082 ].

**JetBrains PyCharm:** es un entorno de desarrollo integrado utilizado en programación informática, concretamente para el lenguaje de programación Python. Así ofrece un gran soporte específico para marcos de desarrollo web modernos como Django. Proporciona completamiento de código inteligente, inspecciones de código, resaltado de errores sobre la marcha y correcciones rápidas, junto con refactorizaciones de código automatizadas y ricas capacidades de navegación [ CITATION Jet22 \l 3082 ].

Se dispuso emplear JetBrains PyCharm (v2022.2) debido al gran soporte que ofrece al *framework* Django permitiéndole al programador un editor de Python inteligente, navegación, refactorización e inspecciones del código.

### I.9 Sistema gestor de base de datos

Un sistema gestor de base de datos o SGBD es un software que permite administrar una base de datos. Esto significa que mediante este programa se puede utilizar, configurar y extraer información almacenada. Igualmente permite una vista muy centralizada y clara de los datos para que sean accedidos de la mejor manera posible. Por tanto, se encarga de gestionar adecuadamente los datos, evitando a los usuarios o programas que les requieren, tener que entender dónde se encuentran físicamente los datos[CITATION com23 \l 3082 ].

**PostgreSQL:** es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto. Está diseñado para ambientes de altos volúmenes de datos y al ser multiplataforma está disponible para muchos sistemas operativos, ha ganado mucha reputación por su confiabilidad e integridad en los datos[ CITATION Ord19 \l 3082 ].

Es multiplataforma, tiene soporte nativo para PHP. También permite todo tipo de consultas a bases de datos. Brinda soporte multiusuario, para el control de concurrencia, dispone de los tipos de datos del estándar SQL, así como soporte para tipos de datos creados por el usuario. En términos de licencia y recursos, es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, lo que posibilita atender un mayor número de peticiones concurrentes[ CITATION Ord19 \l 3082 ].

Se selecciona PostgreSQL (v14.2) para el desarrollo de esta aplicación por las ventajas que posee ante el desarrollo de este sistema. Además, está en correspondencia, según[ CITATION Pér15 \l 3082 ], con la estrategia para la migración a aplicaciones de código abierto definida para la universidad.

### I.10 Servidor web

Un servidor web es un software que forma parte del servidor y tiene como misión principal devolver información (páginas) cuando recibe peticiones por parte de los usuarios. En informática es una pieza de software de comunicaciones que intermedia entre el servidor en el que están alojados los datos solicitados y el computador del cliente, permitiendo conexiones bidireccionales o unidireccionales, síncrona o asíncronas, con cualquier aplicación del cliente, incluso con los navegadores que traducen

un código traducible (renderizable) a una página web determinada. O sea, se trata de programas que median entre el usuario de Internet y el servidor en donde está la información que solicita [CITATION Edi23 \l 3082 ].

**Nginx:** es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico. Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada. NGINX entrega el contenido estático del sitio web rápidamente, es fácil de configurar y tiene un bajo consumo de recursos [ CITATION Edg20 \l 3082 ]. Se decide utilizar Nginx (v1.20) en la propuesta de solución teniendo en cuenta las características antes descritas y las tendencias actuales de la universidad, además de que este servidor web tiene buen soporte para configurar y desplegar sistemas web desarrollados con el *framework* Django.

### I.11 Control de versiones

Los sistemas de control de versiones son software que ayudan a realizar un seguimiento de los cambios realizados en el código a lo largo del tiempo. A medida que un desarrollador edita el código, el sistema de control de versiones toma una instantánea de los archivos. Después, guarda esa instantánea de forma permanente para que se pueda recuperar más adelante si es necesario [CITATION Azu23 \l 3082 ].

**Git:** es un sistema de control de versiones, un software que sirve básicamente para gestionar las versiones por las que va pasando el código de los proyectos. Git es el más popular de los sistemas de control de versiones en la actualidad y una de las herramientas más indispensables para el desarrollo de proyectos. Aunque un sistema de control de versiones sirva justamente para controlar los estados por los que ha pasado un código, lo cierto es que la herramienta Git facilita mucho el desarrollo de los proyectos en equipo, así como otras operaciones relacionadas con el flujo de trabajo de los desarrolladores [CITATION Cri \l 3082 ].

Se decide utilizar Git (v2.2) en la propuesta de solución teniendo en cuenta las características antes descritas y las tendencias actuales de la universidad.

### Conclusiones del capítulo

Luego de haber realizado el establecimiento de los fundamentos teóricos-metodológicos en los que se sustenta la presente investigación se tiene que:

- El examen realizado a los diferentes conceptos relacionados con la gestión de indicadores de calidad de los cursos virtuales disponibles, permitió establecer las bases para adquirir un

previo conocimiento acerca de la presente investigación.

- El análisis de los diferentes sistemas homólogos evidenció la necesidad de desarrollar un sistema capaz de gestionar los indicadores de calidad de los cursos virtuales disponibles en el aula virtual del CENED.
- La selección de la metodología AUP en su variación para la UCI en el escenario #4 contribuyó a sentar las bases que guiaran el desarrollo de la solución propuesta, partiendo desde la especificación de los requisitos de software hasta su implementación y pruebas.
- La caracterización de las herramientas y lenguajes de desarrollo demostró que estas están en correspondencia con las políticas de migración de software de la UCI, aportando a la independencia tecnológica a la que aspira el país.

En este capítulo se abordan los elementos centrales relacionados con la propuesta de solución. Primeramente, se especifican los requisitos funcionales y no funcionales, donde se encapsulan los funcionales mediante historias de usuarios. Asimismo, se establece la estructura básica de la arquitectura del sistema, considerando detalles específicos de diseño e implementación. A partir de ello, se elaboran los diagramas de clases del diseño utilizando estereotipos web, los cuales brindan una visión más clara de cómo se implementará el sistema. Por otro lado, se describen los patrones de diseño empleados y se define el modelo de datos de la solución. Por último, y como parte de la vista de implementación del sistema, se generan el diagrama de componente de la historia de usuario a describir y el diagrama de despliegue de la solución propuesta.

### **II.1 Descripción de la solución propuesta**

El sistema que se propone como solución al problema que da paso a la presente investigación hace referencia a una aplicación web que le permitirá a los profesores del CENED gestionar los indicadores de calidad para la posterior revisión y evaluación de los cursos disponibles en el aula virtual. Donde contará con un mecanismo de notificaciones, enviando recordatorios automáticos a los usuarios que no hayan realizado la revisión dentro de la fecha límite establecida. Estas notificaciones podrán incluir información sobre las consecuencias de no cumplir con la revisión, como posibles sanciones o retrasos en la publicación del curso. Además, permitirá generar reportes de la revisión, contando con una función que le permitirá al profesor supervisor generar informes detallados sobre el estado de cada curso. Para el caso de la administración del sistema, contará con procedimiento mediante el cual se podrán autenticar por el directorio activo propio de la universidad, permitiendo activar o desactivar usuarios, así como asignarles permisos y roles específicos. Esto le permitirá al administrador tener un control total sobre quién puede acceder y realizar determinadas funciones dentro del sistema.

### **II.2 Requisito de software**

Un requisito de software es una declaración que especifica una función o propiedad que debe cumplir un sistema o componente de software. Estos requisitos describen las necesidades y expectativas de los usuarios y otras partes interesadas, y sirven como base para el diseño, desarrollo y prueba del software[ CITATION Kar13 \l 3082 ].

Con el objetivo de identificar las especificidades del negocio y así lograr identificar las funcionalidades requeridas y las restricciones a tener en cuenta durante el desarrollo de la solución propuesta, se

## CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

realizaron varias reuniones con el cliente, donde se formularon un conjunto de preguntas con el objetivo de entender las necesidades del mismo y concebir un lenguaje común entre el cliente y el equipo de desarrollo. Una vez aclarados cada uno de los puntos durante las reuniones, se definieron los requisitos a cumplir para el desarrollo de la solución propuesta, dejando documentado y especificados los requisitos funcionales y no funcionales, los cuales se detallan a continuación.

### II.2.1 Requisitos funcionales (RF)

Los requisitos funcionales (RF) de un sistema se refieren a las funciones y acciones específicas que el sistema debe ser capaz de realizar. Estos requisitos describen las tareas y operaciones que el sistema debe llevar a cabo para cumplir con los objetivos y necesidades del usuario [ CITATION new \l 3082 ]. Para darle solución al problema planteado que da paso a la presente investigación, se identificaron 20 RF, en la **Tabla 2** se muestra la especificación de estos requisitos:

**Tabla 2.**Listado de Requisitos Funcionales del sistema.

| Nº  | Nombre             | Descripción  |
|-----|--------------------|--|
| RF1 | Autenticar usuario | El sistema debe permitir al revisor principal (RP), el revisor general (RG) autenticarse para lo cual se deberán ingresar los siguientes datos siempre y cuando sean del dominio UCI: <ul style="list-style-type: none"><li>• Usuario</li><li>• Contraseña</li></ul>   |
| RF2 | Desactivar Usuario | El sistema debe permitir al administrador desactivar al usuario.   |
| RF3 | Revisar Curso      | El sistema debe permitir al revisor principal y revisor general revisar los cursos para lo cual se indica: <ul style="list-style-type: none"><li>• Título de cada curso</li><li>• Profesores</li><li>• Fecha de revisión preliminar</li><li>• Plazo de fecha</li><li>• Fecha de la primera revisión de calidad</li><li>• Fecha de la segunda revisión de calidad</li></ul> |
| RF4 | Modificar revisión | El sistema debe permitir modificar revisión al profesor principal y directivos del CENED.  |

## CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

|      |                             |   |
|------|-----------------------------|---|
| RF5  | Modificar indicador         | El sistema debe permitir modificar indicadores al profesor principal y directivos del CENED para lo cual se indica: <ul style="list-style-type: none"> <li>• Iteración de la Revisión</li> <li>• Institución</li> <li>• Modalidad</li> <li>• Profesor principal</li> <li>• Otros profesores</li> <li>• Revisores</li> <li>• Revisor responsable</li> <li>• Periodo de evaluación</li> </ul> |
| RF6  | Recibir reporte             | El sistema debe permitir enviar reporte del estado de los cursos en cualquier etapa de revisión.  |
| RF7  | Asignar curso               | El sistema debe permitir al profesor principal asignar curso a los profesores revisores.  |
| RF8  | Listar profesores revisores | El sistema debe mostrar un listado con los nombres de todos los profesores revisores.   |
| RF9  | Registrar profesor revisor  | El sistema debe permitir el registro de un nuevo profesor revisor para lo cual se indica: <ul style="list-style-type: none"> <li>• Nombre y apellidos</li> <li>• Categoría científica</li> </ul>  |
| RF10 | Listar curso                | El sistema debe permitir mostrar un listado con todos los cursos.   |
| RF11 | Registrar curso             | El sistema debe permitir el registro de un nuevo curso para lo cual se tiene: <ul style="list-style-type: none"> <li>• Nombre del curso.</li> <li>• Tipo de curso.</li> <li>• Estado del curso.</li> </ul>  |
| RF12 | Modificar curso             | El sistema debe permitir modificar curso al profesor principal y directivos del CENED para lo cual se indica: <ul style="list-style-type: none"> <li>• Nombre del curso.</li> <li>• Tipo de curso.</li> <li>• Estado del curso.</li> </ul>  |
| RF13 | Eliminar curso              | El sistema debe permitir al administrador eliminar curso.   |
| RF14 | Buscar cursos               | El sistema debe permitir la busca de un curso a partir de la introducción de los siguientes datos: <ul style="list-style-type: none"> <li>• Nombre del curso.</li> <li>• Tipo de curso.</li> <li>• Estado del curso.</li> </ul>   |
| RF15 | Enviar notificación         | El sistema debe enviar un correo de notificación automática a los revisores que debe comenzar revisión preliminar.  |
| RF16 | Enviar informe              | El sistema debe permitir que el profesor principal envíe los resultados de la revisión a los profesores de los cursos.  |
| RF17 | Generar modelo de la        | El sistema debe permitir generar un modelo con los datos estableci-   |

**CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO**

|      |                                       |  |
|------|---------------------------------------|--|
|      | 1ra revisión general                  | dos en la revisión general.  |
| RF18 | Generar modelo de revisión preliminar | El sistema debe permitir generar el modelo preliminar de revisión.   |
| RF19 | Generar modelo de la segunda revisión | El sistema debe permitir generar un modelo de la segunda revisión.   |
| RF20 | Evaluar indicadores                   | <p>El sistema debe permitir al profesor revisor evaluar en Listo o No listo los indicadores:</p> <p>1 – Cumplimiento de la estructura:</p> <ul style="list-style-type: none"> <li>• Bienvenida al curso</li> <li>• Guía de estudio</li> <li>• Espacio estructurado para los Recursos Educativos Digitales (opcional)</li> <li>• Glosario de términos de la asignatura</li> <li>• Foro de aviso, novedades y dudas</li> <li>• Sala de encuentros</li> <li>• Foro de profesores (Opcional)</li> <li>• Orientaciones de los temas</li> <li>• Espacio estructurado para los Recursos Educativos Digitales</li> <li>• Espacio para la bibliografía del tema</li> <li>• Actividades autoevaluativas por temas</li> <li>• Actividades evaluativas por temas</li> <li>• Actividad de cierre para la asignatura</li> <li>• Retroalimentación del curso</li> </ul> <p>2 – Diseño general del curso:</p> <ul style="list-style-type: none"> <li>• Correspondencia entre el nombre del archivo, recurso y actividad y su contenido.</li> <li>• Estructuración de la secuencia pedagógica en inicio, desarrollo y cierre.</li> <li>• Cumplimiento de la estructura del curso (imagen de portada, contraseña del curso).</li> <li>• Orientación hacia el trabajo individual y/o colaborativo de los estudiantes.</li> <li>• Coherencia entre el vocabulario utilizado y el nivel de los estudiantes para quienes está diseñado el curso.</li> <li>• Uso adecuado de colores en correspondencia con el contexto de aprendizaje.</li> <li>• Cumplimiento de normas ortográficas y de redacción.</li> <li>• Homogeneidad en la identidad visual de los diferentes elementos que conforman el curso.</li> <li>• Pertinencia de los términos del glosario.</li> </ul> |

## CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

|  |  |   |
|--|--|---|
|  |  | <ul style="list-style-type: none"> <li>• Integración de los contenidos en las actividades de cierre del tema y el curso.</li> </ul> <p>Diversidad de recursos educativos (OA, infografías, mapas, líneas de tiempo, animaciones, etc.)</p> <p>3 –Contenidos:</p> <ul style="list-style-type: none"> <li>• Contextualización de los contenidos a situaciones prácticas.</li> <li>• Vigencia de los conceptos y términos emitidos.</li> <li>• Suficiencia y profundidad de la información.</li> <li>• Calidad técnica de los elementos multimedia (imagen, audio y video).</li> <li>• Relevancia de los ejemplos para ilustrar los contenidos.</li> <li>• Correspondencia entre las orientaciones y los recursos y actividades de cada tema.</li> <li>• Integridad de los enlaces de navegación.</li> <li>• Coherencia entre las fuentes de información empleadas y los objetivos.</li> <li>• Uniformidad en el uso de citas y referencias bibliográficas.</li> <li>• Licenciamiento abierto de las fuentes de información utilizadas para el desarrollo de los contenidos.</li> </ul> <p>4–Diseño de actividades:</p> <ul style="list-style-type: none"> <li>• Diversidad de actividades para la construcción colectiva de conocimientos (wiki, foros, chat, talleres, entre otros).</li> <li>• Correspondencia y pertinencia de las actividades (ejercitación, autoevaluación y evaluación) y los objetivos del tema.</li> <li>• Precisión de las instrucciones para el desarrollo de las actividades de aprendizaje.</li> <li>• Niveles de ayuda para el desarrollo de las actividades (rúbricas, tiempo estimado, sugerencias y/o modelos para su desarrollo).</li> <li>• Tratamiento de las respuestas acertadas o incorrectas presentes en los ejercicios de autoevaluación (retroalimentación).</li> <li>• Disponibilidad de actividades de autoevaluación y evaluación individuales y colaborativas.</li> </ul> |
|--|--|---|

**Fuente:** elaboración propia.

## II.2.2 Requisitos no funcionales (RnF)

Los requisitos no funcionales (RnF) se refieren a los atributos y características del sistema que no están relacionados directamente con las funciones y tareas específicas que el sistema debe realizar, sino más bien a cómo debe comportarse o cumplir ciertos criterios. Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir [ CITATION new \l 3082 ]. A diferencia de los requisitos funcionales, que se refieren a las funciones y tareas específicas que debe realizar el sistema, los requisitos no funcionales se centran en cómo debe comportarse y cumplir ciertos criterios. Teniendo en cuenta que forman parte de la solución del presente trabajo de diploma se definieron como RnF los siguientes:

**RnF 1. Requisito de Usabilidad 1.1** El sistema será una aplicación web con una curva de aprendizaje baja que pueda ser usada por cualquier persona que posea un nivel básico de conocimientos de computación.

**RnF 2. Requisito de Usabilidad 1.2** El sistema debe utilizar nombres sugerentes para lograr que el usuario encuentre lo que busca en el menor tiempo posible, las acciones a realizar serán fáciles de acceder.

**RnF 3. Requisito de Usabilidad 1.3** El sistema debe cumplir con la regla de los tres clics para encontrar los recursos deseados dentro del sistema.

**RnF 4. Requisito de Usabilidad 1.4** El sistema debe visualizar todos los mensajes en idioma español.

**RnF 5. Requisito de Usabilidad 1.5** El sistema debe utilizar una iconografía de forma tal que se identifiquen claramente con lo que representan.

**RnF 6. Requisito de Interfaz 2.1** El diseño de la interfaz debe estar regido por la filosofía establecida en Bootstrap. Esto enfoca el uso de los colores, las sombras, la profundidad, las superficies y los bordes en un diseño limpio y agradable a la vista.

**RnF 7. Requisito de Interfaz 2.2** El sistema debe garantizar una correcta organización de la información para permitir una adecuada interpretación.

**RnF 8. Requisito de Interfaz 2.3** El sistema debe poseer la siguiente gama de colores en su diseño: #F58800, #F8BC24, #266867, #1A4645, #051821.

**RnF 9. Requisito de Seguridad 3.1** El sistema debe usar roles para especificar los privilegios de cada usuario.

**RnF 10. Requisito de Seguridad 3.2** Debe estar preparado ante ataques *Cross Site Request Forgery* (CSRF o XSRF) para evitar peticiones a una aplicación web vulnerable.

**RnF 11. Requisito de Seguridad 3.3** Debe eliminar la posibilidad de ser víctima de ataques tipo XSS (*Cross Site Scripting*) que ejecutan código malicioso en el navegador del usuario de la aplicación.

**RnF 12. Requisito de Software 4.0** El sistema se desplegará en un servidor que tenga instalado: sistema gestor de bases de datos PostgreSQL (v12.2) o superior, servidor de aplicaciones web Nginx (v1.20), Python (v3.8) o superior.

**RnF 13. Requisito de Software 4.1** Del lado del cliente es necesario tener instalado un navegador que cumpla los estándares de la W3C, preferentemente Microsoft Edge, Chrome o Firefox.

**RnF 14. Requisito de Software 4.2** Es necesario tener instalados en el servidor los paquetes de Python: django (4.0.2), django-auth-ldap (4.0.0), django-bootstrap-modal-forms (2.2.0), django-notifications-hq (1.6.0), django-widget-tweaks (1.4.12), Pillow (9.0.1), python-ldap (3.4.0).

**RnF 15. Requisito de Hardware 5.0** El sistema debe poder ejecutarse en una PC cliente con hardware con microprocesadores con Intel Pentium o superior.

**RnF 16. Requisito de Hardware 5.1** El servidor donde se va a instalar la solución informática debe disponer de aproximadamente 8GB de RAM y 500GB de espacio de almacenamiento en disco como mínimo y un procesador Intel Core i3 a 2,4GHz en el hardware de despliegue.

**RnF 17. Requisito de Hardware 5.2** En las máquinas clientes es necesario que posean al menos 1GB de memoria RAM y podrán visualizarse desde cualquier sistema de cómputo (móvil o de escritorio) con el software necesario.

**RnF 18. Requisito de Hardware 5.3** Los hardware clientes y servidor deben poseer una tarjeta de interconexión de red que permita acceder y responder a las peticiones respectivamente.

### **II.3 Historias de usuario**

Las Historias de Usuario (HU) son una forma eficiente y ágil de capturar los requisitos de los usuarios, utilizando un lenguaje sencillo y comprensible para todos los involucrados en el proyecto. Permiten gestionar los requisitos de manera más ágil y flexible, sin la necesidad de elaborar documentos extensos y formales. Esto facilita la adaptación y respuesta rápida a los cambios en los requisitos del usuario [ CITATION Ken17 \l 3082 ]. En resumen, las historias de usuario son una técnica ágil para describir las funcionalidades y requisitos del producto desde la perspectiva del usuario final, facilitando su implementación [ CITATION Rom10 \l 3082 ]. Para la encapsulación de los requisitos funcionales fueron definidas un total de 20 HU, una por cada uno de los RF identificados. A continuación, se describe dos HU de prioridad Alta para el cliente, obtenidas en el desarrollo de la solución propuesta.

**Tabla 3.** HU\_14 Revisar Curso.

|   |  |                                 |  |
|---|--|---------------------------------|--|
| <b>Número: 3</b>  |  | <b>Requisito: Revisar curso</b> |  |
| <b>Programador:</b> Jesús Alarcón Aguilar   |  | <b>Iteración Asignada:</b> 3    |  |
| <b>Prioridad:</b> Alta  |  | <b>Tiempo Estimado:</b> 8 horas |  |
| <b>Riesgo en Desarrollo:</b> N/A  |  | <b>Tiempo Real:</b> 7 horas     |  |
| <p><b>Descripción:</b> El sistema permitirá revisar curso. La HU inicia cuando el usuario se dirige a la opción “Revisar curso” en el módulo de cursos. Para que el usuario pueda revisar un curso en el sistema debe entrar los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Título de cada curso (obligatorio, campo de texto).</li> <li>• Profesores (obligatorio, campo de texto).</li> <li>• Fecha de revisión preliminar (obligatorio, campo de rango de fecha)</li> <li>• Plazo de fecha (obligatorio, campo de rango de fecha)</li> <li>• Fecha de la primera revisión de calidad (obligatorio, campo de rango de fecha)</li> <li>• Fecha de la segunda revisión de calidad (obligatorio, campo de rango de fecha)</li> </ul> <p>Luego dar clic en la opción: “Guardar registro”, finalizando así la HU. En caso de dar clic en la opción: “Cancelar”, el sistema regresa hacia la página anterior descartando la operación, finalizando así la HU.</p> |  |                                 |  |
| <p><b>Observaciones:</b> El usuario debe estar autenticado en el sistema y con permisos de rol profesor revisor o supervisor.</p>   |  |                                 |  |
| <p><b>Prototipo elemental de interfaz gráfica de usuario:</b></p>   |  |                                 |  |

**1 - Cumplimiento de la estructura:**

- Bienvenida al curso
- Guía de estudio
- Espacio estructurado para los Recursos Educativos Digitales (opcional)
- Glosario de términos de la asignatura
- Foro de aviso, novedades y dudas
- Sala de encuentros
- Foro de profesores (Opcional)
- Orientaciones de los temas
- Espacio estructurado para los Recursos Educativos Digitales
- Espacio para la bibliografía del tema
- Actividades autoevaluativas por temas
- Actividades evaluativas por temas
- Actividad de cierre para la asignatura
- Retroalimentación del curso

**Observaciones**

Escriba la descripción.

Cancelar

Guardar

*Fuente: elaboración propia.*

**Tabla 4. HU\_14 Registrar Curso.**

|   |                                   |
|---|-----------------------------------|
|   |                                   |
| <b>Número:</b> 11   | <b>Requisito:</b> Registrar curso |
| <b>Programador:</b> Adachely Miyares Magaña   | <b>Iteración Asignada:</b> 11     |
| <b>Prioridad:</b> Alta  | <b>Tiempo Estimado:</b> 7 horas   |
| <b>Riesgo en Desarrollo:</b> N/A  | <b>Tiempo Real:</b> 6 horas       |
| <p><b>Descripción:</b> El sistema permitirá registrar un curso. La HU inicia cuando el usuario se dirige a la opción “Registrar curso” en el módulo de cursos. Para que el usuario pueda registrar un curso en el sistema debe entrar los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Nombre del curso (obligatorio, campo de texto).</li> <li>• Tipo de curso (obligatorio, campo de texto).</li> </ul> |                                   |

- Estado del curso (obligatorio, campo de texto).

Luego dar clic en la opción: “Guardar registro”, finalizando así la HU. En caso de dar clic en la opción: “Cancelar”, el sistema regresa hacia la página anterior descartando la operación, finalizando así la HU.

**Observaciones:** El usuario debe estar autenticado en el sistema y con rol de profesor o profesor supervisor.

**Prototipo elemental de interfaz gráfica de usuario:**

El prototipo muestra un formulario con el título "Añadir curso". El formulario contiene los siguientes elementos:

- Nombre del curso:** Un campo de texto con el placeholder "Ingrese el nombre del curso".
- Modalidad:** Un menú desplegable con "Semipresencial" seleccionado.
- Categoría:** Un menú desplegable con "Maestría" seleccionado.
- Botones:** Dos botones al final del formulario: "Atrás" (rojo) y "Guardar" (verde).

*Fuente: elaboración propia.*

### II.4 Arquitectura del sistema

La arquitectura del sistema se refiere a la estructura y organización del software y hardware que forman parte de un sistema informático. Esta arquitectura define cómo se comunican entre sí los diferentes componentes y cómo se distribuyen las tareas y responsabilidades [ CITATION Len96 \l 3082 ].

En su forma más simple, la arquitectura se refiere a la estructura y organización de los componentes de un programa, así como a la forma en que estos interactúan y utilizan la estructura de datos. Sin embargo, en un sentido más amplio, los componentes pueden representar elementos importantes del sistema y sus interacciones. La arquitectura no es más que un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de software. Esto permite a los programadores, analistas y desarrolladores compartir una línea de trabajo común y abordar los objetivos

y restricciones de la aplicación[ CITATION new \l 3082 ]. Como parte de la arquitectura, a continuación, se describirá el patrón arquitectónico utilizado en la solución propuesta.

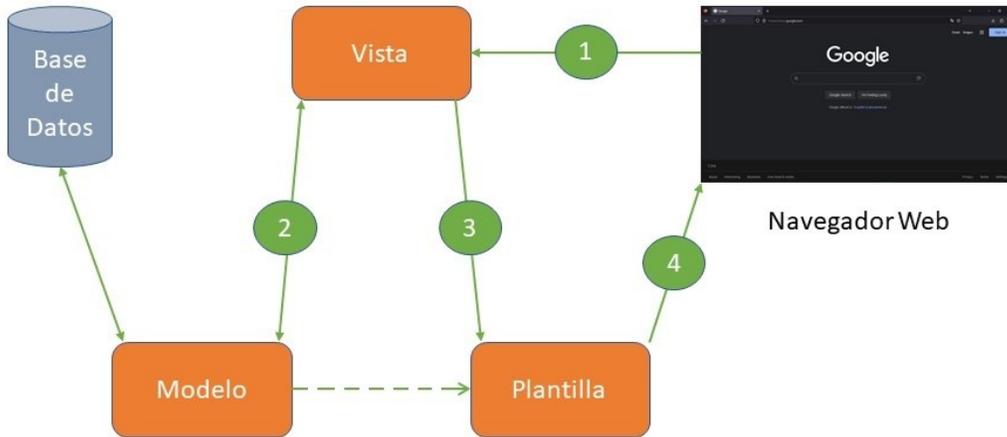
### **II.4.1 Patrón arquitectónico**

Un patrón de arquitectura de software es una solución para problemas relacionados con el diseño arquitectónico de un sistema. Estos patrones describen la relación entre los componentes del sistema y están compuestos por módulos más pequeños. Además, proporcionan un conjunto de subsistemas predefinidos que especifican sus responsabilidades y también incluyen reglas y pautas para organizar las relaciones entre ellos[ CITATION Eri94 \l 3082 ].

En general, los sistemas que separan la lógica de acceso a la base de datos, la lógica de negocios y la lógica de presentación se basan en el patrón de arquitectura de software Modelo-Vista-Controlador (MVC). En este patrón, el "Modelo" se refiere al componente encargado de acceder a la capa de datos. La "Vista" es la parte del sistema encargada de seleccionar qué mostrar y cómo mostrarlo. Por último, el "Controlador" es responsable de decidir qué vista utilizar, dependiendo de la entrada del usuario, y puede acceder al modelo si es necesario[ CITATION Mar03 \l 3082 ].

