

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



**Universidad de las Ciencias
Informáticas**

Facultad 1

Sitio web para la Junta de Acreditación Nacional de Cuba

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Yohan González Almaguer

Tutor:

Aylin Estrada Velazco

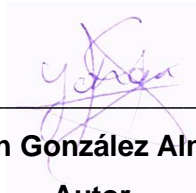
La Habana, noviembre de 2021

“Año 63 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo **Yohan González Almaguer**, con carné de identidad **97100908262** soy el autor principal del trabajo titulado “Sitio web para la Junta de Acreditación Nacional de Cuba” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____
de _____



Yohan González Almaguer
Autor

Msc. Aylin Estrada Velazco
Tutor

Resumen

Las instituciones acreditadoras de la enseñanza superior son un eslabón esencial en la acreditación universitaria. Lograr que esas instituciones tengan un alto nivel de difusión es de vital importancia en los días que corren. Por lo que tener un portal web es un gran avance en ese camino. En la actualidad la Junta de Acreditación Nacional (JAN) no presenta un sitio web que le permita difundir su labor en la comunidad universitaria en los procesos de acreditación de instituciones, carreras y programas de postgrado y maestrías. En el presente trabajo de diploma se plantea el desarrollo un portal web. Dicho portal dará la posibilidad a la JAN de tener mayor presencia en la red y lograr un mayor nivel de difusión. Se muestran las distintas facetas del desarrollo, como el proceso de levantamiento de requisitos, análisis y diseño de la propuesta de solución a la problemática y por último el diseño de pruebas de software que validen la calidad del producto final.

Palabras claves: instalación, acreditación, enseñanza superior, portal web, difusión

Índice

Introducción	1
Problema de Investigación	3
Objeto de Estudio	3
Objetivo General	3
Objetivos Específicos	3
Hipótesis	4
Posibles resultados	4
Métodos Científicos	4
Estructura del documento	4
Capítulo 1. Referentes Teórico-Methodológicos sobre los procesos de evaluación de la calidad en la Educación Superior	6
1.1 La evaluación de la calidad en la educación superior	6
1.2 Análisis de organismos/instituciones evaluadoras de la calidad	11
1.3 Tecnologías para el desarrollo del portal web	16
1.4 Conclusiones del capítulo	26
Capítulo 2. Diseño del portal web para la Junta de Acreditación Nacional	27
2.1 Propuesta de portal web para la Junta de Acreditación Nacional	27
2.2 Modelo Conceptual	28
2.3 Especificación de requisitos	28
2.4 Historias de usuario	34
2.4 Diseño arquitectónico	36
2.5 Diagrama de clases de diseño con estereotipos web	39
2.6 Diagramas de secuencia	42
2.7 Modelo de datos	46
2.8 Diagrama de despliegue	47
2.9 Conclusiones del capítulo	48
Capítulo 3: Implementación y Pruebas del Sistema	49
3.1 Pruebas de software para la evaluación de la propuesta de solución	49
3.2 Estrategia de pruebas	49
3.3 Conclusiones del capítulo	55
Conclusiones	56
Referencias Bibliográficas	57

Índice de Tablas

Tabla 1: Análisis de herramientas homólogas	15
Tabla 2: Especificación de Requisitos Funcionales	29
Tabla 3: Especificación de Requisitos No Funcionales	33
Tabla 4: HU Crear Publicación	34
Tabla 5: HU Buscar Publicación	35
Tabla 6: HU Responder Inquietud	35

Índice de Figuras

Figura 1: Representación gráfica de los escenarios de la metodología AUP-UCI	25
Figura 2: Modelo Conceptual	28
Figura 3: Diagrama de Clases con Estereotipos web de CU Crear Publicación	40
Figura 4: Diagrama de clases con Estereotipos web CU Buscar institución	41
Figura 5: Diagrama de clases con estereotipos web de CU Responder inquietud.....	42
Figura 6: Diagrama de Secuencia de CU Crear publicación.....	43
Figura 7: Diagrama de Secuencia de CU Buscar Institución	44
Figura 8: Diagrama de Secuencia de CU Responder Inquietud	45
Figura 9: Modelo de Datos	46
Figura 10: Diagrama de Despliegue.....	47
Figura 11: Diagrama de Despliegue.....	47
Figura 12: Especificación de No Conformidades.....	¡Error! Marcador no definido.
Figura 13: Especificación Detallada de las No Conformidades	¡Error! Marcador no definido.
Figura 14: Especificación de Resultado de Pruebas de usabilidad.....	¡Error! Marcador no definido.

Introducción

La educación es el pilar fundamental del crecimiento de los pueblos. La capacidad de respuesta de los individuos ante situaciones concretas, su efectividad en la toma de decisiones, su nivel de actividad y su actitud ante situaciones problemáticas, están determinados por la calidad de la educación recibida.

El contexto socioeconómico existente se torna cada vez más complejo y competitivo, con demandas crecientes de eficiencia y transparencia de resultados, lo que lleva a las instituciones educativas a replantear su accionar y establecer modelos de calidad acordes a sus fines y características propias. Por otra parte, el concepto de calidad es uno de los más discutidos en educación y sobre él existe una variedad de definiciones. Es por tanto de interés profundizar en los aspectos que intervienen en la calidad educativa y la forma en que hoy se concibe este término en las instituciones docentes.

Adquirir una buena educación es parte fundamental en la vida de las personas, esto puede determinar en cuanto impacta en la sociedad y como se proyecta hacia esta. Por lo que recibir una educación de calidad es casi o tan importante como recibirla en primer lugar [1].

Hemos de reconocer que, en las dos últimas décadas, y por factores externos a la universidad, el tema de la calidad ha cobrado especial relevancia: es necesario evaluar en qué medida es posible reducir ciertos costes universitarios dada su escasa contribución a la calidad de la educación universitaria. Inicialmente el motivo no ha sido una loable preocupación por la calidad, sino la presión ejercida por la reducción presupuestaria. Así, las universidades europeas se han tenido que enfrentar con una situación relativamente nueva como ha sido la de la evaluación institucional.

Si consideramos la institución universitaria como un sistema organizativo (racional, natural o abierto, según concepciones y estadios de desarrollo), el logro de la calidad de una institución (rendimiento en términos de producto) viene determinado tanto por las acciones individuales como por la interacción de los individuos con sus unidades organizativas. Por tanto, es totalmente reduccionista aproximarse a la evaluación de la calidad a través de la evaluación individual de los agentes, mientras que es pertinente y necesario un análisis del rendimiento institucional, así como de los factores organizativos y ambientales que contextualizan posibles y posteriores análisis individuales [2].

La idea de **calidad educativa**, hace alusión a **cómo se lleva a cabo este proceso de formación**. Cuando los resultados y los efectos de la educación son valorados de manera positiva por la **comunidad**, la calidad educativa es alta. En cambio, cuando esto no sucede, la calidad educativa será calificada como baja.

Existen múltiples factores que inciden en la calidad educativa. Además de la **pedagogía**, que se relaciona a cómo se educa, influyen cuestiones como el **tipo de contenidos**, las **instalaciones** donde se enseña y la validez o utilidad de los **títulos** otorgados.

De la misma manera, hay que tener en cuenta que se considera que un sistema tiene una innegable calidad educativa cuando reúne estas otras características:

- Promueve el cambio y la modernización en las aulas.
- Consigue que la comunidad educativa en general se implique.
- Se ajusta a las necesidades del alumnado para que pueda alcanzar las herramientas y recursos en pro de conseguir su progreso a nivel académico.
- Vela, de igual modo, por el bienestar del profesorado y se preocupa por su motivación.

Una buena calidad educativa se alcanza cuando los procesos **satisfacen las necesidades** del individuo y de la **sociedad** en general. Esto se logra si los recursos son suficientes, además están aprovechados de manera adecuada para que la educación sea equitativa y eficaz. Si un adolescente completa la educación secundaria y no tiene los conocimientos necesarios para cursar con éxito una carrera universitaria o para insertarse en el mercado laboral, habrá sido víctima de un sistema con una calidad educativa deficiente. Si la escuela prepara al alumno para afrontar los desafíos de la vida adulta, la calidad educativa será digna de destacar [3].

La importancia de la evaluación para los sistemas educativos es un punto en el que existe pleno consenso en que es de vital importancia para que la sociedad como unidad se mantenga evolucionando. Es casi imposible pensar en acciones educativas que no sean sometidas a evaluación. La discusión toma la dirección de cómo llevarla a cabo, de manera tal que apoye al sistema educativo. Dicho debate hace referencia a criterios de distinta naturaleza que engloban elementos de orden político, teórico, metodológico, instrumental y ético que dan forma y sustentan este campo de reciente aparición [4].

En Cuba, el Ministerio de Educación Superior (MES), tiene el encargo estatal de controlar el desempeño de las instituciones de educación superior por ser un elemento fundamental para lograr un mejoramiento continuo de la calidad educativa. El MES tiene establecido el Sistema

Universitario de Programas de Acreditación (SUPRA), dirigido a fortalecer la cultura de la calidad en la educación superior, y a promover, estimular y certificar la calidad de los diferentes procesos e instituciones de este nivel de enseñanza. El sistema SUPRA es rectorado por la Junta de Acreditación Nacional (JAN), un órgano constituido por profesionales seleccionados de diferentes Instituciones de Educación Superior (IES) del país u otros organismos, que tiene la misión de contribuir al mejoramiento de la calidad de la educación superior.