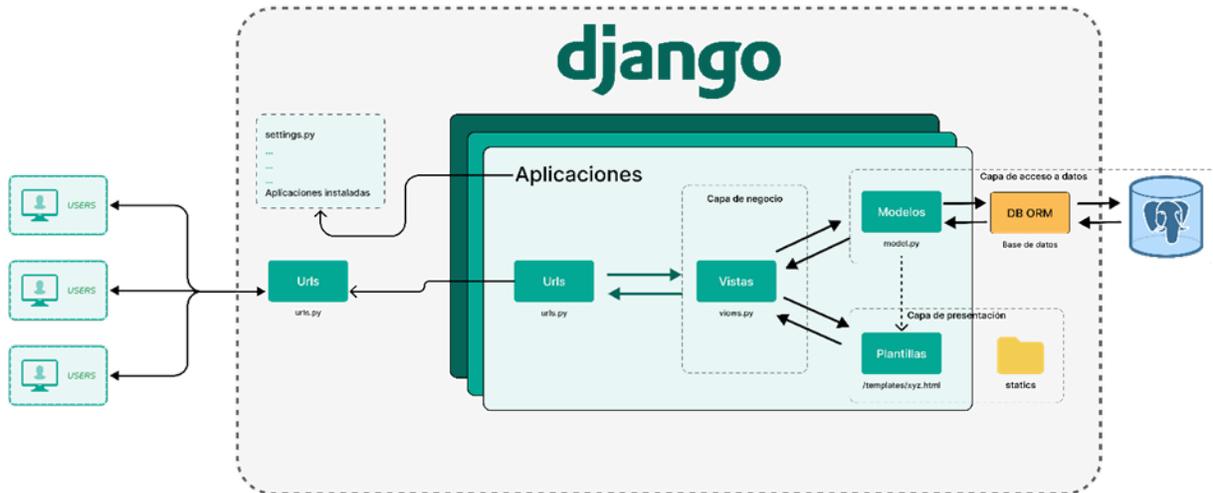
Django aplica de manera similar el patrón Modelo-Vista-Controlador (MVC). El Controlador es manejado por el propio framework y se encarga de enrutar las solicitudes del usuario a la vista correspondiente. La parte más importante se produce en los modelos, las vistas y las plantillas. Django es conocido por su uso del patrón arquitectónico Modelo-Vista-Plantilla (MVT), que es una variante del patrón MVC[CITATION Aru \l 3082 ].

Cuando se realiza una solicitud desde un navegador, las vistas se encargan de procesar la lógica del negocio utilizando los datos del modelo de base de datos. Luego, los resultados de esta lógica se envían a la plantilla correspondiente, la cual se devuelve como respuesta al proceso y se muestra al usuario.



**Figura 1.** Funcionamiento del patrón arquitectónico Modelo-Vista-Plantilla.

**Fuente:**[CITATION Egu11 \I 3082 ]



**Figura 2.** Arquitectura del proyecto.

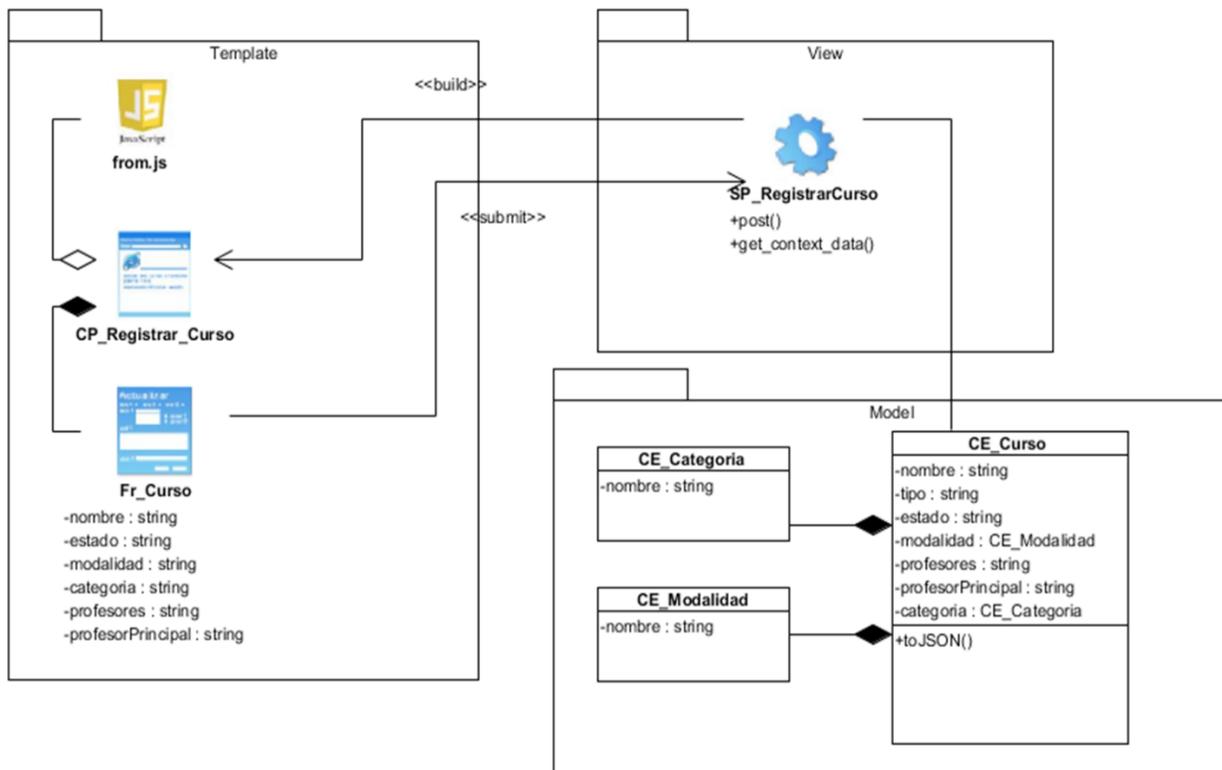
**Fuente:**[CITATION Yur22 \I 3082 ].

Una aplicación Django incluye un archivo principal urls.py que apunta a las direcciones de las URLs de cada aplicación en el proyecto. También contiene un archivo llamado settings.py donde se establecen todas las configuraciones del sistema, como la seguridad, la autenticación, la conexión al servidor de base de datos, los nombres de los módulos o aplicaciones del sistema y otras configuraciones necesarias. Cada módulo de las aplicaciones tiene su propio archivo urls.py que declara las URLs correspondientes a ese módulo. El archivo views.py controla la lógica de negocio y contiene las vistas basadas en clases o funciones que procesan los datos de los modelos y los envían a las plantillas. El archivo model.py contiene los modelos del módulo o aplicación, y, por último, la carpeta templates

contiene las plantillas HTML relacionadas con cada módulo del sistema. Esta carpeta, junto con la carpeta statics (que contiene los archivos CSS y JS de la aplicación), definen la apariencia de la arquitectura.

### II.5 Diagrama de clases del diseño

El diagrama de clases del diseño es una representación gráfica de las clases, interfaces y sus relaciones en un sistema de software. Muestra la estructura estática del sistema, incluyendo las clases, atributos, métodos y relaciones entre ellas. Son útiles para comunicar y comprender la estructura de un sistema, identificar las clases y sus responsabilidades, y definir las relaciones entre ellas. También son importantes para el diseño orientado a objetos y la implementación de sistemas de software [ CITATION Eri94 \l 3082 ]. En el siguiente diagrama de clases del diseño con estereotipos web se muestran las clases involucradas en el requisito “Registrar Curso”. Los restantes diagramas pueden encontrarse en el expediente del proyecto.



**Figura 3.** Diagrama de clases del diseño del RF\_Registrar Curso.

**Fuente:** elaboración propia.

Los patrones de diseño son soluciones probadas y comprobadas para problemas comunes en el diseño de software. Estos patrones proporcionan una forma estructurada de abordar y resolver problemas recurrentes, lo que permite a los desarrolladores crear software flexible, reutilizable y fácil de mantener. Hay patrones que abarcan las distintas etapas del desarrollo, entre ellos los patrones GRASP (General Responsibility Assignment Software Patterns o Patrones Generales para Asignar Responsabilidades) y GoF (Gang of Four o Banda de los Cuatro) [ CITATION Rus07 \l 3082 ].

### II.6.1 Patrones Generales de Software para Asignación de Responsabilidades

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP por sus siglas en inglés de *General Responsibility Assignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. En el desarrollo de software, existen varios patrones generales que se utilizan para asignar responsabilidades a las diferentes partes de un sistema. Estos patrones ayudan a organizar y estructurar el código de manera eficiente y modular [ CITATION Rob00 \l 3082 ].

Django aprovecha varios patrones de diseño que vienen incorporados por defecto en su arquitectura y, además, está diseñado de manera que fomenta su aplicación por parte del programador. Algunos de los patrones de diseño utilizados en Django incluyen:

- **Bajo acoplamiento:** es un principio de diseño de software que busca reducir la dependencia entre los componentes de un sistema. Favorece el modularidad, la flexibilidad y la escalabilidad del sistema. Además, promueve buenas prácticas de programación [ CITATION Pér20 \l 3082 ].

Dentro del marco de trabajo, se puede observar este patrón en las clases que implementan la lógica del negocio y el acceso a datos, ubicadas en el paquete Modelo de la arquitectura. Estas clases no están asociadas directamente con las clases del paquete Vistas, lo que resulta en una baja dependencia entre ellas. Esta baja dependencia permite una mayor reutilización de código entre las clases mencionadas. Por lo tanto, se puede afirmar que este patrón se evidencia en el marco de trabajo.

- **Alta Cohesión:** es clave para el éxito de un grupo o equipo, ya que promueve la colaboración, la comunicación efectiva y la motivación de los miembros para trabajar juntos hacia un objetivo común.

Una de las características clave del marco de trabajo Django es su enfoque en la organización estructurada de proyectos. Esto permite que el software sea flexible y pueda adaptarse a cambios importantes con un impacto mínimo. Este patrón se utiliza al asignar responsabilidades

específicas a cada clase, asegurando que estén estrechamente relacionadas entre sí y evitando que realicen tareas excesivas[ CITATION Cra03 \l 3082 ].

- **Experto:** este patrón se basa en el principio de responsabilidad única, que establece que una clase debe tener una única razón para cambiar. Al asignar responsabilidades a las clases adecuadas, se evita la duplicación de código y se mejora la cohesión y el acoplamiento en el sistema[ CITATION Eri04 \l 3082 ]. Este patrón se aplica en todas las clases porque cada una de ellas es especialista en proporcionar información exclusiva que solo ella posee, lo que facilita la comprensión, ampliación y mantenimiento del sistema.
- **Creador:** se compone principalmente de dos elementos: una clase creadora y una clase producto. La clase creadora es responsable de crear instancias de la clase producto y puede tener métodos adicionales para configurar o manipular el objeto creado. La clase producto es la clase que se crea y se devuelve al cliente[ CITATION Eri041 \l 3082 ]. El patrón se puede observar en las clases controladoras que, para cada uno de los módulos de configuración básica, se encargan de instanciar los objetos que gestionan, lo que promueve la reutilización y la baja dependencia, tal como se describe en el paquete Model de la arquitectura, representado en la **Figura 3**.
- **Controlador:** es un patrón arquitectónico utilizado en el desarrollo de software para separar la lógica de negocio (modelo), la interfaz de usuario (vista) y la interacción entre ambos (controlador). En resumen, se utiliza para estructurar y organizar el código de una aplicación, separando la lógica de negocio de la interfaz de usuario y permitiendo una mayor flexibilidad y mantenibilidad del software[ CITATION Eri04 \l 3082 ]. Este patrón se evidencia en la clase RegistrarCurso, que se encuentra dentro del paquete Views.
- 

### II.6.2 Patrones GoF

Los patrones GoF, también conocidos como patrones de diseño Gang of Four, se refieren a un conjunto de patrones de diseño de software desarrollados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en su libro "*Design Patterns: Elements of Reusable Object-Oriented Software*" (Patrones de diseño: elementos de software orientado a objetos reutilizable). Estos patrones proporcionan soluciones probadas y comunes a problemas recurrentes en el diseño de software[ CITATION Gam94 \l 3082 ]. A continuación, se describen algunos de los patrones utilizados por Django en su arquitectura, los cuales también fueron empleados en la propuesta de solución:

- **Iterador:** permite manejar de manera eficiente grandes conjuntos de resultados de consultas sin tener que cargar todos los resultados en la memoria al mismo tiempo. En su lugar, los resultados

se van cargando de forma perezosa a medida que se van necesitando, lo que puede mejorar considerablemente el rendimiento y la eficiencia de las consultas en Django [ CITATION Wil19 \l 3082 ]. Este patrón se evidencia en múltiples casos dentro del software, por ejemplo; en la plantilla de visualización de listado de los cursos, se recorre el diccionario con la información básica de los cursos para su vinculación con las estructuras HTML y obtener un resultado completo para este fin.

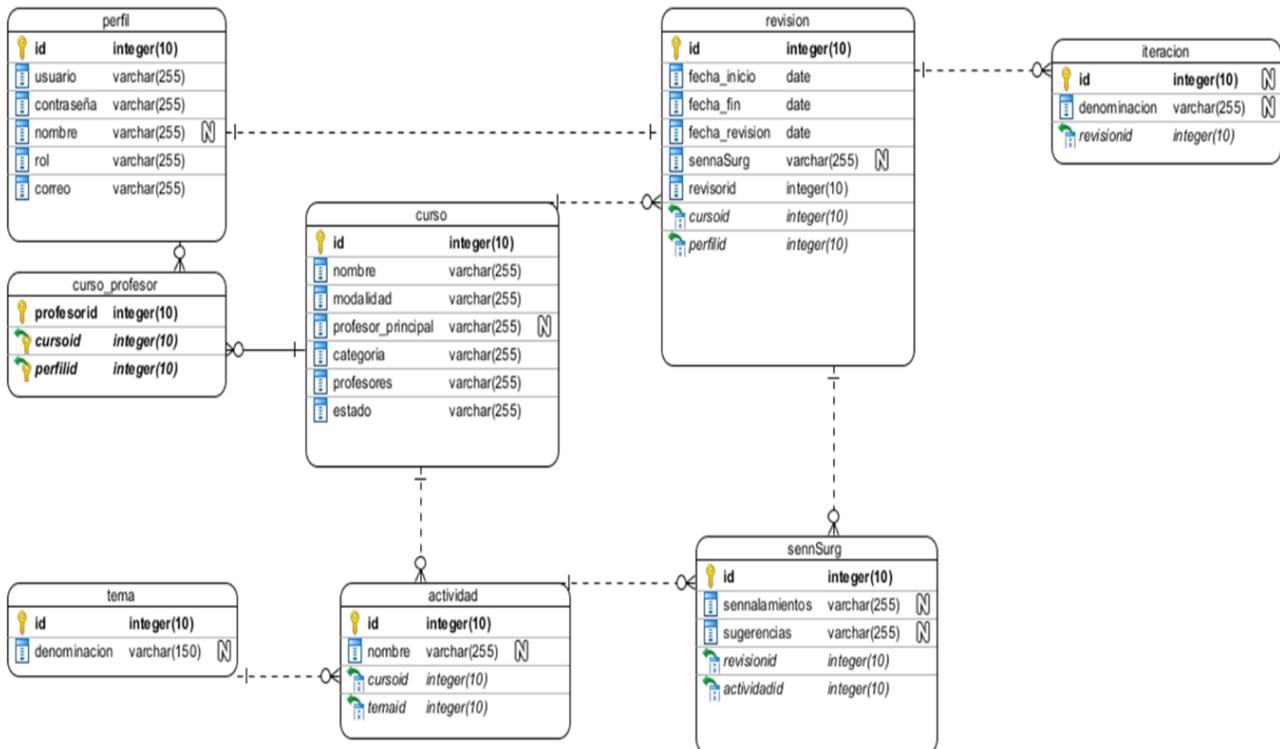
- **Decorador:** busca agregar responsabilidades adicionales a un objeto de forma dinámica, ofreciendo una alternativa flexible a la especialización a través de la herencia al momento de añadir funcionalidades [ CITATION Cra03 \l 3082 ]. El uso de decoradores en Django para la especificación de clases proporciona una manera conveniente y legible de agregar funcionalidad adicional a las clases sin modificar directamente su código. Al utilizar decoradores en la especificación de clases en Django, se puede separar la lógica principal de la clase de la lógica adicional que se desea agregar. Esto facilita la organización del código y mejora la legibilidad, ya que cada decorador se puede colocar en la parte superior de la clase correspondiente, indicando claramente qué funcionalidad se está agregando.
- **Método plantilla:** se utiliza para definir un esqueleto de un algoritmo en una clase base, pero permite que las subclasses proporcionen la implementación de ciertos pasos del algoritmo. Al utilizar el método de plantilla, se puede reutilizar el código común en la clase base y solo implementar los pasos específicos en las subclasses. Esto puede ayudar a mejorar la mantenibilidad y extensibilidad del código, ya que los cambios en el algoritmo solo requieren modificaciones en la clase base o en las subclasses, en lugar de tener que modificar todo el código [ CITATION Cra03 \l 3082 ]. Se evidencia en el paquete Template descrito en la arquitectura utilizada para el desarrollo de la solución propuesta, aterrizada en la Figura 3 a través de la clase CP\_Registrar\_Curso.
- 

### II.7 Modelo de datos

Un modelo de datos es una representación estructurada de los datos que se utilizan en un sistema o aplicación. Ayuda a organizar y describir la información de una manera lógica y coherente. Cada modelo tiene su propia estructura y reglas para representar los datos y las relaciones entre ellos. El modelo de datos adecuado a utilizar depende de las necesidades y requisitos específicos del sistema o aplicación [ CITATION Dat19 \l 3082 ].

## CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

En Django, el modelo de datos se utiliza para definir la estructura y las relaciones de las tablas en la base de datos. Para ello, se definen clases en Python que heredan de la clase base `models.Model`. Estas clases representan las tablas en la base de datos y los atributos de las clases representan las columnas de las tablas. Django se encarga de crear las tablas en la base de datos y proporciona una interfaz para interactuar con los datos [CITATION Jam \l 3082 ]. A continuación, se muestra el modelo de datos correspondiente a la solución propuesta.



**Figura 4.** Modelo de datos.

**Fuente:** elaboración propia.

El modelo de datos propuesto para la solución consta de un total de 8 tablas, de las cuales 2 son las principales para almacenar la información relacionada con la evaluación y revisión de la calidad de los cursos virtuales ellas son curso y revisión. Estas tablas son fundamentales para agilizar el proceso y garantizar la eficiencia en la gestión de la calidad de los cursos virtuales.

### II.8 Diagrama de componentes

Una vez finalizado el diseño y el análisis de la propuesta de solución, se pasa a diseñar la implementación de dicha propuesta a través de un diseño a nivel de componentes. En esta etapa, se establece

la estructura general de los datos y del programa del software. El objetivo principal es traducir el modelo de diseño en un software operativo [ CITATION new \l 3082 ]. En resumen, el diseño a nivel de componentes es una etapa crucial en el proceso de desarrollo de software, ya que permite traducir el diseño en una implementación operativa. Es importante realizar un diseño detallado y coherente, teniendo en cuenta los requerimientos y las restricciones del sistema.

Los componentes son elementos autocontenidos y encapsulados que pueden encapsular estructuras de cualquier grado de complejidad. Se comunican con otros componentes a través de interfaces y pueden utilizar las interfaces de otros componentes. Esto promueve la reutilización, modularidad y bajo acoplamiento en el desarrollo de sistemas [ CITATION Eri94 \l 3082 ]. Para una mejor comprensión, a continuación, se describe el diagrama de componentes relacionado con la implementación de la HU\_11 Registrar curso.

### II.9 Diagrama de despliegue

Los diagramas de despliegue son una poderosa herramienta para modelar la estructura de hardware y software de un sistema en un entorno de implementación, lo que permite una comprensión clara de cómo se distribuyen y comunican los componentes del sistema [ CITATION Mar04 \l 3082 ].

La Figura muestra el diagrama de despliegue de la propuesta de solución, el cual ilustra cómo y dónde se implementa el sistema. Los nodos representan las máquinas físicas y los procesadores, mientras que los artefactos embebidos o los nodos pueden representar la construcción interna. Los estereotipos se utilizan para indicar la naturaleza del equipo, como dispositivos, procesadores y memoria. Los elementos principales de este diagrama son los nodos (representados como prismas), los componentes (representados como cajas rectangulares con protuberancias en el lado izquierdo) y las asociaciones. A continuación, se presenta el entorno donde se debe implementar la solución.

### Conclusiones del capítulo

En este capítulo se presentó la solución propuesta partiendo de las disciplinas de requisitos, análisis y diseño que se definen en la metodología que guía la presente investigación, donde:

- La descripción general de la propuesta de solución brindó una visión clara de los flujos de información y las actividades que abarcan el negocio.
- La documentación de la especificación de los requisitos e historias de usuarios permitió registrar de manera completa toda la información relacionada con la solución propuesta.

## **CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO**

- La definición clara de la arquitectura y la selección adecuada de los patrones de diseño, sentaron las bases para promover la reutilización de código y la adopción de buenas prácticas de programación por parte de los desarrolladores durante la etapa de implementación.
- El modelo de datos permitió describir tanto la estructura lógica como la estructura física de la información persistente que es gestionada por el sistema que se propone para la gestión de los indicadores de calidad de los cursos virtuales en el CENED.
- El diagrama de componentes permitió identificar los elementos necesarios para mostrar la vista requerida por el usuario, así como visualizar las dependencias y la integración entre estos componentes durante la implementación de la solución propuesta.
- El diagrama de despliegue permitió modelar la disposición física del sistema, mostrando el hardware utilizado, los componentes instalados en dicho hardware y las conexiones físicas entre ellos.

En este capítulo se presentan los resultados obtenidos durante la validación del sistema propuesto para gestionar los indicadores de calidad a utilizar durante la revisión y evaluación de los cursos del aula virtual del Centro Nacional de Educación a Distancia. En primer lugar, se aplican técnicas de validación de requisitos para asegurar que el sistema cumple con las necesidades y expectativas establecidas. Posteriormente, se lleva a cabo la validación del diseño utilizando las métricas: Tamaño Operacional de las Clases y Relación entre Clases. Una vez validado el diseño, se procede a verificar el correcto funcionamiento de la solución propuesta a través de pruebas de software, con el fin de garantizar la calidad del sistema, asegurando que cumpla con los requerimientos funcionales y no funcionales establecidos. Por último, se verificó el cumplimiento de la relación causa-efecto entre la variable independiente y las variables dependientes de la investigación.

### III.1 Técnicas de validación de requisitos

Con el objetivo de confirmar que los requisitos del software capturan las necesidades del cliente y definen el sistema deseado, se realizó un proceso de validación de estos requisitos. Para llevar a cabo esta validación, se emplearon las siguientes técnicas:

#### III.1.1 Revisión técnica formal

El primer mecanismo utilizado para la validación de los requisitos es la revisión técnica formal (RTF). En esta revisión, se involucra un equipo compuesto por ingenieros del sistema, clientes, usuarios y otros interesados relevantes. El objetivo de esta revisión es examinar detalladamente la especificación del sistema en busca de posibles errores en el contenido o la interpretación, áreas que requieran aclaraciones, información incompleta, requisitos contradictorios o requisitos que sean imposibles o inalcanzables[ CITATION new \l 3082 ].

Durante la RTF, el equipo de revisión se dedica a analizar minuciosamente la especificación del sistema, buscando cualquier aspecto que pueda comprometer la calidad o el éxito del proyecto. Se busca asegurar que los requisitos sean claros, comprensibles y factibles de implementar. Además, se busca detectar y corregir cualquier error o conflicto en los requisitos identificados[CITATION Som112 \l 3082 ]. Para llevar a cabo este proceso de validación, se utilizó la métrica de Calidad de la especificación, la cual se describe a continuación.

**Métrica Calidad de la especificación:** es una técnica que ampliamente utilizada en la ingeniería de requisitos para evaluar la calidad de la documentación de los requisitos del sistema. Esta métrica se

basa en diferentes criterios y dimensiones de calidad, como la claridad, la completitud, la consistencia y la verificabilidad de la especificación. Se refiere a la comprensibilidad y legibilidad de los requisitos, asegurando que estén expresados de manera clara y sin ambigüedades [ CITATION new \l 3082 ]. Para alcanzar este objetivo, es necesario calcular inicialmente el total de requisitos de software, como se muestra a continuación:

$Nr$ : el total de requisitos de especificación.

$Nf$ : cantidad de requisitos funcionales.

$Nnf$ : cantidad de requisitos no funcionales.

$$Nr = Nf + Nn$$

Como resultado de la sustitución de los valores, se obtiene:

$$Nr = 20 + 18$$

$$Nr = 38$$

Para determinar la Especificidad de los Requisitos (ER) o la ausencia de ambigüedad en los mismos, se utiliza una fórmula que consiste en dividir el número de requisitos específicos entre el total de requisitos identificado:

$$ER = \frac{Nui}{Nr}$$

Donde  $Nui$  representa el número de requisitos en los cuales todos los revisores tuvieron interpretaciones idénticas. Cuanto más cercano a 1 sea el valor de  $ER$ , menor será la ambigüedad presente en los requisitos.

En el caso de los requisitos obtenidos, se identificaron 4 requisitos que generaron contradicciones en las interpretaciones. Sustituyendo las variables en la fórmula, se obtiene el siguiente resultado:

$$ER = \frac{34}{38}$$

$$ER = 0,89$$

El resultado final es satisfactorio, lo que indica que el grado de ambigüedad de los requisitos es bajo, representando solo el 11% del total. Por lo tanto, el 89% restante de los requisitos es fácilmente entendible y no presenta ambigüedades. Los requisitos que fueron identificados como ambiguos han sido modificados y validados para garantizar una correcta interpretación. Este proceso de revisión y mejora ha permitido eliminar cualquier ambigüedad presente en los requisitos, asegurando que sean claros y comprensibles para todos los involucrados en el proyecto. Esta acción es crucial para evitar malentendidos y problemas durante el desarrollo del sistema. Al tener requisitos bien definidos y sin

ambigüedades, se facilita la comunicación entre los miembros del equipo y se reduce el riesgo de errores o interpretaciones incorrectas.

### **III.2 Validación del diseño**

La validación del diseño es un proceso crucial para garantizar que el diseño de un sistema cumpla con los requisitos y expectativas establecidas. Consiste en verificar y confirmar que el diseño propuesto es correcto, funcional y adecuado para cumplir con los objetivos del proyecto[ CITATION new \l 3082 ].

Un elemento clave de cualquier proceso de ingeniería es la medición. Las medidas se utilizan para comprender mejor los atributos de los modelos creados y para evaluar la calidad de los productos o sistemas sometidos a ingeniería que se construyen. La validación del diseño a través de métricas permite medir de forma cuantitativa la calidad de los atributos internos del software, lo cual puede ayudar al desarrollador a evaluar la calidad de un diseño a nivel de componente[ CITATION new \l 3082 ]. Para validar el diseño que se obtiene en la presente investigación, se decide aplicar dos métricas que se describen a continuación.

#### **III.2.1 Métricas Tamaño Operacional de las Clases**

Las métricas basadas en el Tamaño Operacional de las Clases (TOC) se centran en contar la cantidad de atributos y operaciones que tiene una clase individual, así como el promedio que presenta el sistema en su conjunto. Estas métricas ayudan a evaluar diferentes atributos de calidad de las clases, como[ CITATION new \l 3082 ]:

- Responsabilidad se refiere a la cantidad de atributos y operaciones que una clase tiene asignados.
- Complejidad de implementación se relaciona con la cantidad y la complejidad de las operaciones que una clase realiza.
- Reutilización se refiere a la capacidad de una clase para ser utilizada en diferentes contextos o en diferentes partes del sistema.