Por otra parte, se puede destacar que la JAN realiza en el año múltiples actividades: evaluaciones periódicas, dos plenos en el año, seminarios de capacitación para sus evaluadores, entre otras. La JAN en la actualidad carece de medios y herramientas para la gestión y divulgación de las tareas que llevan a cabo en el país durante la acreditación de instituciones de la educación superior.

Problema de Investigación

¿Cómo contribuir al proceso de evaluación de la calidad de la educación superior por la Junta de Acreditación Nacional de Cuba y de sus sistemas de acreditación?

Objeto de Estudio

Como **objeto de estudio** de esta investigación se plantea: los procesos de la evaluación de la calidad de las instituciones de la educación superior. Enfocado al **campo de acción**: proceso de evaluación de la calidad por la Junta de Acreditación Nacional y sus sistemas de acreditación.

Objetivo General

Desarrollar una solución informática que contribuya al proceso de evaluación de la calidad de la educación superior, potenciando la divulgación del quehacer de la Junta de Acreditación Nacional de Cuba y sus sistemas de acreditación.

Objetivos Específicos

- Construir los referentes teóricos que sustentan la investigación relacionados con la evaluación de la calidad de la educación superior por la Junta de Acreditación Nacional de Cuba y de sus sistemas de acreditación.
- Diagnosticar el estado actual de sistema informáticos para la gestión y divulgación de información en instituciones e identificar las actividades que puedan ser desarrolladas como funcionalidades del sistema.
- Diseñar las funcionalidades del sistema para la gestión y divulgación del quehacer de la Junta de Acreditación Nacional de Cuba y sus sistemas de acreditación.

- Implementar las funcionalidades del sistema para la gestión y divulgación del quehacer de la Junta de Acreditación Nacional de Cuba y sus sistemas de acreditación.

Hipótesis

Con el desarrollo de una solución informática que contribuya al proceso de evaluación de la calidad de la educación superior se potenciará la divulgación del quehacer de la Junta de Acreditación Nacional de Cuba y sus sistemas de acreditación.

Posibles resultados

1. Marco teórico referencial sobre el proceso de gestión y divulgación de la Junta de Acreditación Nacional de Cuba y sus sistemas de acreditación.
2. Aplicación web que contribuya a la gestión y divulgación del quehacer de la Junta de Acreditación Nacional de Cuba y sus sistemas de acreditación.

Métodos Científicos

Métodos Empíricos

- **Observación:** para captar información pertinente acerca de los mecanismos de la evaluación de la calidad de instituciones.
- **Encuestas:** para obtener información sobre el proceso de evaluación de la calidad que lleva a cabo la JAN. Para identificar los elementos de mayor impacto e importancia durante este proceso. El instrumento a utilizar es el cuestionario.
- **Entrevista:** para evaluar la importancia de cada uno de los pasos que conlleva un proceso de evaluación de calidad por parte de la JAN en las instituciones de la educación superior. El instrumento a utilizar es la guía de entrevista.

Métodos Teóricos

- **Histórico-lógico:** para analizar en profundidad los conceptos asociados a los mecanismos y procesos de la evaluación de la calidad de instituciones.
- **Analítico-sintético:** para el estudio de trabajos similares e investigaciones que aborden el tema de evaluación de la calidad de instituciones.

Estructura del documento

El presente trabajo de diploma tendrá la siguiente estructura: introducción, tres capítulos, conclusiones y recomendaciones. Además, se agregan los anexos que permiten comprender mejor la investigación realizada.

En el **Capítulo 1. Referentes Teórico-Metodológicos sobre los procesos de evaluación de la calidad en la Educación Superior**, se realiza un análisis de los conceptos como el conocimiento, la calidad y evaluación de la calidad y del conocimiento. Se establece un modelo del funcionamiento del proceso de evolución de la calidad y definir las tecnologías informáticas para el desarrollo de la aplicación.

En el **Capítulo 2. Diseño del portal web para la Junta de Acreditación Nacional**, describe el desarrollo de las actividades correspondientes al análisis y diseño de la herramienta. Se explican las características principales de los productos de trabajo generados como resultado de esas actividades.

En el **Capítulo 3: Implementación y Pruebas del Sistema**, validación de la solución propuesta, implementación y pruebas del sistema, se muestra una descripción de la validación de la propuesta. Se muestran los elementos para realizar las pruebas y los resultados obtenidos de las mismas.

Capítulo 1. Referentes Teórico-Metodológicos sobre los procesos de evaluación de la calidad en la Educación Superior

En este capítulo se desarrollan las bases teóricas y conceptuales de la evaluación de la calidad de la educación superior y los procesos que la conforman. Para ello se abordan los conceptos más relevantes asociados con la temática con referencia de sus autores. Se analizan otras soluciones informáticas similares que tributen a lograr un producto final de calidad. Para culminar el capítulo se define la arquitectura que se empleará en la solución y se determinará la metodología a implementar durante el desarrollo.

1.1 La evaluación de la calidad en la educación superior

Para Davenport y Prusak (1999) el conocimiento es una mezcla de experiencia, valores, información y “saber hacer” que sirve como marco para la incorporación de nuevas experiencias e información, y es útil para la acción. Se origina y aplica en la mente de los conocedores. En las organizaciones con frecuencia no sólo se encuentra dentro de documentos o almacenes de datos, sino que también está en rutinas organizativas, procesos, prácticas, y normas [5].

Según el diccionario de la Real Academia Española, puede definirse el concepto conocimiento como la “acción o efecto de conocer” [6], luego también se define como el acto humano de conocer. Poniéndolo a partir del punto de vista filosófico, el conocimiento humano es la característica que principal que lo diferencia de un animal, es decir, se refiere a que el hombre es racional. El desarrollo del conocimiento hace posible interiorizar mejor el mundo, desarrollar la ciencia y su concreción práctica [7].

Se denomina educación a la facilitación del aprendizaje o de la obtención de conocimientos, habilidades, valores y hábitos en un grupo humano determinado, por parte de otras personas más versadas en el asunto enseñado y empleando diversas técnicas de la pedagogía: la narración, el debate, la memorización o la investigación [8]. La educación puede definirse como el proceso de socialización de los individuos. Al educarse, una persona asimila y aprende conocimientos. La educación también implica una concienciación cultural y conductual, donde las nuevas generaciones adquieren los modos de ser de generaciones anteriores [9].

En su sentido más amplio, por educación se entiende el proceso por el cual se transmite el conocimiento, los hábitos, las costumbres y los valores de una sociedad a la siguiente generación. Educación viene del latín *educere* que significa “sacar”, “extraer”, y *educare* que significa “formar”, “instruir”. La educación comprende también la asimilación y práctica de las

normas de cortesía, delicadeza y civismo. De allí que en el lenguaje popular la práctica de estos hábitos de socialización sea calificados como signos de una buena educación [10].

Cuando hablamos de **calidad en la educación** se refieren a aquel funcionamiento en los **centros educativos** que permite tener un control de todos los procesos llevados a cabo en los mismos, así como la **correcta gestión** de estos [11]. La idea de calidad educativa, en este marco, hace alusión a cómo se lleva a cabo este proceso de formación. Cuando los resultados y los efectos de la educación son valorados de manera positiva por la comunidad, la calidad educativa es alta. En cambio, cuando esto no sucede, la calidad educativa es calificada como baja.

La etimología del término “calidad” se encuentra en el vocablo latino “*qualitas- atis*”. En el Diccionario General de la Lengua Española se define como “un conjunto de cualidades o propiedades inherentes a algo o alguien” y se hace referencia a la calidad como “superioridad o excelencia, como importancia o gravedad de algo, como lo que tiene más valor, propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie” [1].

Es frecuente que cuando se menciona una **institución educativa** sea equivalente a instituto u organización educativa. Pero una institución no se refiere siempre a un sitio físico. A las costumbres y normas de conducta que definen a una sociedad también se les denomina como instituciones. De manera que una institución puede entenderse también como una estructura de tipo social que determina el accionar de una sociedad [12].

Una evaluación es un juicio cuya finalidad es establecer, tomando en consideración un conjunto de criterios o normas, el valor, la importancia o el significado de algo. Como tal, la evaluación es aplicable a distintos campos de la actividad humana, como la educación, la industria, la salud, la psicología, la gestión empresarial, la economía, las finanzas, la tecnología, entre otros. De allí que puedan evaluarse muchas actividades: el desempeño laboral de un individuo, el valor de un bien en el mercado, el desarrollo de un proyecto, el estado de salud de un paciente, la calidad de un producto, la situación económica de una organización, etc. [9].

Se denominada **Organismo Acreditador**, asociación civil sin fines de lucro, nacional o internacional, externa a las Instituciones de Educación Superior, dedicada a la acreditación de éstas y sus programas. Evalúa la calidad educativa y acredita públicamente, entre otras posibilidades, programas e instituciones [13]. Se trata de organizaciones que realizan auditorías para revisar que las empresas cumplen una serie de requisitos de una norma y certificar ese

hecho. Dichas auditorías son realizadas por auditores designados por el organismo certificador que tengan las competencias necesarias para llevar a cabo ese trabajo [14].

Según la RAE la educación o enseñanza superior se define por: enseñanza que comprende los estudios especiales que requiere cada profesión o carrera [15]. Educación que forma parte del sistema educativo y comprende la enseñanza universitaria, las enseñanzas artísticas superiores, la formación profesional de grado superior, las enseñanzas profesionales de artes plásticas y diseño de grado superior y las enseñanzas deportivas de grado superior [16].

A medida que los sistemas de educación superior van creciendo y diversificándose, la sociedad se va preocupando más por la calidad de sus programas y concede mayor importancia a las valoraciones públicas y a los rankings internacionales de las instituciones universitarias. Sin embargo, este tipo de comparaciones tiende a poner demasiado énfasis sobre la investigación, y a usar los resultados de esta como medida estándar del valor de cada institución [1].

Desde instituciones como UNESCO o la OCDE se han realizado estudios exploratorios alrededor del tema de la preocupación e interés por la calidad y su aseguramiento en las instituciones de educación superior. Algunas de sus conclusiones preliminares han girado sobre aspectos como los siguientes:

- La participación de los más altos niveles de la jerarquía universitaria es clave para el éxito en la introducción de una cultura de la calidad.
- La enseñanza es considerada importante por las instituciones de educación superior. A pesar de que la calidad de la enseñanza varía considerable y constantemente en cuanto a definición y conceptos, existe un cierto número de iniciativas (acciones, estrategias, políticas) que tienden inequívocamente a mejorar la calidad educativa.
- La calidad de la enseñanza se debe ver como algo dinámico, en el contexto más amplio y cambiante de la educación superior. Los estudios se están internacionalizando y la educación superior debe aportar soluciones en nuevas áreas del conocimiento (como la innovación y el desarrollo cívico y regional) para poder producir una fuerza de trabajo apta y capaz de enfrentarse a los desafíos de nuestro siglo XXI.
- Las instituciones quieren que se les reconozca como proveedoras de enseñanza universitaria de calidad. Entienden que competir sobre la sola base de la investigación no basta ya para asegurarse una buena reputación.
- Los responsables de las universidades deben tratar de actuar sobre todas las dimensiones que afectan la calidad de la enseñanza. Se exige, por tanto, la

identificación de las mismas a partir de un riguroso análisis de la realidad desde una perspectiva internacional.

- Los estudiantes quieren asegurarse de que sus estudios les darán la posibilidad de hallar trabajo, y que les conferirán los conocimientos necesarios para la sociedad de hoy y del futuro. La movilidad estudiantil y el aumento de las tasas universitarias incrementan la presión de los estudiantes sobre la calidad educativa.

En lógica consecuencia con esta preocupación, aparecen una serie de iniciativas de apoyo a la calidad de la enseñanza:

- Políticas de garantía de calidad en toda la institución: incluyen proyectos globales diseñados para desarrollar una cultura de calidad a nivel institucional.
- Control sobre los programas: incluye acciones para evaluar el diseño, el contenido y la ejecución de los programas (mayoritariamente mediante la evaluación de cada programa).
- Sistemas de apoyo a estudiantes y profesorado: mejora de los entornos de trabajo, refuerzos pedagógicos, formación permanente del profesorado, etc.
- Un compromiso institucional con relación a la calidad educativa a los niveles de responsabilidad más altos, así como departamentales requiere que los rectores y docentes identifiquen normas a seguir, reconozcan las buenas prácticas, las vayan introduciendo en todos los departamentos y establezcan una serie de apoyos efectivos tanto para los docentes como para los estudiantes [2].

Junta de Acreditación Nacional

La Junta de Acreditación Nacional (JAN) fue fundada por el Ministro de Educación Superior en el año 2000 y constituye un órgano asesor de la dirección de este. La acreditación de los Programas por la JAN es el proceso que reconoce, en una institución, el programa académico de una especialidad.

Los objetivos del JAN consisten en trazar lineamientos y decidir acerca de la actividad de acreditación de los procesos e instituciones de la educación superior; ejecuta las funciones de asesoría y validación de los procesos de evaluación externa y propone los ajustes del Sistema de autoevaluación de la especialidad de posgrado en cuestión (SEA-EP) en correspondencia con los resultados que se obtengan de su aplicación práctica sistemática. En dichos procesos también intervienen otros órganos como el Comité Académico del programa de cada

especialidad, la Comisión Asesora para la Educación de Posgrado (COPEP), la Comisión Evaluadora de un Programa y el Comité Técnico Evaluador.

Para que la JAN acepte la solicitud, se requiere realizar una carta, así como una evaluación exhaustiva por el Comité Académico de la especialidad de los últimos cinco años. Dentro de los objetivos principales de esta autoevaluación, se analizan una serie de variables dirigidas a constatar las evidencias que avalan la calidad de la formación a través del programa académico de cada especialidad.

Estas variables son:

- Pertinencia social
- Tradición de la institución y colaboración
- Profesores y tutores
- Aseguramiento didáctico y administrativo del Programa
- Escenario laboral
- Estudiantes
- Currículo

Cada una de estas variables debe ser desarrollada de forma pormenorizada, pues justamente el grado de evidencias que sean capaces de recolectar, contribuirá a un mejor resultado final. Mientras mayor sean las evidencias, mejor será el resultado.

Una vez entregada la autoevaluación, se realizará la evaluación externa y se revisarán todas las evidencias por los evaluadores externos y los miembros de Comité Técnico Asesor. Terminado este análisis, se realizará un dictamen para discutirlo en el comité académico, el colectivo de profesores, los directivos del hospital y de la JAN.

Este dictamen se lleva al pleno de la JAN, el cual decide si se acredita el programa y otorga cuatro categorías [50]:

- Programa autorizado
- Programa avalado
- Programa certificado
- Programa de excelencia

1.2 Análisis de organismos/instituciones evaluadoras de la calidad

En este epígrafe se presentan prestigiosas instituciones dedicadas a la evaluación de la calidad de la educación en diferentes locaciones que cuentan con sitios web para divulgar su quehacer y potenciar el logro de una educación de calidad. La comparativa se basará en las siguientes características: Descarga de documentos normativos, Búsquedas de instituciones y programas acreditados, Cronograma de evaluaciones, Comunicación con los usuarios, Boletines informativos, Redes sociales y RSS, Publicación de noticias y Publicación de artículos sobre la temática.

1.2.1 Junta de Acreditación de Ingeniería y Tecnología

Fundada originalmente en 1932 como el *Engineers' Council for Professional Development* (ECPD), en Baltimore, Estados Unidos en el año 1980, se convirtieron en la Junta de Acreditación de Ingeniería y Tecnología para describir con mayor precisión el énfasis en la acreditación. En 2005, para reflejar su ampliado alcance, comenzaron a utilizar el acrónimo ABET únicamente.

Actualmente acreditan 4,307 programas en 846 colegios y universidades en 41 países. Cada año, 2.200 expertos de 35 sociedades miembros contribuyen al objetivo de ABET de garantizar la confianza en la educación en ciencia, tecnología, ingeniería y matemáticas (*Science, Technology, Engineering, and Mathematics*, STEM) [19]. Acreditan programas universitarios y universitarios en las disciplinas de ciencias naturales y aplicadas, informática, ingeniería y tecnología de ingeniería en los niveles de asociado, licenciatura y maestría.

ABET cuenta con un sitio web donde se puede apreciar un buen diseño de interfaz, además de ser *responsive* lo que da posibilidad de navegación con mayor comodidad a los usuarios que accedan a la web desde un dispositivo portátil. Se le da la posibilidad a los usuarios de descargar los documentos normativos de los procesos de evaluación y acreditación que lleva a cabo la institución. En este sitio se pueden realizar revisiones de otras instituciones y programas acreditados con anterioridad por la propia institución. Se publican los cronogramas de las futuras evaluaciones para que las instituciones involucradas puedan mantenerse informados. Permite que los usuarios envíen mensajes al equipo de soporte directamente comentando alguna inquietud. El sitio presenta un espacio donde busca mantener al usuario actualizado con los resultados de la institución. La web tiene enlaces directos a las redes sociales de la institución, como punto debilidad no cuentan con opciones de suscripción RSS. En el sitio se publican noticias relevantes sobre la institución, los procesos evaluativos que

realiza y sus resultados, así como pone a disposición de los usuarios artículos relacionados con la temática [17].

1.2.2 Consejo para la Acreditación de la Educación Superior

El Consejo para la Acreditación de la Educación Superior A.C. (COPAES) es una asociación civil sin fines de lucro que actúa como la única instancia autorizada por el Gobierno Federal de México a través de la Secretaría de Educación Pública (SEP), para conferir reconocimiento formal y supervisar a organizaciones cuyo fin sea acreditar programas académicos del tipo superior que se imparten en México, en cualquiera de sus modalidades (escolarizada, no escolarizada y mixta).