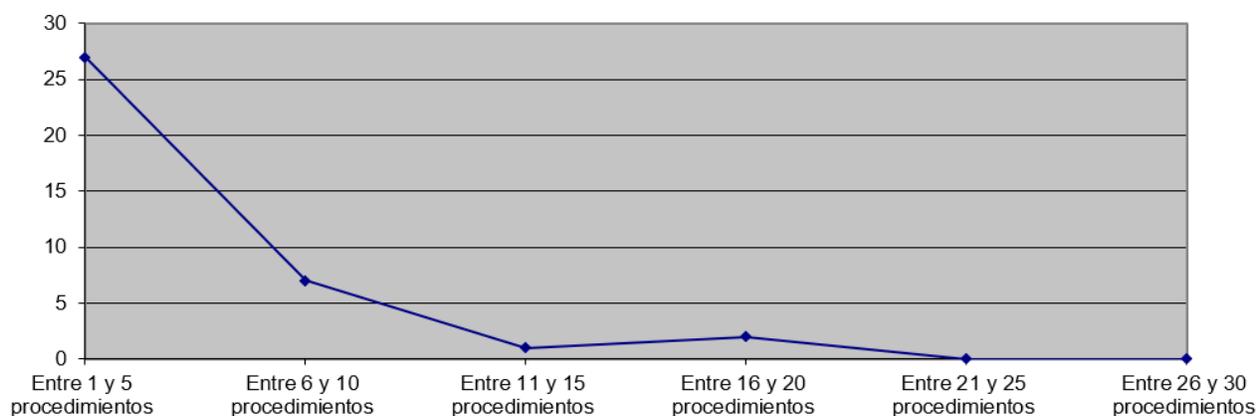
Después de examinar el indicador de dimensión de clase, si el valor resultante se inclina hacia el incremento, es posible que la clase tenga una amplia carga de responsabilidades. Esto implica que la reutilización sea mínima y la implementación sea altamente complicada. Para evaluar los atributos de calidad, se han establecido los siguientes límites que se muestran en la Tabla 5, en la misma se utilizan las siglas Prom (promedio):

**Tabla 5.** Métrica TOC. Categoría por atributos y criterio de evaluación.

| Atributo de calidad           | Categoría | Criterio                    |
|-------------------------------|-----------|-----------------------------|
| Responsabilidad               | Baja      | TOC $\leq$ Promedio (Prom). |
|                               | Media     | TOC Entre Prom. y 2* Prom.  |
|                               | Alta      | TOC $>2*$ Prom.             |
| Complejidad de implementación | Baja      | TOC $\leq$ Prom.            |
|                               | Media     | Entre Prom. y 2* Prom.      |
|                               | Alta      | TOC $>2*$ Prom.             |
| Reutilización                 | Baja      | TOC $>2*$ Prom.             |
|                               | Media     | TOC Entre Prom. y 2* Prom.  |
|                               | Alta      | TOC $\leq$ Prom.            |

**Fuente:** [ CITATION Lor94 \l 3082 ].

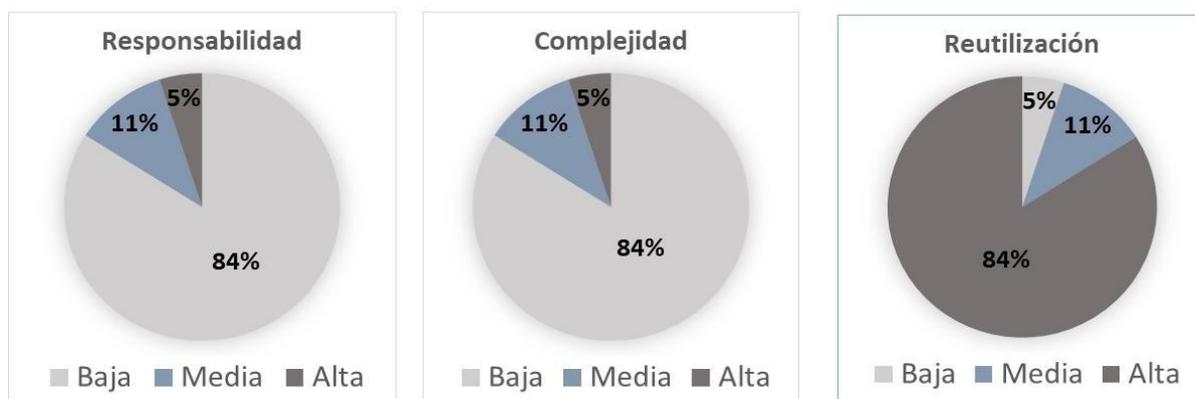
Después de evaluar cada una de estas categorías en el diseño final de la solución propuesta, se obtuvieron los siguientes resultados, en términos del número de métodos de cada una de las clases del diseño:



**Figura 5.** Representación de la cantidad de clases por cantidad de procedimientos.

**Fuente:** elaboración propia.

De acuerdo con la imagen anterior, se puede observar que veintisiete (27) de todas las clases tienen entre 1 y 5 procedimientos, mientras que siete (7) clases tienen entre 6 y 10 procedimientos. Además, una clase tiene entre 11 y 15 procedimientos, y, por último, dos (2) clases tienen entre 16 y 20 procedimientos. Después de analizar estos resultados relacionados con la métrica TOC, se obtuvieron los siguientes indicadores para cada atributo de calidad:



**Figura 6.** Resultados de la métrica TOC.

**Fuente:** elaboración propia.

Tras analizar los datos recopilados en relación a los atributos de calidad evaluados por la métrica TOC, se obtuvieron los siguientes resultados:

- El 84% de las clases presentan una responsabilidad y complejidad de implementación baja.
- El 84% de las clases tienen una reutilización elevada.

Estos resultados evidencian la excelencia del diseño de la solución propuesta, ya que, al lograr índices reducidos de responsabilidad y complejidad, junto con un nivel elevado de reutilización de las clases, se simplifica considerablemente la implementación de cada una de las clases del diseño, como resultado de la solución propuesta.

### III.2.2 Métricas Relación entre Clases

El resultado de la aplicación de la métrica Relación entre Clases (RC) se determina mediante el conteo de las conexiones de uso que se establecen entre una clase y las demás clases presentes. Esta métrica evalúa los atributos de calidad siguientes [ CITATION Som112 \l 3082 ]:

- Reutilización: un incremento en el RC conlleva a una reducción en el grado de reaprovechamiento de la clase.
- Cantidad de Pruebas: un aumento en el RC implica un incremento en la cantidad de pruebas necesarias para evaluar una clase.
- Acoplamiento: un aumento en el RC resulta en un aumento en el nivel de interdependencia de la clase.
- Complejidad de Mantenimiento: un incremento en el RC implica un aumento en la dificultad de mantener la clase.

Para alcanzar un nivel óptimo de interconexión entre clases, el resultado obtenido al utilizar esta métrica debe estar en consonancia con el grado de dependencia y la dificultad de mantenimiento de las clases. Además, debe ser inversamente proporcional al grado de reutilización del código. Para aplicar la métrica RC, es esencial clasificar cada una de las clases en función de la cantidad de relaciones que posean.

Es importante que el valor obtenido al aplicar la métrica RC esté directamente relacionado con el acoplamiento y la complejidad de mantenimiento, ya que esto indica el nivel de interdependencia y dificultad asociada a las clases. Si el valor de RC es alto, implica que las clases tienen muchas relaciones de uso, lo cual puede aumentar el grado de acoplamiento y la complejidad del mantenimiento. Por otro lado, si el valor de RC es bajo, indica que las clases tienen menos relaciones y, por lo tanto, menor acoplamiento y complejidad. Para aplicar la métrica RC, es necesario categorizar las clases según la cantidad de relaciones que tienen, lo cual permite obtener una medida precisa y clasificarlas de manera adecuada. A continuación, se presenta una tabla con las categorías para clasificar cada uno de los atributos de calidad mencionados previamente, junto con el criterio de evaluación correspondiente. En la tabla 6 se utiliza la abreviatura Prom (promedio).

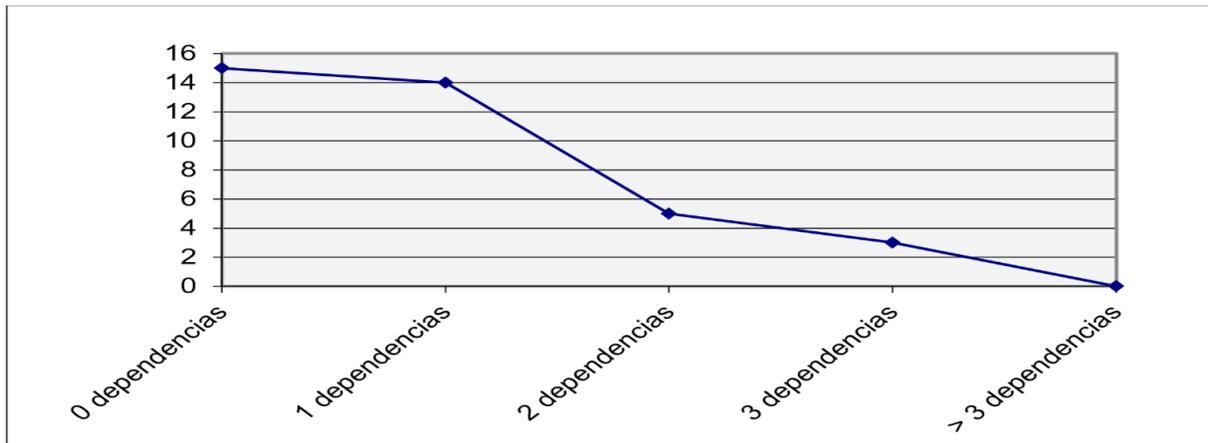
**Tabla 6.** Métrica RC. Categoría por atributos y criterio de evaluación.

| Atributo                     | Categoría | Criterio              |
|------------------------------|-----------|-----------------------|
| Acoplamiento                 | Ningun    | RC = 0                |
|                              | Bajo      | RC = 1                |
|                              | Medio     | RC = 2                |
|                              | Alto      | RC > 2                |
| Complejidad de Mantenimiento | Baja      | RC <= Prom            |
|                              | Media     | Prom <= RC <= 2* Prom |

|                     |       |                                   |
|---------------------|-------|-----------------------------------|
| Reutilización       | Alto  | $RC > 2 * Prom$                   |
|                     | Baja  | $RC > 2 * Prom$                   |
|                     | Media | $Promedio < RC \leq 2 * Promedio$ |
| Cantidad de pruebas | Alto  | $RC \leq Promedio$                |
|                     | Baja  | $RC \leq Promedio$                |
|                     | Media | $Promedio \leq RC < 2 * Promedio$ |
|                     | Alto  | $RC \geq 2 * Promedio$            |

Fuente[ CITATION Lor94 \l 3082 ]:

Después de examinar cuidadosamente cada una de estas categorías en el diseño final de la propuesta de solución, se obtuvieron los siguientes resultados, basados en las interacciones de uso de cada una de las clases del diseño:

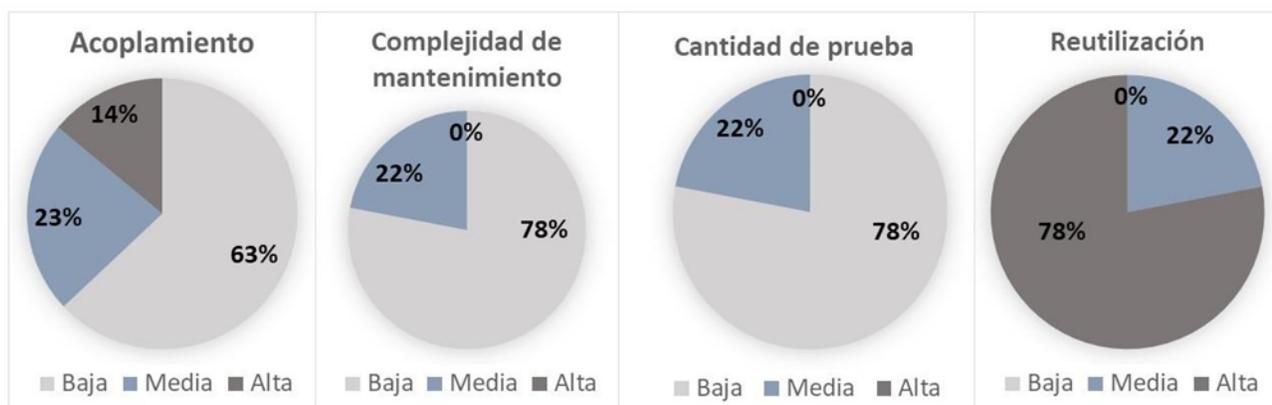


**Figura 7.** Representación de la cantidad de clases por cantidad de dependencias.

**Fuente:** elaboración propia.

Siguiendo el análisis detallado de cada una de estas categorías en el diseño final de la propuesta de solución, se observa que, en la figura anterior, quince (15) de las clases están libres de

dependencias, catorce (14) tienen solo una dependencia, cinco (5) clases cuentan con dos dependencias y, por último, tres (3) de las clases tienen tres dependencias. Tras examinar estos resultados relacionados con la métrica RC, se obtuvieron los siguientes indicadores para cada atributo de calidad:



**Figura 8.** Resultados de la métrica RC.

**Fuente:** elaboración propia.

Tras analizar los resultados de cada una de las categorías en el diseño propuesto, se interpreta lo siguiente:

- El 63% de las clases presentan un acoplamiento reducido.
- El 78% de las clases tienen una complejidad de mantenimiento baja y una cantidad mínima de pruebas.
- El 78% de las clases muestran una alta capacidad de reutilización.

Como se puede observar en la figura, el grado de acoplamiento presenta un nivel reducido, lo que implica que hay escasa dependencia entre las clases, lo cual conlleva a una elevada reutilización. Además, se puede apreciar que hay una baja complejidad de mantenimiento, lo que significa que no es necesario llevar a cabo un gran número de pruebas y optimizaciones en los métodos y demás operaciones. Esto a su vez reduce el tiempo necesario para implementar y realizar pruebas a las funcionalidades que engloban el sistema que se propone para gestionar los indicadores de calidad a utilizar durante la revisión y evaluación de los cursos en el aula virtual del CENED.

Las evaluaciones son un medio apropiado para determinar el estado de la excelencia de un producto en las cuales un sistema o componente es ejecutado bajo condiciones o requisitos específicos. Ofreciendo la obtención de un producto de mayor excelencia, ya que las fallas son detectadas en un corto período de tiempo y se corrigen de manera más sencilla[ CITATION new \l 3082 ].

Para la validación de la solución propuesta como resultado de la presente investigación se establece una estrategia de pruebas de software a partir de las disciplinas establecidas para el desarrollo de las pruebas por la metodología AUP\_UCI. En la estrategia se determina aplicar la disciplina de Pruebas internas a nivel de unidad, realizando pruebas unitarias y funcionales. Para el caso de las pruebas unitarias se evalúan las características internas utilizando el método de Caja blanca mediante la técnica de Ruta básica. Para llevar a cabo las pruebas funcionales se evalúan las características externas mediante técnicas del método de Caja negra.

#### III.3.1 Pruebas unitarias

Las pruebas unitarias son una práctica común en el desarrollo de software que se enfoca en verificar las funcionalidades internas de un módulo, incluyendo los caminos lógicos, ciclos y condiciones programadas. Estas pruebas se consideran fundamentales para garantizar la calidad del software y se recomienda que sean realizadas por los propios desarrolladores[ CITATION new \l 3082 ]. Para implementar las pruebas unitarias, los autores de esta investigación han decidido utilizar el enfoque de Caja Blanca. Este método, según [ CITATION new \l 3082 ], permite desarrollar casos de prueba que aseguren la ejecución, al menos una vez, de todos los caminos independientes dentro del código fuente.

##### **Método de caja blanca:**

Este método, también conocido como prueba estructural o de caja cristal, implica el análisis interno del código fuente y se basa en la comprensión de cómo están organizadas las estructuras y rutas del programa. Al conocer la estructura interna, es posible identificar los caminos independientes y diseñar casos de prueba que los cubran adecuadamente[ CITATION new \l 3082 ].

Se aplica a las funcionalidades internas del sistema con el objetivo de evaluar la complejidad lógica de un diseño procedimental y utilizar esta medida como guía para definir un conjunto

básico de caminos de ejecución. Esta prueba garantiza que cada sentencia del programa se ejecute al menos una vez en los casos de prueba obtenidos a través de la ruta básica. Para generar el grafo de flujo, se descomponen las condiciones compuestas en condiciones sencillas[ CITATION new \l 3082 ]. Para ejecutar este método, se decide utilizar la técnica de ruta básica, la cual se describe a continuación.