COPAES tiene un sitio web con un diseño de interfaz simple e intuitivo y adaptable a diferentes dispositivos, muy cómodo para los usuarios que lo visitan. Se le da la posibilidad a los usuarios a que accedan a los documentos normativos de los procesos de evaluación y acreditación que lleva a cabo la institución. En este espacio es posible encontrar y descargar los documentos normativos relacionados con la evaluación de la calidad de la educación superior en México sin embargo no ofrece el cronograma de evaluaciones del COPAES. Permite que los usuarios envíen mensajes al equipo de soporte directamente comentando alguna inquietud. No existe una sección en el sitio donde el usuario pueda buscar información con el objetivo de mantenerse al tanto de los resultados de la institución. La web tiene enlaces directos a las redes sociales de la institución, pero en este caso no hay opciones de RSS. En el sitio web no existe una sección donde se expongan las noticias de mayor relevancia sobre la institución. La institución no tiene un espacio donde publique artículos sobre la temática. Como punto a resaltar de esta web es que cuentan con un glosario de conceptos y acrónimos relaciones con la temática de la institución [18].

1.2.3 Agència per a la Qualitat del Sistema Universitari de Catalunya

La Agència per a la Qualitat del Sistema Universitari de Catalunya, AQU Catalunya, es el principal instrumento para la promoción y la evaluación de la calidad en el sistema universitario catalán. AQU Catalunya tiene como objetivo la evaluación, la acreditación y la certificación de la calidad en el ámbito de las universidades y de los centros de enseñanza superior de Catalunya (titulaciones, profesorado, centros y servicios).

AQU Catalunya es miembro fundador y de pleno derecho de la *European Association for Quality Assurance in Higher Education (ENQA)*, y ha sido una de las tres primeras agencias incluidas en el *European Quality Assurance Register for Higher Education (EQAR)*. Es miembro

de la Red Española de Agencias de Calidad Universitaria (REACU) y de la *International Network for Quality Assurance Agencies in Higher Education* (INQAAHE), de la que ejerce el Secretariado desde 2013. Ha sido la primera agencia de calidad europea certificada con la norma ISO.

El año 2015, con la aprobación de la Ley 15/2015, de 21 de julio, de la Agencia para la Calidad del Sistema Universitario de Cataluña, AQU Catalunya queda regulada y se fortalece como instrumento principal para la mejora del sistema universitario catalán con funciones actualizadas, una estructura más flexible y una mejora del cumplimiento de los estándares europeos en cuanto a la independencia. Asimismo, refuerza la vertiente internacional y establece un sistema de nombramiento del director por medio de un proceso abierto y competitivo de libre competencia.

AQU Catalunya cuenta con un sitio web donde se puede apreciar un buen diseño de interfaz, además de ser responsive lo que da posibilidad de navegación con mayor comodidad a los usuarios que accedan a la web desde un dispositivo portátil. En la web no existe un apartado en el que los usuarios puedan hacerle llegar al equipo de soporte alguna inquietud. Da la posibilidad que hacer revisiones de otras instituciones y programas acreditados con anterioridad por la propia institución. Pone a disposición un cronograma de las futuras evaluaciones. Permite que los usuarios envíen mensajes al equipo de soporte directamente comentando alguna inquietud. El sitio presenta un espacio donde busca mantener al usuario actualizado con los resultados de la institución. La web tiene enlaces directos a las redes sociales de la institución, pero en este caso no hay opciones de RSS. En el sitio se publican noticias relevantes sobre la institución. La institución hace publicaciones de artículos la temática. Puntos a destacar de este sitio web es que presenta un directorio con información relevante de las personas que integran la institución, además de poner a disposición de la comunidad una base de datos bien detallada de todos los procesos de evaluación y acreditación en los que ha participado [19].

1.2.4 Agencia Suiza de Acreditación y Garantía de Calidad

La Agencia Suiza de Acreditación y Garantía de Calidad bajo la autoridad del Consejo de Acreditación de Suiza. La AAQ lleva a cabo procedimientos de acreditación institucional en instituciones de educación superior suizas de acuerdo con la Ley Federal de Financiamiento y Coordinación del Sector de Educación Superior (HEdA).

La Agencia de Acreditación y Garantía de Calidad protege y promueve la calidad de la enseñanza y la investigación en las universidades de Suiza. Es independiente, utiliza métodos reconocidos internacionalmente y puede aprovechar el conocimiento y la experiencia de los principales expertos. La AAQ desarrolla pautas y estándares de calidad, realiza procedimientos de acreditación y evaluación y trabaja a nivel internacional. Percibe su papel como socio externo para el aseguramiento de la calidad y el desarrollo, y proporciona una gama de servicios de apoyo. La AAQ está bajo la autoridad del Consejo de Acreditación de Suiza.

AAQ cuenta con un sitio web donde se puede apreciar un buen diseño de interfaz, además de ser responsive lo que da posibilidad de navegación con mayor comodidad a los accedan a la web desde un dispositivo portátil. Se le da la posibilidad a los usuarios que puedan descargar los documentos normativos de los procesos de evaluación y acreditación que lleva a cabo la institución. Es posible hacer revisiones de otras instituciones y programas acreditados con anterioridad por la propia institución desde el sitio web. No están disponible información alguna acerca de las futuras evaluaciones por parte de la institución. Permite que los usuarios envíen mensajes al equipo de soporte directamente comentando alguna inquietud. No existe una sección en el sitio donde el usuario pueda buscar información con el objetivo de mantenerse al tanto de los resultados de la institución. La institución no tiene presencia en las redes sociales, y tampoco presenta una alternativa con suscripción RSS. En el sitio se publican noticias relevantes sobre la institución. La institución hace publicaciones de artículos la temática. Punto a destacar de este sitio web es lo bien internacionalizado que esta, pone a disposición de los usuarios hasta 4 opciones de idiomas, estas son: alemán, francés, italiano e inglés [20].

1.2.5 Quality Assurance Agency for Higher Education

La *Quality Assurance Agency for Higher Education* (QAA) es el organismo independiente que verifica los estándares y la calidad de la educación superior del Reino Unido [21]. Lleva a cabo revisiones de evaluación de la calidad, desarrolla puntos de referencia y orientación para los proveedores, y realiza o encarga investigaciones sobre temas relevantes.

QAA comprueba cómo las universidades, colegios y proveedores alternativos de educación superior del Reino Unido mantienen sus estándares y calidad académicos. Lo hace a través de una revisión por pares externos. Los revisores comprueban que se cumplan las expectativas básicas del Código de Calidad [22], acordado y reconocido por el sector de la educación superior del Reino Unido. También brinda asesoramiento al Consejo Privado del Reino Unido sobre las solicitudes de las instituciones de poderes para otorgar títulos y el derecho a ser llamada universidad.

QAA cuenta con un sitio web donde se puede apreciar un buen diseño de interfaz, además de ser responsive lo que da posibilidad de navegación con mayor comodidad a los accedan a la web desde un dispositivo portátil. Se le da la posibilidad a los usuarios que puedan acceder a los documentos normativos de los procesos de evaluación y acreditación de la institución. No es posible hacer revisiones de otras instituciones y programas acreditados con anterioridad por la propia institución desde el sitio web. Pone a disposición de los usuarios un cronograma de las futuras evaluaciones. Permite que los usuarios envíen mensajes al equipo de soporte comentando alguna inquietud. En la existe una sección en el sitio donde el usuario pueda buscar información con el objetivo de mantenerse al tanto de los resultados de la institución. La institución tiene presencia en las redes sociales, pero no tiene opción de suscripción RSS. En el sitio se publican noticias relevantes sobre la institución. La institución hace publicaciones de artículos la temática. Punto a destacar de este sitio web es que tienen un fórum a disposición de los usuarios donde se le permite debatir y comentar sobre temáticas vinculadas a la institución [21].

Tabla 1: Análisis de herramientas homólogas

Variables	ABET	COPAES	AQU	AAQ	QAA
Descarga de documentos normativos	✓	✓	✓	✓	✓
Búsquedas de instituciones y programas acreditados	✓	✓	✓	✓	×
Cronograma de evaluaciones	✓	×	✓	×	✓
Comunicación con los usuarios	✓	✓	×	✓	✓
Boletines informativos	✓	×	✓	×	✓
Redes sociales y	✓	✓	✓	×	✓

RSS					
Publicación de noticias	✓	×	✓	✓	✓
Publicación de artículos sobre la temática	✓	×	✓	✓	✓

De manera general todas las instituciones utilizan un sitio web para potenciar altos índices de calidad en la educación superior de cada uno de los países que representan. Los sitios cuentan con un diseño agradable y sencillo que permite a los usuarios que los visitan encontrar información sobre la temática sin ser expertos en el uso de aplicaciones informáticas. Se pudo constatar que es muy útil la publicación de los documentos normativos de la temática que, aunque tienen elementos comunes, tienen sus particularidades para cada región. Estos sitios permiten a las instituciones divulgar actividades de superación, cronograma de evaluaciones, investigaciones sobre la temática y permiten en su mayoría que los usuarios puedan interactuar con las autoridades competentes y aclarar dudas sobre los procesos de acreditación. A partir de este estudio se identificaron un conjunto de funcionalidades que son necesarias incluir en el sitio de la Junta de Acreditación Nacional de Cuba, para promover el trabajo que realizan y fomentar una cultura de calidad en la Educación Superior cubana.

1.3 Tecnologías para el desarrollo del portal web

En este epígrafe se detallarán las tecnologías a utilizar en el desarrollo de la solución informática. Se definirá que *frameworks* se emplearan, la arquitectura del sistema, las herramientas y la metodología a seguir para lograr un producto de calidad.

Angular 12

Hoy en día existe un gran mercado en el sector del desarrollo web, y con esto se han creado distintas tecnologías para la creación de plataformas en línea. Para entender qué es Angular primero debemos saber la arquitectura de una plataforma web. Al haber interacción entre los equipos de los usuarios y los servidores en internet, se debe tomar en cuenta las distintas tecnologías para ambos extremos, el lado del usuario se le conoce como “*frontend*” y el lado del servidor se le conoce como “*backend*”.

Angular es un *framework* mantenido por Google para el desarrollo de aplicaciones web del lado del cliente, su estructura está hecha en base a TypeScript, el cual es una versión avanzada de *JavaScript*. Comenzó en el 2010 bajo el nombre de AngularJS. proporcionaba controladores y directivas para dividir el código en JavaScript para poder manejar la interfaz gráfica de usuario y las peticiones al servidor por aparte. Para la segunda versión de AngularJS, el equipo desarrollador decidió cambiar la nomenclatura del *framework* a simplemente Angular.

Uno de los objetivos de Angular es fortalecer la estructura MVC en el desarrollo web (Modelo-Vista-Controlador) y las SPA (*Single Page Application*). El MVC es una arquitectura que se encarga de separar la lógica del código en una aplicación, en donde la vista es la interfaz gráfica de usuario y esta cambia de acuerdo a un controlador que gestiona el contenido en ella. Las SPA son páginas que cargan únicamente un archivo HTML en el cliente y que su contenido cambia dinámicamente a medida el usuario interactúa con ella [23].

Se decide emplear Angular como *framework* del lado del cliente por:

- **Reusabilidad:** La estructura de Angular basada en componentes hace que los componentes sean altamente reutilizables en toda la aplicación.
- **Pruebas unitarias simplificadas:** Al ser independientes entre sí, los componentes hacen que las pruebas unitarias sean mucho más fáciles.
- **Mejor legibilidad:** La coherencia en la codificación hace que la lectura del código sea una tarea fácil para los nuevos desarrolladores. Lo que aumenta su productividad.
- **Facilidad de mantenimiento:** Los componentes pueden ser reemplazados con mejores implementaciones. En pocas palabras, permite el mantenimiento y actualización eficiente del código [24].

NestJS 8

NestJS (Nest) fue desarrollado por Kamil Myśliwiec con la misión de crear aplicaciones *backend* de Node.js escalables y efectivas. El marco es compatible con JavaScript y el lenguaje TypeScript. Además, mezcla componentes de FP (Programación funcional), OOP (Programación orientada a objetos) y FRP (Programación reactiva funcional). Proporciona un nivel de abstracción por encima de estos marcos comunes de Node.js (Express / Fastify), pero también expone sus API directamente al desarrollador. Esto les da a los desarrolladores la libertad de usar la gran cantidad de módulos de terceros que están disponibles para la plataforma subyacente [25].

Se decide emplear NestJS como framework del lado del servidor por:

Arquitectura:

- Es un marco con muchas opiniones.
- Las aplicaciones NestJS están escritas en Typescript.
- Tiene una arquitectura sólida similar a Angular.
- Ofrece una interfaz de línea de comandos (CLI) que permite a los desarrolladores ampliar fácilmente el proyecto.

Rendimiento:

- Tiene una alta capacidad de respuesta en términos de rendimiento.
- Es altamente escalable.

Pruebas:

- Su iniciador de pruebas automatizado aumenta la simplicidad de las pruebas [26].

1.3.2 Lenguajes de programación**HTML5**

El HTML es el principal componente de una página web, sin un documento HTML no existe una página web. Es el lenguaje que le da la estructura y organización a una página web. Es el *esqueleto* de una página. Dentro del documento HTML se definen todos los elementos que componen una página, cualquier información adicional o recurso externo que esta vaya a utilizar [27].

Ventajas:

- Nuevas etiquetas semánticas (los buscadores deben poder distinguir que es importante y que no dentro del código de una página web).
- Multimedia sin *plugins* (audio, video y animaciones de acceso universal).
- Formularios más usables y que validan automáticamente lo ingresado por el usuario, reduciendo – idealmente, veremos que aún falta implementar bastante- el uso de JavaScript [28].

CSS3

CSS es la sigla de *Cascading Style Sheets* (hojas de estilo en cascada). Es el lenguaje utilizado para definir la apariencia de los elementos definidos en un documento HTML. El CSS no es tanto un lenguaje de programación como lo es de reglas, una expresión básica en CSS se compone de dos partes, un selector y un conjunto de reglas. Cada regla a su vez se compone de dos fragmentos, una propiedad y un valor. Las reglas van al interior de corchetes, se separa la propiedad del valor con dos puntos, y las reglas se separan entre si con un punto y coma. Existen un gran número de propiedades que abarcan desde el tamaño y tipo de fuente hasta transformaciones geométricas y animaciones simples [27].

Ventajas:

- Uso de cualquier tipografía con `@font-face` de CSS3 (aunque el usuario no la tenga instalada en su dispositivo).
- Nuevos modelos de color que incluyen translucidez y degradado sin imágenes.
- Efectos visuales como sombras en textos y cajas, bordes redondeados y con imágenes, múltiples fondos para un mismo elemento.
- Técnicas para manipular la presentación y el movimiento de elementos con CSS, como transformaciones, transiciones y animaciones.
- Mejora en la sintaxis CSS gracias a nuevos selectores [28].

Sass 1.38.2

Sass es un metalenguaje de Hojas de estilo en cascada. Es un lenguaje script que traduce todo su contenido insertado en CSS. Con Sass conseguimos tener nuestro código mucho más organizado (estilo en cascada). Esto nos permite no tener “mil parches” a la hora de realizar futuras modificaciones en nuestro código y tener todo el código de un sector en concreto junto. Otra de las ventajas que incorpora Sass es que permite la creación de variables. Estas variables sobre todo las utilizamos para definir los colores y las fuentes de nuestra web, con esto evitamos tener que recordar el color en hexadecimal o copiarlo cada vez que lo necesitemos [29].

Ventajas:

- Reduce el tiempo para crear y mantener el CSS.
- Permite tener una organización modular de los estilos, lo cual es vital para proyectos grandes.

- Proporciona estructuras avanzadas propias de los lenguajes de programación, como variables, listas, funciones y estructuras de control.
- Permite generar distintos tipos de salida, comprimida, normal o minimizada, trabajando tanto en desarrollo como en producción, además se hace una forma muy fácil [30].

JavaScript ES2017

JavaScript (JS) es un lenguaje de programación interpretado que está estrechamente integrado con la web y es capaz de manipular los elementos del DOM (*Document Object Model*), esto es cada elemento que compone una página web. Todos los navegadores soportan el uso de JavaScript y su estructura ha tenido muy pocos cambios y adiciones desde su creación en 1995. El propósito de JavaScript es permitir la creación de aplicaciones funcionales para la web, no solo documentos de texto relacionados por hipervínculos. JS es un lenguaje de programación orientado a objetos que se ejecuta una vez al cargar la página web o al ser llamado por algún evento definido. [23].

Ventajas:

- Nuevas posibilidades interactivas con el nuevo DOM y sus nuevas funciones.
- Geolocalización del usuario.
- Posibilidad de arrastrar y soltar cualquier elemento HTML.
- Animaciones interactivas con *canvas*.
- Almacenamiento de datos con *Session Storage* y *Local Storage*.
- Trabajo offline con Cache de Aplicaciones, *File System API* y base de datos del lado del cliente.
- Comunicación con el servidor Web Socket.
- Eficiencia mediante hilos (*Web Workers*) [28].

Typescript 4.4.3

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft, el cual cuenta con herramientas de programación orientada a objetos, muy favorable si se tienen proyectos grandes. Anders Hejlsberg, arquitecto principal del desarrollo del lenguaje de programación C#, es el principal participante en el desarrollo de este lenguaje. TypeScript convierte su código en JavaScript común. Es llamado también *Superset* de JavaScript, lo que significa que, si el navegador está basado en Javascript, este nunca llegará a saber que el código original fue realizado con TypeScript y ejecutará el JavaScript como lenguaje original [23].

Ventajas:

- Añade el uso un tipado muy estricto, en lugar del tipado dinámico de JavaScript, para intentar solucionar una serie de problemas. Aunque estos tipados no son requeridos, es recomendable utilizarlos para tener un código mucho más limpio.
- Añade la posibilidad de poder utilizar interfaces, para poder definir nuestros propios tipos o intentar aplicar programación orientada objetos luego.
- También añade la funcionalidad de genérica, que permite poder definir funciones que sean reutilizables, independientemente del tipo de datos que vayamos a tratar [31].