**Técnica de ruta básica:** esta técnica es ampliamente utilizada en el enfoque de pruebas de caja blanca para evaluar la efectividad de los métodos asociados a una clase. Su objetivo principal es garantizar que cada camino independiente dentro de un componente o programa sea ejecutado al menos una vez, lo que proporciona una medida de la complejidad lógica del diseño. Esta técnica de prueba es esencial para garantizar la calidad y el funcionamiento óptimo de un sistema, ya que permite identificar posibles errores o fallos en la implementación de los caminos de ejecución. Al asegurarse de que cada camino sea probado exhaustivamente, se puede garantizar una mayor cobertura de pruebas y una mayor confiabilidad del sistema[ CITATION new \l 3082 ]. Para obtener los casos de prueba utilizando la técnica de ruta primaria, es necesario crear el diagrama de flujo para cada uno de los métodos desarrollados. A continuación, se presenta un ejemplo de la aplicación de esta técnica al código.

```

def post(self, request, **kwargs):
    rpk = kwargs['rpk']
    ipk = int(kwargs['ipk'])
    form = IdicadorForm(request.POST)

    if form.is_valid():
        revision = Revision.objects.get(pk=rpk)
        indicador = Indicador.objects.filter(revision__pk=revision.pk).filter(iteracion=ipk).first()

        if not indicador:
            Indicador.objects.create(
                revision = revision,
                iteracion = ipk,
                a = form.cleaned_data.get('a', ''),
                b = form.cleaned_data.get('b', ''),
                c = form.cleaned_data.get('c', ''),
                d = form.cleaned_data.get('d', ''),
            )
        else:
            indicador.a = form.cleaned_data.get('a', '')
            indicador.b = form.cleaned_data.get('b', '')
            indicador.c = form.cleaned_data.get('c', '')
            indicador.d = form.cleaned_data.get('d', '')
            indicador.save()

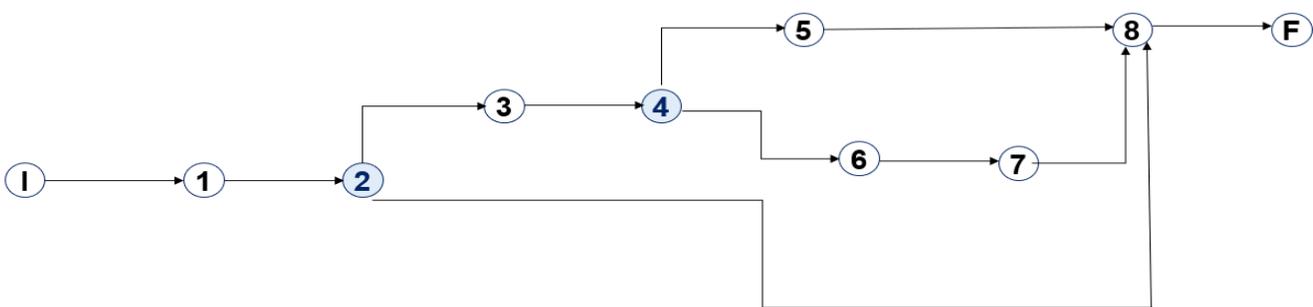
    return redirect('/')

```

**Figura 9.** Método utilizado como ejemplo para la técnica de ruta básica.

**Fuente:** elaboración propia.

A continuación, se presenta el diagrama de flujo obtenido utilizando el método previamente mencionado:



**Figura 10.** Grafo del flujo del método Registrar Revisión.

**Fuente:** elaboración propia.

Posteriormente se calcula la complejidad ciclomática  $V(G)$  del grafo resultante, la cual es un indicador del número de rutas independientes que existen en un grafo, es decir, cualquier ruta dentro del código que introduce al menos un nuevo conjunto de sentencias de proceso o una nueva condición. La complejidad ciclomática se determina mediante las siguientes alternativas:

1.  $V(G) = R$
2.  $V(G) = E - N + 2$
3.  $V(G) = P + 1$

Conociendo que:

- **G**: Grafo de flujo (grafo)
- **R**: El número de regiones contribuye a estimar el valor de la complejidad ciclomática.
- **E**: Número de aristas
- **V(G)**: Complejidad ciclomática
- **N**: Número de nodos del grafo
- **P**: Número de nodos predicados (nodos de donde parten al menos dos aristas).

Realizando los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

$$V(G) = R = 3$$

$$V(G) = E - N + 2 = 11 - 10 + 2 = 3$$

$$V(G) = P + 1 = 2 + 1 = 3$$

Una vez determinada la complejidad ciclomática, el resultado obtenido indica el máximo número de pruebas que deben ser realizadas. En consecuencia, los conjuntos de caminos fundamentales son:

Camino 1: nodo1 - nodo2 - nodo8

Camino 2: nodo1 - nodo2 - nodo3 - nodo4 - nodo5 - nodo8

Camino 3: nodo1 - nodo2 - nodo3 - nodo4 - nodo6 - nodo7 - nodo8

### Resultados al aplicar la técnica de ruta básica

Esta técnica se empleó en los métodos de las clases controladoras; estas clases fueron seleccionadas debido a que abarcan las funcionalidades del sistema. Django cuenta con un sistema de pruebas automatizadas con el marco de ejecución de pruebas y otras herramientas; puede simular solicitudes, insertar datos de prueba, inspeccionar la salida de la aplicación y en general verificar la corrección del código. Para validar y completar las pruebas se crearon seis clases de pruebas que

realizan verificaciones en los modelos y las vistas de la plataforma, asegurando que se haya probado el código en los principales módulos del sistema.

Se llevaron a cabo todas las pruebas de los casos de prueba seleccionados en las clases con el objetivo de identificar errores de programación. Como criterio de finalización, se consideró el resultado esperado definido en cada caso de prueba. En ninguno de los casos se encontraron inconformidades, ya que en todos los casos el resultado coincidió con el esperado. En el **Anexo 2**, se muestra el resultado para el caso de prueba de la ruta independiente.

### **III.3.2 Pruebas funcionales**

Las pruebas funcionales son pruebas planificadas basadas en los requisitos del sistema y se realizan para verificar si el software cumple con las funciones esperadas y las especificaciones funcionales. Se emplea el enfoque de Caja negra, que se centra en probar la funcionalidad del software sin conocer su implementación interna [ CITATION new \l 3082 ].

#### **Método de Caja negra:**

Este método permite al ingeniero de software obtener conjuntos de datos de entrada que prueben de manera exhaustiva todos los requisitos funcionales de un programa. Esta técnica no reemplaza las técnicas del método de caja blanca, sino que se considera un enfoque complementario que busca identificar diferentes tipos de errores en comparación con los métodos de caja blanca. En resumen, la técnica de caja negra y el método de caja blanca son enfoques complementarios que se utilizan en conjunto para obtener una cobertura de pruebas más completa y descubrir diferentes tipos de errores en el software. Estas pruebas permiten identificar [ CITATION new \l 3082 ]:

- Funcionalidades incorrectas o faltantes.
- Problemas en la interfaz.
- Errores en la estructura de datos o en el acceso a bases de datos externas.
- Problemas de rendimiento.
- Fallos en la inicialización y finalización del software.

Las pruebas funcionales, al evaluar la funcionalidad del software desde una perspectiva externa, son capaces de detectar funcionalidades incorrectas o ausentes, así como errores en la interfaz de usuario. Además, estas pruebas también pueden revelar problemas relacionados con la manipulación de estructuras de datos o el acceso a bases de datos externas. Además, las pruebas funcionales pueden ayudar a identificar problemas de rendimiento, como tiempos de respuesta lentos o consumo excesivo de recursos. Por último, estas pruebas también pueden descubrir errores en la inicialización y ter-

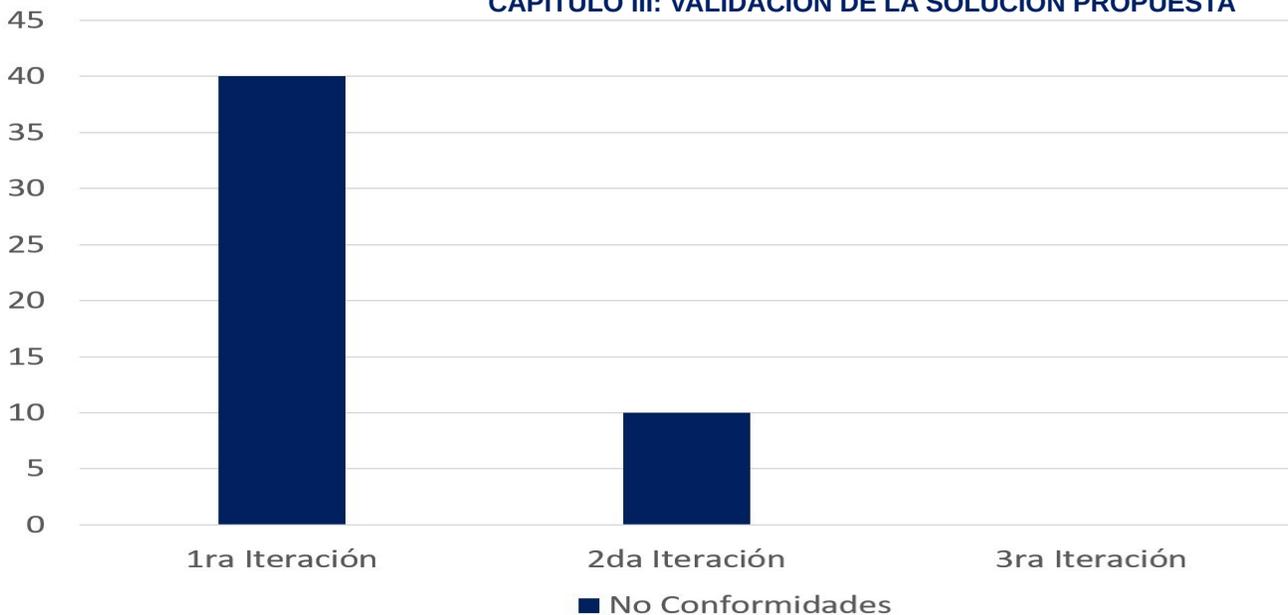
minación del software, como problemas de carga o cierre incorrecto[ CITATION Som112 \l 3082 ]. Para llevar a cabo el método de Caja negra se utiliza la técnica de Partición equivalente que a continuación se describe.

**Técnica de prueba: Partición equivalente:** esta técnica segmenta el conjunto de entrada de un programa en categorías de datos a partir de las cuales se pueden generar casos de prueba. El objetivo de este método es definir casos de prueba que puedan revelar ciertos tipos de errores, lo que a su vez reduce la cantidad total de casos de prueba que se deben desarrollar. Una clase de equivalencia representa un conjunto de condiciones de entrada válidas e inválidas. Al utilizar esta técnica, se puede seleccionar un caso de prueba representativo de cada clase de equivalencia, lo que ayuda a descubrir errores comunes asociados con esa clase en particular. Esto permite una cobertura más amplia y eficiente de las pruebas funcionales[ CITATION new \l 3082 ]. Para aplicar esta técnica, es necesario realizar previamente el Diseño de Casos de Prueba (DCP) con el objetivo de obtener un conjunto de pruebas que tengan la mayor probabilidad de detectar los defectos del software. Según [ CITATION new \l 3082 ] un DCP se define como un conjunto de condiciones o variables mediante las cuales se determina si una aplicación o característica cumple parcial o completamente con los requisitos establecidos.

Como primer paso en el DCP, se encuentra la descripción de las variables para el caso del requisito "Registrar Curso", la cual se muestra en el **Anexo 2**. Una vez que se obtiene la descripción de cada una de estas variables, se procede a generar los escenarios de prueba en el sistema. En el **Anexo 3** se presentan los escenarios correspondientes al requisito en cuestión.

Para aplicar el DCP a la solución, se efectuaron un total de 3 iteraciones para poder alcanzar resultados satisfactorios atendiendo al correcto comportamiento del sistema ante diferentes situaciones. En la **Figura 13** se muestra una gráfica con 40 No conformidades (NC) en una primera iteración y 10 en una segunda iteración y en la **Figura 14**, un resumen según su clasificación por cada iteración.

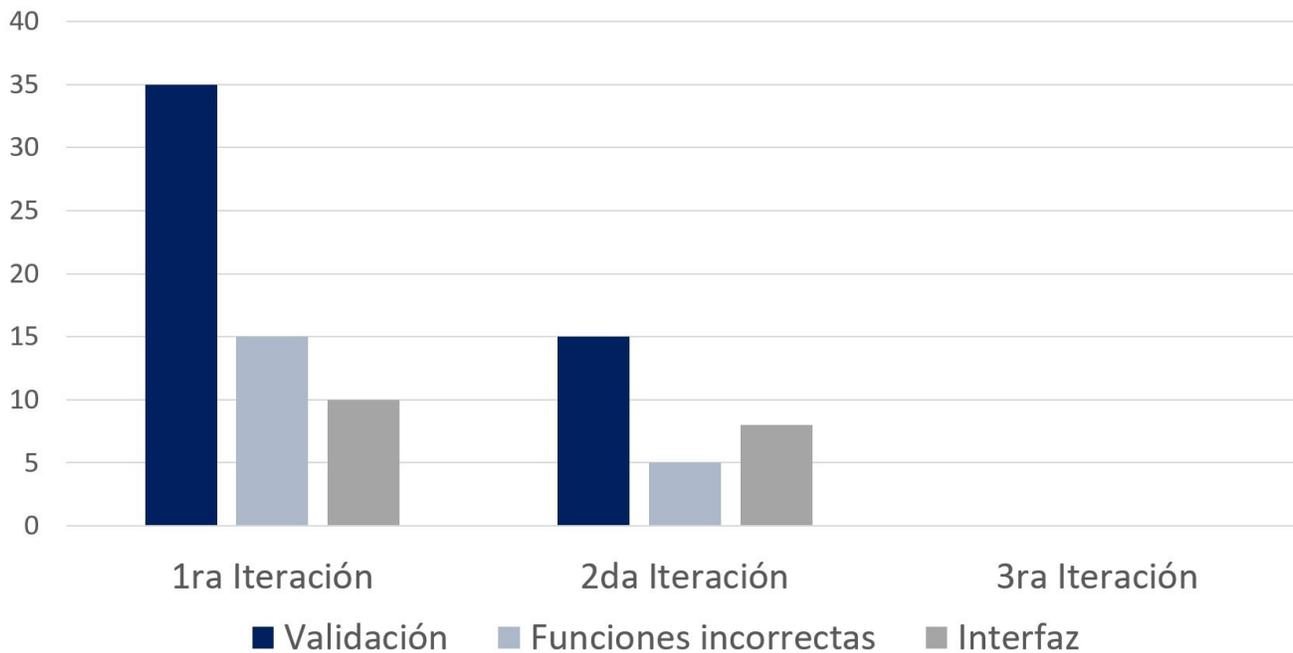
**Total de No Conformidades**  
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA



**Figura 11.** Total de NC detectadas por iteración.

*Fuente:* elaboración propia.