UML 2.5

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación. Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

Ventajas:

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos [32].

1.3.3 Arquitectura del sistema**Servidor Web**

Un servidor web que se ejecuta en un ordenador se mantiene a la espera de peticiones por parte de un cliente (un navegador web o un programa que hace una llamada a un servicio web). Cuando el servidor recibe una petición, responde adecuadamente mediante una página web que se exhibirá en el navegador, o bien mostrará el mensaje de error correspondiente [27].

Frontend

Frontend se encarga de estilizar la página de tal manera que la misma pueda presentar la información de forma agradable para el usuario. La persona responsable del *Frontend*, debe de conocer las técnicas de experiencia de usuario para brindar una mejor interacción entre la persona y la página que visita, así mismo debe tener conocimientos de diseño de Interacción para colocar los elementos de tal manera que el usuario las pueda ubicar de forma rápida y cómoda. Existen muchas tecnologías relacionadas a estos tres lenguajes que deben ser conocidas por el desarrollador *Frontend*. Por ejemplo, para JavaScript existen Angular y BackboneJS, los cuales se pueden apoyar en librerías como animate.css, JQuery y otras. También existen lenguajes de transferencia de información como XML, JSON y Ajax para hacer solicitudes al servidor sin necesidad de refrescar la página completa [35].

Backend

Se denomina *Backend* a la capa de acceso a los datos de un software que no es accesible para el usuario final. Además, esta capa contiene toda la lógica de la aplicación que maneja los datos. Cabe destacar que los datos de una aplicación se encuentran almacenados en una base de datos dentro de un servidor.

La persona encargada del *Backend* debe tener conocimientos, según el lugar donde trabaje, de los lenguajes del lado del servidor, como ser: Java, C#, PHP, Node.JS, entre otros. Además, de aquellos que interactúan con la base de datos, como ser: MySQL, PostgreSQL, SQLServer, MongoDB, entre otras [35].

API

Una API de REST, o API de RESTfull, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTfull. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST).

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta). Por ejemplo, el diseño de una API de servicio meteorológico podría requerir que el usuario escribiera un código postal y que el productor diera una respuesta en dos partes: la primera sería la temperatura máxima y la segunda, la mínima [33].

Base de datos

Es una colección de archivos relacionados que almacenan tanto una representación abstracta del dominio de un problema del mundo real cuyo manejo resulta de interés para una organización, como los datos correspondientes a la información acerca del mismo. Tanto la representación como los datos están sujetos a una serie de restricciones, las cuales forman parte del dominio del problema y cuya descripción esta también almacenada en esos archivos [36].

1.3.4 Herramientas

WebStorm 2021.2.2

WebStorm es un entorno de desarrollo integrado para codificar en JavaScript y sus tecnologías relacionadas. Al igual que IntelliJ IDEA y otros IDE de JetBrains, WebStorm hará que su experiencia de desarrollo sea más agradable, automatizando el trabajo de rutina y ayudándole a manejar tareas complejas con facilidad.

Es directamente compatible con JavaScript, TypeScript, React, React Native, Electron, Vue, Angular, Node.js, HTML, hojas de estilo y muchas otras tecnologías. Puede comenzar a programar directamente, sin tener que pensar en instalar ni mantener ningún complemento. Comprende perfectamente la estructura de su proyecto y le puede ayudar en todos los aspectos de la escritura de código. Autocompletará su código, detectará y sugerirá correcciones para los errores y redundancias, y le ayudará a refactorizar el código con seguridad [34].

Postman 7.36

Postman es una plataforma de colaboración para el desarrollo de API. Las funciones de Postman simplifican cada paso de la creación de una API y agilizan la colaboración para que pueda crear mejores API, más rápido [36].

Se trata de una herramienta dirigida a desarrolladores web que permite realizar peticiones HTTP a cualquier API. Postman es muy útil a la hora de programar y hacer pruebas, puesto que nos ofrece la posibilidad de comprobar el correcto funcionamiento de nuestros desarrollos. Postman no es una herramienta exclusiva para profesionales del entorno web, de hecho, es muy útil para todo aquel que tenga que interactuar con una API [37].

Postgres 13.3

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto con más de 30 años de desarrollo activo que le ha ganado una sólida reputación por su fiabilidad, solidez de funciones y rendimiento. Se puede encontrar una gran cantidad de información que describe cómo instalar y usar PostgreSQL a través de la documentación oficial. La comunidad de PostgreSQL proporciona muchos lugares útiles para familiarizarse con la tecnología, descubrir cómo funciona y encontrar oportunidades profesionales [38].

Navicat Premium 15

Navicat Premium es una herramienta de desarrollo de bases de datos que le permite conectarse simultáneamente a bases de datos MySQL, MariaDB, MongoDB, SQL Server, Oracle, PostgreSQL y SQLite desde una sola aplicación. Compatible con bases de datos *cloud* como Amazon RDS, Amazon Aurora, Amazon Redshift, Microsoft Azure, Oracle Cloud, Google Cloud y MongoDB Atlas. Podrá crear, administrar y mantener sus bases de datos de forma rápida y fácil [39].

Visual Paradigm 15.2

Visual Paradigm es un proveedor líder y mundialmente reconocido de soluciones de software de transformación empresarial y de TI. Permite a las organizaciones mejorar la agilidad empresarial y de TI y fomentar la innovación a través de estándares abiertos populares. Los productos galardonados cuentan con la confianza de más de 320.000 usuarios en empresas que van desde pequeñas empresas, consultores hasta organizaciones de primera línea, universidades y unidades gubernamentales de todo el mundo.

Permite a su equipo gestionar la complejidad de la transformación empresarial para hacer frente a los mercados, las tecnologías y los requisitos normativos que cambian rápidamente. Es una solución integral ideal para la planificación de la arquitectura empresarial y la transformación empresarial, la gestión de proyectos y el desarrollo ágil de software, para que su empresa pueda mantener el control y fomentar el crecimiento [40].

1.3.5 Metodología de desarrollo de software

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del *Rational Unified Process* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas. (*Test Driven Development* - TDD en inglés)
- Modelado ágil.
- Gestión de Cambios Ágil.
- Refactorización de Base de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva. De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre.

Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a 1 la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11.

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos (CUN, DPN o MC) y existen tres formas de encapsular los requisitos (CUS, HU, DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma [41]:

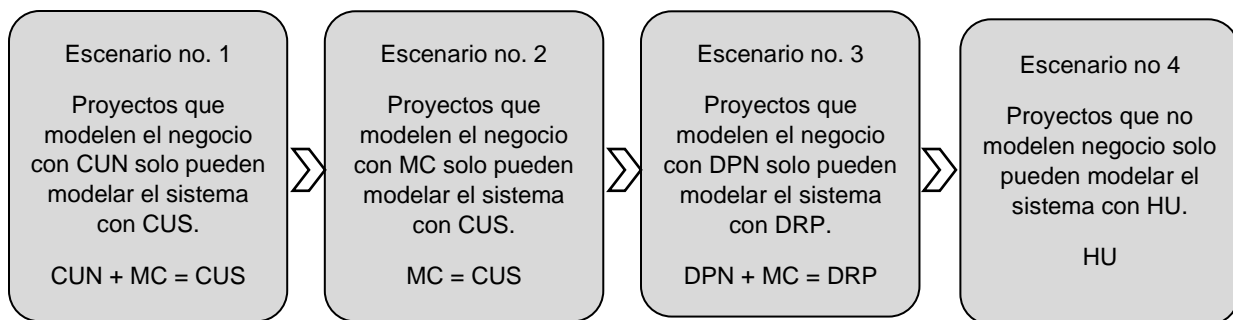


Figura 1: Representación gráfica de los escenarios de la metodología AUP-UCI

Para el desarrollo de la solución informática se decide hacer uso del escenario No. 4. El cual aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. En este caso se busca gestionar y presentar la información de la Junta de Acreditación Nacional relacionada con el proceso de evaluación de la calidad de la educación superior en Cuba.

1.4 Conclusiones del capítulo

Con el estudio de este capítulo se puede concluir que:

- El proceso de acreditación de las instituciones de la enseñanza de la educación superior es de vital importancia en la búsqueda incesante por la excelencia educacional.
- El estudio de homólogos sentó las bases de los requisitos funcionales y no funcionales que deberá tener la propuesta de solución, garantizado que este a la altura de los estándares del mercado.
- Tomando como referencia las tecnologías para el desarrollo web más actuales, se decide emplear para el desarrollo del sitio web, en el lado del cliente Angular y del lado del servidor NestJS.
- La definición de la metodología AUP vUCI en su escenario 4 determinar que artefactos ingenieriles se deberán elaborar como parte de la descripción de la aplicación.

Capítulo 2. Diseño del portal web para la Junta de Acreditación Nacional

El diseño de un producto es fundamental para un desarrollo exitoso del mismo, en este se desarrollan, revisan y documentan los requisitos del producto final y los detalles para su implementación. En el capítulo se describe la propuesta de solución para el problema de la investigación. Mediante el modelo conceptual se describe la estructura que se desea representar en el portal web a desarrollar. Describiendo los principales conceptos del entorno que serán objeto de análisis para la realización de la fase de Análisis y Diseño. Posteriormente se realiza el levantamiento de requisitos funcionales y no funcionales para darle cumplimiento al objetivo general de la investigación. Además, se muestran los diagramas de clases del diseño con estereotipos web, por medio de los cuales se representa la estructura estática del sistema y los patrones de arquitectura y diseño utilizados en el sistema. Seguido se detallarán los diagramas de iteración lo que permitirá ilustrar la interacción secuencial de los componentes del portal web. Por último, se expone el modelo de datos que se empleará y el modelo de despliegue que define la estructura que tendrá el producto final una vez culminado.

2.1 Propuesta de portal web para la Junta de Acreditación Nacional

Tomando en cuenta las herramientas y tecnologías seleccionadas en el capítulo anterior se propone desarrollar un Portal Web para la JAN con el objetivo de fortalecer la comunicación de este organismo y el proceso de evaluación-acreditación de las instituciones de la educación superior de Cuba. El portal cumplirá con la relación de requisitos funcionales y no funcionales que se definirán a continuación en el capítulo. Dándole la posibilidad a los usuarios de tener disponibles publicaciones donde se detalle la actualidad del organismo, poder comentar dichas publicaciones, documentación de los procesos de acreditación, conocer los cronogramas de evaluación y hacer búsquedas sobre las instituciones que han sido acreditadas o los programas de acreditación disponibles.

2.2 Modelo Conceptual

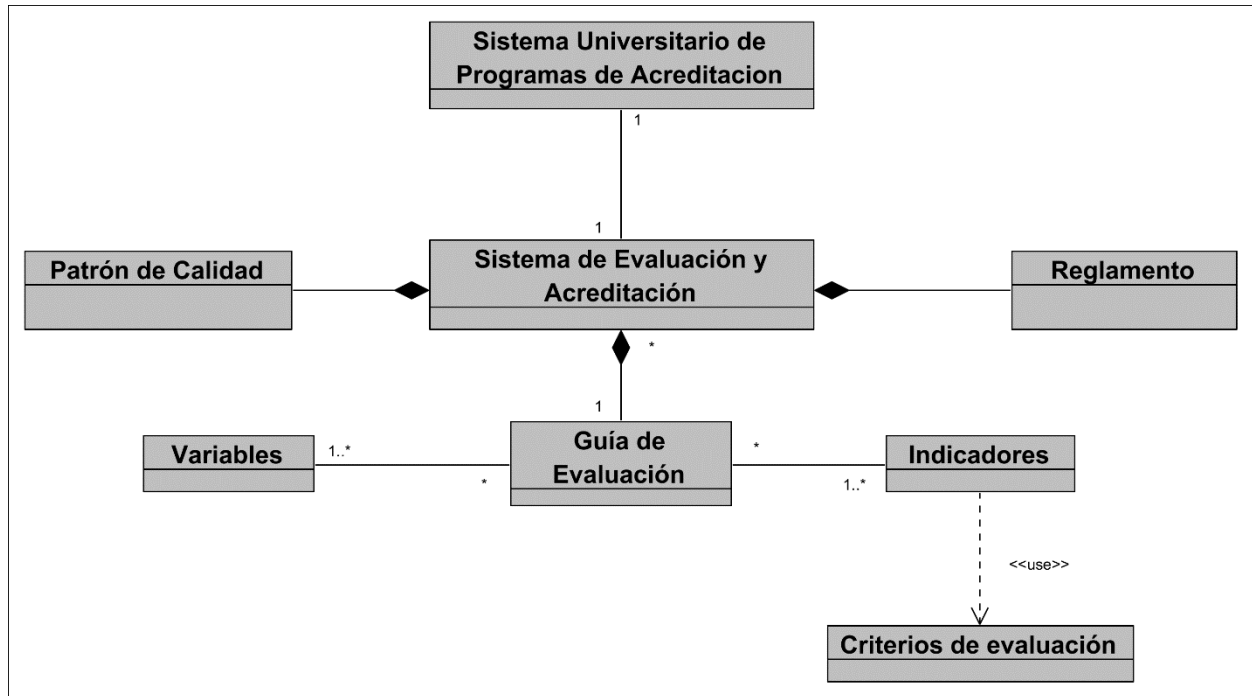


Figura 2: Modelo Conceptual

Este modelo representa el sistema de acreditación de todo el sistema cubano de forma general. El Sistema de Evaluación y Acreditación de Programas e Instituciones de Educación Superior en Cuba (SEAES), está conformado por cinco subsistemas, ellos son: de programas de maestría (SEA-M), de carreras universitarias (SEA-CU), de instituciones (SEA-IES), de programas de doctorado (SEA-Dr) y de programas de especialidades de posgrado (SEA-EP); implementados en este orden mediante resoluciones que norman cada proceso en particular. Cada subsistema cuenta con un patrón una guía y un reglamento.

2.3 Especificación de requisitos

Los requisitos para un sistema de software determinan lo que hará el sistema y definen las restricciones de su operación e implementación. Se pueden clasificar en: funcionales y no funcionales. Partiendo de la descripción de clases representadas en el Modelo de Dominio y las necesidades del cliente, fueron determinados los requisitos funcionales y no funcionales del sistema [42].

2.3.1 Requisitos funcionales

Son capacidades o condiciones que el sistema debe cumplir, se obtienen las de las actividades que serán objeto de automatización. Estas actividades no son exactamente los requisitos

funcionales, pero sí son el punto de partida para identificar qué debe hacer el sistema. Los requisitos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen [42].

Tabla 2: Especificación de Requisitos Funcionales

N.º	Nombre	Descripción	P	C
RF#1	Subir Archivo	Permitir al usuario subir ficheros al sistema	Media	Alta
RF#2	Eliminar Archivo	Permitir al usuario eliminar uno o varios de los archivos que se han subido al sistema.	Media	Alta
RF#3	Actualizar Archivo	Permitir al usuario actualizar un archivo subido al sistema.	Baja	Baja
RF#4	Mostrar Archivo	Permitir al usuario ver la información específica de uno de los archivos subidos al sistema.	Media	Baja
RF#5	Listar Archivos	Permitir al usuario ver un listado de los archivos subidos al sistema.	Baja	Baja
RF#6	Crear Evento	Permitir al usuario registrar un evento en el calendario, que puede o no estar vinculado con una institución.	Alta	Media
RF#7	Actualizar Evento	Permitir al usuario actualizar los datos de un evento registrado en el sistema.	Media	Baja
RF#8	Mostrar Evento	Permitir al usuario ver los datos de un evento en específico registrado antes en el sistema.	Alta	Baja
RF#9	Eliminar Evento	Permitir al usuario eliminar unos de los eventos registrados antes en el sistema.	Media	Baja
RF#10	Listar Eventos	Permitir al usuario ver un listado de todos los eventos registrados en el sistema.	Media	Baja
RF#11	Crear Publicación	Permitir al usuario crear una publicación en el sistema.	Alta	Media
RF#12	Actualizar Publicación	Permitir al usuario actualizar una publicación registrada con anterioridad en el sistema.	Media	Baja
RF#13	Eliminar Publicación	Permitir al usuario eliminar una de las publicaciones que se registró antes en el sistema.	Media	Baja
RF#14	Listar Publicaciones	Permitir al usuario ver el listado de las publicaciones registradas en el sistema.	Media	Baja
RF#15	Mostrar Publicación	Permitir al usuario ver una publicación que haya	Alta	Media

		sido registrada en el sistema.		
RF#16	Buscar Publicación	Permitir al usuario hacer una búsqueda global entre todas las publicaciones registradas en el sistema.	Media	Alta
RF#17	Crear Comentario	Permitir al usuario publicar un comentario sobre una publicación.	Baja	Baja
RF#18	Eliminar Comentario	Permitir al usuario con los permisos suficientes eliminar un comentario publicado con anterioridad.	Media	Baja
RF#19	Listar Comentarios	Permitir al usuario ver los comentarios publicaciones de una publicación en específico.	Baja	Baja
RF#20	Crear Etiqueta	Permitir al usuario crear una etiqueta.	Baja	Baja
RF#21	Actualizar Etiqueta	Permitir al usuario actualizar los datos de una etiqueta.	Baja	Baja
RF#22	Eliminar Etiqueta	Permitir al usuario eliminar una etiqueta registrada antes en el sistema.	Baja	Baja
RF#23	Listar Etiqueta	Permitir al usuario ver un listado de las etiquetas registradas en el sistema.	Baja	Baja
RF#24	Mostrar Etiqueta	Permitir al usuario ver los datos de una etiqueta registrada en el sistema.	Baja	Baja
RF#25	Registrar Institución	Permitir al usuario registrar una institución en el sistema.	Alta	Media
RF#26	Modificar Institución	Permitir al usuario modificar una de las instituciones registradas en el sistema.	Alta	Media
RF#27	Listar Instituciones	Permitir al usuario ver un listado de todas las instituciones registradas en el sistema.	Media	Baja
RF#28	Mostrar Institución	Permitir al usuario ver los datos de una de las instituciones registradas.	Media	Baja
RF#29	Buscar Institución	Permitir al usuario realizar una búsqueda sobre todas las instituciones que estén registradas en el sistema.	Alta	Alta
RF#30	Eliminar Institución	Permitir al usuario al usuario eliminar una institución que este registrada en el sistema.	Alta	Baja
RF#31	Registrar Usuario	Permitir al usuario con los permisos suficientes registrar a otro usuario en el sistema.	Alta	Media
RF#32	Modificar Usuario	Permitir al usuario con los permisos suficientes	Alta	Baja