**Método de caja negra**



**Figura 12.** Total, de NC por tipo de clasificación.

*Fuente:* elaboración propia.

### III.4 Validación de las variables de la investigación

Considerando la idea a defender planteada en esta investigación, que busca analizar la relación causa-efecto entre la variable independiente "gestión de los indicadores de calidad" y las variables dependientes "celeridad", se propone desarrollar un instrumento con criterios para evaluar antes y después de implementar el sistema. Estos criterios se identifican a partir de las principales deficiencias encontradas en la situación problemática y de los resultados obtenidos en una encuesta realizada a los expertos en el Centro Nacional de Educación a Distancia:

- Tiempo de recopilar los datos y evaluar un curso a través de los indicadores de calidad.
- Tiempo de búsqueda de un curso y generar el informe de revisión asociada.
- Tiempo de notificación del retraso de la revisión de un curso.

En la siguiente **Tabla 7** se evalúan los distintos criterios de medición establecidos. Se realiza una evaluación previa a la implementación del sistema y otra posterior, con el objetivo de verificar cómo la solución propuesta contribuye a mejorar la rapidez del proceso de gestión de calidad de cursos virtuales en el aula virtual del Centro Nacional de Educación a Distancia. Para realizar esta comparación, se comprobó el tiempo promedio que los especialistas demoran en cada una de las actividades relacionadas con los criterios de medición. Estos datos fueron obtenidos a partir de los resultados de una encuesta realizada a los empleados del Departamento del Centro Nacional de Educación a Distancia (Ver **Anexo 5**).

Por su parte, para evaluar el tiempo de respuesta del sistema durante las condiciones de carga esperada, y así corroborar la agilización en el proceso, los autores de la presente investigación definen el empleo de la herramienta automatizada JMeter que a continuación se describe.

**JMeter:** es un software de código abierto diseñado en Java que permite realizar pruebas de rendimiento y de funcionalidad sobre aplicaciones tipo cliente/servidor escritas en cualquier lenguaje. Se puede utilizar para simular una carga pesada en un servidor, la red o un objeto para poner a prueba su resistencia o para analizar el rendimiento global en diferentes tipos de carga. Igualmente puede ser utilizado para hacer un análisis gráfico de rendimiento o para probar el comportamiento de diferentes elementos con un gran volumen de carga y concurrencia[ CITATION The18 \l 3082 ]. En el **Anexo 6** se evidencian los resultados de las pruebas realizadas para una carga de trabajo máxima de 5 y 10 usuarios. La selección de esta carga está en correspondencia con el número de usuarios a los que va dirigida la solución.

**Tabla 7.** Validación de las variables de investigación.

| Atributo  | Antes   | Después   |
|---|---|---|
| Tiempo para recopilar los datos y evaluar un curso a través de los indicadores de calidad | De acuerdo con los hallazgos de la encuesta, se concluyó que el 100% de los entrevistados plantean que demoran entre 20 y 30 minutos para llevar a cabo esta tarea. | Se reduce de manera significativa el tiempo requerido, ya que ahora solo toma aproximadamente 34 segundos para completar el registro esta operación.  |
| Tiempo de búsqueda de un curso y generar el informe de revisión asociada                  | De acuerdo con los resultados obtenidos en la encuesta, los expertos indicaron que ahora requieren entre 3 y 5 minutos para llevar a cabo esta actividad.           | Tras la implementación del sistema, se verificó que el tiempo requerido para realizar esta actividad se redujo a 31 segundos aproximadamente.   |
| Tiempo de notificación del retraso de la revisión de un curso                             | Según los resultados de la encuesta, los especialistas informaron que ahora requieren de 4 a 7 días para completar esta actividad.                                  | Mediante el uso del sistema, se constató que de manera automática se notifica al usuario, cuando el plazo de retraso de la revisión ha vencido, tanto por correo electrónico como a través del sistema. |

**Fuente:** elaboración propia.

La confrontación llevada a cabo en la tabla precedente, mediante los estándares de evaluación previamente establecidos, evidencia cómo, gracias a la adaptación lograda como consecuencia de la investigación actual, se fomenta la agilidad de la gestión de calidad de cursos virtuales en el aula virtual del Centro Nacional de Educación a Distancia.

### Conclusiones del capítulo

En este capítulo se examinó la propuesta de solución, donde:

- La implementación de técnicas de validación de requisitos permitió confirmar la concordancia de los mismos con las solicitudes del cliente.
- La validación del diseño a través de las métricas TOC y RC permitió obtener, en términos generales, el nivel de complejidad de implementación y mantenimiento, de responsabilidad, de reutilización, de acoplamiento y la cantidad de pruebas necesarias para llevar a cabo las cla-

ses, lo cual implica que las clases puedan ser reutilizadas, favoreciendo la implantación del sistema.

- El método de prueba de caja negra permitió verificar que las funciones son operativas a través de la interfaz del software, conservando así la integridad de la información externa.
- El método de prueba de caja blanca posibilitó verificar internamente las funciones de los componentes, facilitando la detección de irregularidades para su corrección.
- La evaluación del comportamiento de las variables incluidas en el problema de investigación, demostró que el sistema desarrollado contribuye a realizar los procesos con mayor rapidez, ya que se demuestra que con la solución se disminuyen el tiempo considerablemente.

## CONCLUSIONES FINALES

Teniendo en cuenta los resultados expuestos en este informe, la exigencia y el propósito establecido por la investigación se llegan a las siguientes conclusiones:

- La construcción del marco teórico de la investigación a través de la exploración y el análisis de los principales fundamentos teóricos en los que se basa la investigación facilitó la obtención de los conocimientos requeridos para la solución propuesta.
- La determinación de los requisitos, junto con el examen y la planificación de los elementos del sistema, posibilitaron obtener una aproximación para concebir los elementos indispensables de su implementación.
- Con la creación y puesta en marcha del sistema para la gestión de calidad de cursos virtuales se consiguió aportar a la rapidez y supervisión del proceso de gestión de calidad de cursos virtuales en el aula virtual del Centro Nacional de Educación a Distancia.
- Con los resultados adquiridos mediante las mediciones de validación del diseño y las pruebas de software se pudo verificar de manera cuantitativa la excelencia de los elementos obtenidos durante la creación de la solución propuesta.
- Mediante la comprobación de las variables de la investigación y los resultados obtenidos a través de los criterios evaluados antes y después del desarrollo del sistema para la gestión de calidad de cursos virtuales se pudo constatar la rapidez y supervisión en el proceso.

## RECOMENDACIONES

Se recomienda para optimizar el rendimiento y la eficiencia del sistema:

- Incorporar mecanismos para el monitoreo y control de las acciones críticas de seguridad en el sistema.
- Implementar funcionalidades que permita guardar un registro estadístico de todas las revisiones realizadas.

- **Alvarez, Miguel Angel. 2010.** Manual de jQuery. *Manual de jQuery*. [En línea] 2010. [www.desarrolloweb.com/manuales/manual-jquery.html](http://www.desarrolloweb.com/manuales/manual-jquery.html).
- **ANECA.es. 2021.** ANECA. *Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA)*. [En línea] 2021. <https://www.aneca.es/Quienes-somos>.
- **Apache.org. 2018.** The Apache Software Foundation. *The Apache Software Foundation*. [En línea] 2018. <http://jmeter.apache.org/>.
- **Bennett, James. 2008.** "Django Models and Databases: A Practical Guide". s.l. : APress, 2008.
- **Cabalé Miranda, Elizabeth y Rodríguez Pérez de Agreda, Gabriel. 2020.** *Sistemas de gestión. Importancia de su integración y vínculo con el desarrollo*. s.l. : Revista Estudios del Desarrollo Social: Cuba y América Latina, 2020. Artículo de revista. ISSN 2308-0132.
- **CHEA.org. 2021.** Council for Higher Education Accreditation (CHEA). (2021). About CHEA. *Sistema de Acreditación de la Educación Superior (CHEA)*. [En línea] 2021. <https://www.chea.org/about-chea>.
- **Coelho, Fabián. 2023.** Significados. *Significados*. [En línea] 15 de Mayo de 2023. <https://www.significados.com/metodologia/>.
- **Date, C. J. 2019.** *An Introduction to Database Systems*. Pearson. 2019.
- **DEAC.org. 2021.** Distance Education Accrediting Commission (DEAC). (2021). Who We Are. [En línea] 2021. <https://www.deac.org/Who-We-Are.aspx>.
- **DevOps, Azure. 2023.** Microsoft. *Microsoft*. [En línea] Abril de 2023. <https://learn.microsoft.com/es-es/devops/develop/git/what-is-version-control>.
- **Editorial Etecé. 2023.** concepto. *concepto*. [En línea] 2023. [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/Web\\_mechanics/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server).
- **Eguiluz, J. 2011.** *Desarrollo web ágil con Symfony 2-Primera edición*. 2011. 2011.
- **Eguiluz, Javier. 2020.** uniwebsidad. *uniwebsidad*. [En línea] 2020. <https://uniwebsidad.com/libros/javascript/capitulo-1>.
- **Eric Freeman, Elisabeth Robson, Bert Bates y Kathy Sierra. 2004.** "Head First Design Patterns". s.l. : O'Reilly Media, 2004.
- **Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. 1994.** "Design Patterns: Elementos de Software Orientado a Objetos Reutilizable". Boston, Estados Unidos. : Addison-Wesley , 1994.

- **Fowler, Martin. 2003.** *"Patrones de Arquitectura de Aplicaciones Empresariales.* s.l. : Pearson Educación, 2003.
- —. **2004.** *"UML Distilled: A Brief Guide to the Standard Object Modeling Language".* s.l. : Addison-Wesley Professional, 2004.
- **Gamma, Erich, Richard Helm, Ralph Johnson y John Vlissides. 1994.** *"Design Patterns: Elements of Reusable Object-Oriented Software".* s.l. : Addison-Wesley Professional , 1994.
- **García, J. C. 2019.** *Ventajas y desventajas de los cursos virtuales.* 2019.
- **García-Molina, H., & García, J. 2018.** *Sistemas de bases de datos: el modelo relacional.* s.l. : Pearson Educación, 2018.
- **Gaucha, Juan Diego. 2012.** *El gran libro de HTML5, CSS3 y Javascript.* Barcelona : MARCOMBO, 2012.
- **Gómez, Pérez y. 2020.** *"Mejorando la modularidad y el bajo acoplamiento en sistemas de software".* 2020.
- **Hernández Moya, Arleni Lázara. 2021.** *uciencia@uci.cu. UCIENCIA.* [En línea] octubre de 2021. <https://repositorio.uci.cu/jspui/handle/123456789/9840>.
- **Herrera Lugo, Yurier Jesús. 2022.** *Sistema para la gestión de préstamos en la Dirección de Extensión Universitaria de la UCI.* La Habana : 2022.
- **Higuerey, Edgar. 2020.** *rockcontent. rockcontent.* [En línea] 17 de Febrero de 2020. <https://rockcontent.com/es/blog/nginx/#:-:text=NGINX%20es%20un%20servidor%20web%20open%20source%2C%20de%20alta%20performance,proporciona%20m%C3%A1s%20velocidad%20y%20escalabilidad..>
- **ISO 9001-2015. 2015.** *ESG INNNOVA. NUEVA ISO 9001:2015 ES UNA INICIATIVA DEL GRUPO ESG INNNOVA.* [En línea] 2015. <https://www.nueva-iso-9001-2015.com/2016/09/desarrollo-concepto-calidad/>.
- **Jetbrains. 2022.** *The Python IDE for Professional Developers. The Python IDE for Professional Developers.* [En línea] 2022. <https://www.jetbrains.com/pycharm/>.
- **Larman, Craig. 2003.** *"Aplicando UML y Patrones: Introducción al Análisis y Diseño Orientado a Objetos y al Desarrollo Iterativo".* s.l. : Pearson Education, 2003.
- **Len Bass, Paul Clements y Rick Kazman. 1996.** *"Software Architecture in Practice".* Boston, Massachusetts, Estados Unidos. : Addison-Wesley Professional., 1996.
- **López López, E. 2017.** *La educación a distancia en la universidad cubana.* La Habana, Cuba : Félix Varela., 2017.

- **López Mendoza, Marvin. 2020.** *Qué es un lenguaje de programación.* 2020.
- **Lorenz, M. y Kidd, J. 1994.** *Métricas de Software Orientado a Objetos.* s.l. : s.l. : Prentice-Hall, 1994.
- **Manz, J Román. 2022.** Lenguaje HTML5. *Lenguaje HTML5.* [En línea] 2022. <https://lenguajehtml.com/html/>.
- **Martin, Robert C. 2000.** *"Design Principles and Design Patterns".* s.l. : Object Mentor, 2000.
- **Martinez, David L. la Red. 2012.** *Aprendizaje combinado, aprendizaje electrónico centrado en el alumno y nuevas tecnologías.* Argentina : Red de Universidades con Carreras en Informática (RedUNCI), 2012.
- **MDN Contributors. 2023.** mdn web docs. *mdn web docs.* [En línea] 2 de mayo de 2023. <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- **Ministerio de Educación y Formación Profesional. 2021.** Ministerio de Educación y Formación Profesional. [En línea] 2021. <https://www.educacionyfp.gob.es/servicios-al-ciudadano/catalogo/educacion/universidades/sistema-evaluacion-calidad-educacion-superior.html>.
- **Molina, Ismael El Hamiti. 2021.** *Desarrollo de interfaz gráfica e inclusión de elementos de gamificación en aplicación educativa.* Madrid : s.n., 2021.
- **Motta, Katihuska , Concha, Carlos y Muñoz, Natalie . 2020.** *Educación Virtual como Agente Transformador de los Procesos de Aprendizaje.* s.l. : Revista on line de Política e Gestão Educacional, 2020. vol 4.
- **Olsen, Russ. 2007.** *"Design Patterns in Ruby".* s.l. : Addison-Wesley Professional, 2007.
- **Ordóñez, Mariuxi Paola Zea, Rios, Jimmy Rolando Molina y Castillo, Fausto Fabián Redrován. 2019.** *Administración de Bases de datos con PostgreSQL.* 2019. Vol. Vol.19.
- **Ortiz, Arleni Lázara Ruiz. 2021.** UCIENCIA. *UCIENCIA.* [En línea] Octubre de 2021. <https://repositorio.uci.cu/jspui/handle/123456789/9840>.
- **Otero Vidal, Mari Carmen. 2018.** *Introducción al Modelado y Desarrollo de Software con UML.* s.l. : Dpto. de Lenguajes y Sistemas Informáticos, 2018.
- **Oxford Languages. 2023.** Diccionario de Oxford Languages. [En línea] 2023. <https://languages.oup.com/google-dictionary-es>.
- **Paradigm, Visual. 2013.** *Visual Paradigm for UML-UML tool for software application development.* 2013. pág. 72.

- **Pérez Villazón, Yoandy. 2015.** *Estrategia para la migración a aplicaciones de código abierto*. La Habana : s.n., 2015.
- **Pérez, Martínez. 2017.** *Manual de metodología de la investigación científica*. 2017.
- **Perez, Porto y Gardey, A. 2023.** Definicion.De. *Definicion.De*. [En línea] 15 de mayo de 2023. <https://definicion.de/celeridad/>.
- **Pichler, Roman. 2010.** *"Agile Product Management with Scrum: Creating Products that Customers Love"*. s.l. : Addison-Wesley Professional, 2010.
- **Poth, John W. Creswell y Cheryl N. 2017.** *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*. California, Estados Unidos : Sage Publications, 2017.
- **Pressman, Roger S. y Maxim, Bruce R. 2020.** *Software Engineering: A practitioner's approach*. Nueva York : McGraw-Hill Education, 2020. Vol. 9th. ISBN 978-1-260-54800-6.
- **Ramos, Joan Manuel Guevara. 2017.** Aula formativa. *Aula formativa*. [En línea] 30 de junio de 2017. <https://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>.
- **Ravindran, Arun. 2022.** *"Django Design Patterns and Best Practices"*. 2022.
- **2023.** Red Hat. *Red Hat*. [En línea] 20 de Enero de 2023. <https://www.redhat.com/es/topics/middleware/what-is-ide>.
- **Robson, Eric Freeman y Elisabeth. 2004.** *"Head First Design Patterns"*. s.l. : O'Reilly Media, 2004.
- **Rossum, Guido van. 2009.** *El tutorial de Python*. s.l. : luniverse Inc, 2009.
- **S.L., CriarWeb. 2023.** Manual de Git. *Manual de Git*. [En línea] 2023. <https://desarrolloweb.com/home/git>.
- **Sánchez Rodríguez, Tamara . 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.
- **Sommerville, Ian. 2016.** *Software Engineering*. Tenth edition . England : Pearson Education Limited, 2016. ISBN 10: 1-292-09613-6.
- **Sutherland, Ken Schwaber . Jeff. 2017.** *"The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game"*. s.l. : Scrum.org, 2017.
- **Vincent, William S. 2019.** *"Django for Beginners"*. s.l. : "Independently published", 2019.
- **Wieggers, Karl E. 2013.** *"Software Requirements"(3rd Edition)*. s.l. : Microsoft Press, 2013.



## ANEXOS

### ANEXO 1: Entrevista realizada a los profesores del CENED durante el levantamiento de información

63

Según lo que vimos en el encuentro pasado se identificó que tienen problema con la revisión de calidad de los cursos virtuales a distancia disponibles en el Entorno Virtual de Aprendizaje por lo cual es necesario que responda las siguientes preguntas:

1. ¿Teniendo en cuenta los indicadores a revisar que trae consigo pérdida de tiempo y no eficiencia por el volumen de indicadores a revisar, quisiera saber en su entorno que tan real es?
2. ¿Cuál es la situación actualmente en la entidad o departamento con el proceso de revisión de calidad de los cursos virtuales a distancia disponibles en el Entorno Virtual de Aprendizaje que lo prefieren informatizar?
3. ¿Qué sucede actualmente en la entidad a la hora del proceso de revisión con la generación de los informes y generación de notificaciones?
4. ¿Tiene alguna especificación en cuanto a los elementos no funcionales del sistema?

## ANEXO 2: Resultados de la prueba automatizada para el Requisito Revisar Curso.

```

Adding permission 'Permission object (50)'
Adding permission 'Permission object (51)'
Adding permission 'Permission object (52)'
Adding permission 'Permission object (53)'
Adding permission 'Permission object (54)'
Adding permission 'Permission object (55)'
Adding permission 'Permission object (56)'
Adding permission 'Permission object (57)'
Adding permission 'Permission object (58)'
Adding permission 'Permission object (59)'
Adding permission 'Permission object (60)'
Running post-migrate handlers for application administration
Running post-migrate handlers for application revision
System check identified no issues (0 silenced).
test_example (apps.accounts.tests.test_models.ErrorViewTests)
Test example ... ok
test_asignar_revision (apps.core.tests.testrevision) ... ok

-----

Ran 2 tests in 0.006s

OK
Preserving test database for alias 'default' ('file:memorydb_default?mode=memory&cache=shared')...

```

## ANEXO 3: Descripción de las variables: RF\_Registrar Curso

| No . | Nombre de campo | Clasificación     | Valor Nulo | Descripción                                      |
|------|-----------------|-------------------|------------|--|
| 1    | Nombre          | Lista desplegable | No         | Listado con el nombre de todos los cursos.       |
| 2    | Tipo            | Lista desplegable | No         | Listado con el tipo de todos los cursos.         |
| 3    | Estado          | Lista desplegable | No         | Listado con el estado de revisión de los cursos. |

## ANEXO 4: Escenario del DCP: RF\_ Registrar Curso

| Escenario                    | Descripción   | Nombre                          | Tipo        | Estado   | Respuesta del sistema  | Flujo central                     |
|------------------------------|---|---------------------------------|-------------|----------|--|-----------------------------------|
| EC 1.1<br>Campos correctos   | Funcionalidad que permite registrar un curso en el sistema.         | Introducción a la programación  | Verano      | Listo    | Al dar clic en el botón Guardar el sistema muestra el mensaje "Se ha registrado un curso." y regresa a la interfaz anterior, si selecciona la opción de Cancelar regresa a la interfaz anterior. | Inicio / Cursos / Registrar Curso |
| EC 1.2<br>Campos incorrectos | Al poner un dato incorrecto en un campo el sistema avisa del error. | Historia                        | Invierno    | Regular  | El sistema elimina el formato incorrecto   | Inicio / Cursos / Registrar Curso |
|                              | Al poner un dato incorrecto en un campo el sistema avisa del error. | Metodología de la Investigación | 1998        | No Listo | El sistema muestra el mensaje de error "Este campo debe comenzar con mayúscula y solo contener letras.". Además, no deja Registrar el Curso.   | Inicio / Cursos / Registrar Curso |
| EC 1.3<br>Campos vacíos      | Al dejar un campo vacío el sistema avisa del error.                 | (Vacío)                         | Aniversario | Listo    | El sistema muestra el mensaje de error "Hay campos incompletos". Además, señala los campos q son obligatorios y no deja guardar el pedido.   | Inicio / Cursos / Registrar Curso |
|                              | Al dejar un campo vacío el sistema avisa del error.                 | Programación Web                | (Vacío)     | No Listo | El sistema muestra el mensaje de error "Hay campos incompletos". Además, señala los campos q son obligatorios y no deja guardar el pedido.   | Inicio / Cursos / Registrar Curso |

**ANEXO 5:** Encuesta realizada al personal responsable de gestionar los indicadores de calidad de los cursos del aula virtual del Centro Nacional de Educación a Distancia.

Este cuestionario se considera anónimo y personal y tiene como objetivo determinar el tiempo promedio que normalmente se tarda en realizar la revisión de los cursos. La rapidez es muy importante para mejorar nuestro trabajo.

Por favor, complete el cuestionario cuidadosamente después de leerlo y marque sus respuestas con una "X".

1. ¿Cuánto tiempo tarda en recopilar los datos al registrar un curso?

- a) \_ Menos de 1 minuto.
- b) \_ De 5 a 10 minutos.
- c) \_ De 10 a 15 minutos.
- d) \_ Otro ¿Cuál?

2. ¿Cuánto tiempo tarda en buscar un curso en la documentación que se procesa de forma manual?

- a) \_ Menos de 1 minuto.
- b) \_ De 5 a 10 minutos.
- c) \_ De 10 a 15 minutos.
- d) \_ Otro ¿Cuál?

3. Una vez que vence el plazo de revisión de un curso, es necesario notificar a los usuarios sobre el incumplimiento de la revisión. Para esta actividad, ¿cuánto tiempo promedio se tarda en notificar a los usuarios sobre este incumplimiento?

- a) \_ Menos de 1 día.
- b) \_ Entre 1 y 4 días.
- c) \_ Más de 4 días.

## ANEXO 6: Resultados de las pruebas de Rendimiento al procesar los indicadores de calidad por el sistema propuesto

| Media | Mín | Máx | Std. Dev. | % Error | Rendimiento | Kb/sec | Avg. Bytes |
|-------|-----|-----|-----------|---------|-------------|--------|------------|
| 52    | 42  | 61  | 6,49      | 0,00%   | 5,8/sec     | 50,57  | 8916,6     |
| 50    | 42  | 61  | 8,36      | 0,00%   | 5,9/sec     | 53,67  | 9331,6     |
| 46    | 41  | 54  | 5,04      | 0,00%   | 5,9/sec     | 54,07  | 9413,2     |
| 43    | 40  | 51  | 3,82      | 0,00%   | 5,9/sec     | 53,96  | 9382,8     |
| 44    | 40  | 48  | 2,80      | 0,00%   | 5,9/sec     | 51,30  | 8899,6     |
| 50    | 42  | 63  | 9,99      | 0,00%   | 5,9/sec     | 53,01  | 9152,6     |
| 48    | 40  | 63  | 8,05      | 0,00%   | 5,9/sec     | 53,41  | 9199,2     |
| 48    | 40  | 63  | 7,48      | 0,00%   | 31,4/sec    | 282,07 | 9185,1     |

**Anexo 6.1:** Resultados con una carga de 5 usuarios, y ejecutando el flujo Autenticar/ Buscar curso/ Registrar resultados de la revisión.

| Media | Mín | Máx | Std. Dev. | % Error | Rendimiento | Kb/sec | Avg. Bytes |
|-------|-----|-----|-----------|---------|-------------|--------|------------|
| 106   | 69  | 166 | 43,36     | 0,00%   | 5,2/sec     | 45,02  | 8897,2     |
| 114   | 54  | 182 | 52,82     | 0,00%   | 4,7/sec     | 41,62  | 9145,0     |
| 131   | 50  | 193 | 56,34     | 0,00%   | 4,3/sec     | 39,27  | 9304,4     |
| 117   | 79  | 145 | 22,58     | 0,00%   | 4,1/sec     | 35,42  | 8937,4     |
| 134   | 77  | 170 | 27,85     | 0,00%   | 7,0/sec     | 61,15  | 8985,0     |
| 122   | 76  | 206 | 48,14     | 0,00%   | 3,7/sec     | 32,95  | 9090,8     |
| 125   | 82  | 172 | 31,12     | 0,00%   | 3,7/sec     | 32,84  | 9113,4     |
| 106   | 68  | 179 | 43,82     | 0,00%   | 3,8/sec     | 34,44  | 9325,2     |
| 88    | 56  | 133 | 28,47     | 0,00%   | 3,8/sec     | 34,36  | 9163,2     |
| 87    | 42  | 150 | 43,19     | 0,00%   | 4,0/sec     | 36,22  | 9242,0     |
| 81    | 43  | 157 | 40,29     | 0,00%   | 4,4/sec     | 39,97  | 9323,6     |
| 112   | 42  | 206 | 44,01     | 0,00%   | 28,8/sec    | 256,83 | 9126,0     |

**Anexo 6.2:** Resultados con una carga de 5 usuarios, y ejecutando el flujo Autenticar/ Buscar curso/ Generar informe de revisión.

| Media | Mín | Máx | Std. Dev. | % Error | Rendimiento | Kb/sec | Avg. Bytes |
|-------|-----|-----|-----------|---------|-------------|--------|------------|
| 75    | 46  | 135 | 32,09     | 0,00%   | 5,3/sec     | 46,09  | 8892,4     |
| 58    | 41  | 94  | 19,97     | 0,00%   | 5,1/sec     | 44,94  | 9101,8     |
| 60    | 43  | 71  | 9,79      | 0,00%   | 4,9/sec     | 44,61  | 9301,2     |
| 61    | 43  | 87  | 15,94     | 0,00%   | 5,0/sec     | 43,54  | 8943,8     |
| 61    | 43  | 82  | 11,47     | 0,00%   | 9,4/sec     | 82,53  | 9000,2     |
| 63    | 42  | 105 | 23,03     | 0,00%   | 4,9/sec     | 43,55  | 9079,6     |
| 63    | 46  | 87  | 17,36     | 0,00%   | 4,9/sec     | 43,28  | 9094,2     |
| 58    | 43  | 74  | 12,59     | 0,00%   | 4,9/sec     | 44,42  | 9298,4     |
| 52    | 40  | 71  | 10,30     | 0,00%   | 4,9/sec     | 42,72  | 8906,0     |
| 61    | 40  | 82  | 14,23     | 0,00%   | 5,0/sec     | 44,53  | 9147,8     |
| 61    | 47  | 84  | 12,50     | 0,00%   | 4,9/sec     | 44,24  | 9188,0     |
| 61    | 40  | 135 | 17,80     | 0,00%   | 38,6/sec    | 342,56 | 9079,5     |

**Anexo 6.3:** Resultados con una carga de 10 usuarios, y ejecutando el flujo Autenticar/ Buscar curso/ Registrar resultados de la revisión.

| Media | Mín | Máx | Std. Dev. | % Error | Rendimiento | Kb/sec | Avg. Bytes |
|-------|-----|-----|-----------|---------|-------------|--------|------------|
| 51    | 43  | 62  | 8,32      | 0,00%   | 6,0/sec     | 52,66  | 9026,6     |
| 50    | 41  | 61  | 8,05      | 0,00%   | 5,8/sec     | 52,24  | 9148,2     |
| 47    | 42  | 64  | 8,53      | 0,00%   | 5,8/sec     | 53,14  | 9304,4     |
| 49    | 42  | 65  | 8,28      | 0,00%   | 5,8/sec     | 51,35  | 8990,8     |
| 49    | 40  | 64  | 7,25      | 0,00%   | 16,2/sec    | 146,39 | 9263,7     |
| 47    | 40  | 60  | 6,80      | 0,00%   | 6,1/sec     | 54,77  | 9131,2     |
| 48    | 40  | 65  | 10,95     | 0,00%   | 34,8/sec    | 311,89 | 9174,0     |

**Anexo 6.4:** Resultados con una carga de 10 usuarios, y ejecutando el flujo Autenticar/ Buscar curso/ Generar informe de revisión.