		modificar la información de otro usuario o el propio usuario puede modificar su información.		
RF#33	Listar Usuarios	Permitir al usuario con los permisos suficientes ver un listado de los usuarios registrados en el sistema.	Media	Baja
RF#34	Mostrar Usuario	Permitir al usuario con los permisos suficientes ver la información de otro usuario o los datos del propio usuario.	Baja	Baja
RF#35	Autenticar Usuario	Permitir al usuario iniciar sesión en el sistema	Alta	Baja
RF#36	Cerrar sesión de usuario	Permitir al usuario cerrar su sesión del sistema	Alta	Baja
RF#37	Buscar Usuario	Permitir al usuario con los permisos suficientes realizar una búsqueda sobre todos los usuarios registrados en el sistema.	Alta	Alta
RF#38	Activar Usuario	Permitir al usuario con los permisos suficientes activar un usuario registrado en el sistema.	Alta	Baja
RF#39	Desactivar Usuario	Permitir al usuario con los permisos suficientes desactivar un usuario registrado en el sistema.	Alta	Baja
RF#40	Cambiar contraseña	Permitir al usuario con los permisos suficientes cambiar la contraseña de otro usuario o el propio usuario cambiar su propia contraseña.	Alta	Media
RF#41	Recuperar de contraseña	Permitir al usuario recuperar su contraseña.	Alta	Media
RF#42	Eliminar Usuario	Permitir al usuario con los permisos suficientes eliminar a otro usuario registrado en el sistema.	Alta	Baja
RF#43	Crear Rol	Permitir al usuario con los permisos suficientes crear un rol en el sistema.	Alta	Baja
RF#44	Modificar Rol	Permitir al usuario con los permisos suficientes modificar un rol registrado antes en el sistema.	Media	Media
RF#45	Listar Roles	Permitir al usuario con los permisos suficientes ver el listado de roles registrados en el sistema.	Media	Baja
RF#46	Mostrar Rol	Permitir al usuario con los permisos suficientes ver la información perteneciente a un rol registrado en el sistema.	Media	Baja
RF#47	Asignar Rol	Permitir al usuario con los permisos suficientes asignar un rol a un usuario registrado en el	Alta	Media

		sistema.		
RF#48	Quitar Rol	Permitir al usuario con los permisos suficientes quitarle un rol asignado a un usuario registrado en el sistema.	Alta	Media
RF#49	Eliminar Rol	Permitir al usuario con los permisos suficientes eliminar un rol del sistema.	Alta	Alta
RF#50	Registrar Permiso	Permitir al usuario con los permisos suficientes registrar un permiso en el sistema.	Alta	Baja
RF#51	Modificar Permiso	Permitir al usuario con los permisos suficientes modificar los datos de un permiso registrado en el sistema.	Media	Media
RF#52	Listar Permisos	Permitir al usuario con los permisos suficientes ver el listado de los permisos registrados en el sistema.	Media	Baja
RF#53	Mostrar Permiso	Permitir al usuario con los permisos suficientes ver los datos de un permiso en específico registrado en el sistema.	Media	Media
RF#54	Asignar Permiso	Permitir al usuario con los permisos suficientes asignar un permiso a un rol registrado en el sistema.	Alta	Media
RF#55	Quitar Permiso	Permitir al usuario con los permisos suficientes quitar un permiso a un rol registrado en el sistema	Alta	Media
RF#56	Eliminar Permiso	Permitir al usuario con los permisos suficientes eliminar un permiso del sistema.	Alta	Alta
RF#57	Registrar Suscripción	Permitir al usuario registrarse en los boletines informativos del portal web	Baja	Media
RF#58	Modificar Suscripción	Permitir al usuario modificar las configuraciones de su suscripción	Baja	Media
RF#59	Listar Suscripciones	Permitir al usuario ver un listado de las suscripciones que están registradas en el sistema	Baja	Baja
RF#60	Mostrar Suscripción	Permitir al usuario ver los datos de una suscripción en específico	Baja	Baja
RF#61	Activar Suscripción	Permitir al usuario activar una suscripción registrada en el sistema	Baja	Baja

RF#62	Desactivar Suscripción	Permitir al usuario desactivar una suscripción registrada en el sistema	Baja	Baja
RF#63	Registrar Inquietud	Permitir al usuario registrar una inquietud en el sistema	Media	Baja
RF#64	Responder Inquietud	Permitir al administrador responder una inquietud registrada por otro usuario	Media	Baja
RF#65	Listar Inquietudes	Permitir al administrador ver un listado de las inquietudes registradas en el sistema	Media	Baja
RF#66	Mostrar Inquietud	Permitir al administrador ver los detalles de una inquietud registrada en el sistema	Baja	Baja
RF#67	Eliminar Inquietud	Permitir al administrador eliminar una inquietud del sistema	Baja	Baja

2.3.2 Requisitos no funcionales

Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación [42].

Definición de requisitos no funcionales tomando como guía el Estándar de Calidad Internacional ISO/IEC 9126-1.

Tabla 3: Especificación de Requisitos No Funcionales

N.º	Descripción
Confiabilidad	
RnF#1	Los usuarios tendrán disponibles funcionalidades en el sistema acorde al rol que tengan asignados.
RnF#2	En el sistema solo existirá un super usuario.
RnF#3	El registro de nuevos usuarios debe terminar con la verificación del correo electrónico.
RnF#4	El sistema debe permitir recuperar la contraseña a los usuarios.
RnF#5	La contraseña del usuario debe tener un mínimo de 8 caracteres en combinación de letras, números y caracteres especiales.

Portabilidad	
RnF#6	Se podrá acceder al portal web desde cualquier dispositivo empleando cualquier navegador.
Usabilidad	
RnF#7	El sistema tendrá un diseño <i>responsive</i> que permitirá que se adapte a cualquier dispositivo.
RnF#8	La interacción con el sistema debe ser fácil e intuitiva para el usuario.
RnF#9	El contenido publicado en el portal debe estar bien estructurado y definido siguiendo una jerarquía.
RnF#10	El sistema debe mantener un diseño coherente en toda la web y sus distintos enlaces.
Mantenibilidad	
RnF#11	El sistema permitirá la incorporación de nuevas funcionalidades.
RnF#12	El sistema podrá recibir actualización que solventen cualquier tipo de error.
Eficiencia	
RnF#13	El sistema deberá permitir una concurrencia de al menos 100 usuarios.
RnF#14	Las peticiones al sistema deben durar un máximo de 5 segundos.

2.4 Historias de usuario

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Describen, en una o dos frases, una funcionalidad de *software* desde el punto de vista del usuario, con el lenguaje que éste emplearía. El foco está puesto en qué necesidades o problemas soluciona lo que se va a construir [43].

2.4.1 HU Crear Publicación

Tabla 4: HU Crear Publicación

Historia de Usuario	
Número: 1	Usuario: Administrador
Nombre historia: Crear Publicación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Yohan González Almaguer	

<p>Descripción:</p> <p>Como administrador se quiere registrar una nueva publicación en el sistema para mantener el portal web actualizado de información útil para los usuarios.</p>
<p>Validación:</p> <p>La interfaz debería dejar registrar una nueva publicación cuando se llenen los campos del título, contenido y resumen.</p>

2.4.2 HU Buscar Publicación

Tabla 5: HU Buscar Publicación

Historia de Usuario	
Número: 2	Usuario: Cliente
Nombre historia: Buscar Institución	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Yohan González Almaguer	
<p>Descripción:</p> <p>Como cliente se quiere realizar búsquedas sobre las instituciones que estén registradas en el portal web para facilitar el acceso a la información que se requiere.</p>	
<p>Validación:</p> <p>La interfaz debería realizar búsquedas solo cuando se haya insertado algún criterio de búsqueda y en caso de que la búsqueda sea vacía hacerle saber al usuario.</p>	

2.4.3 HU Responder Inquietud

Tabla 6: HU Responder Inquietud

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre historia: Responder Inquietud	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Yohan González Almaguer	
<p>Descripción:</p> <p>Como administrador se quiere dar respuesta a una inquietud registrada por un usuario para</p>	

potenciar la comunicación con los usuarios del portal web.

Validación:

La interfaz solo dejará responder inquietudes que no tengan respuesta, y solo se completará la respuesta si se completan todos los campos.

2.4 Diseño arquitectónico

Al diseñar una arquitectura de software se crean y representan componentes que interactúan entre sí, con responsabilidades específicas y se organizan de forma tal que se logren los requerimientos establecidos. Se puede partir con patrones de soluciones probados que se conocen con el nombre de estilos arquitectónicos, patrones arquitectónicos y patrones de diseño. Un estilo arquitectónico es un conjunto coordinado de restricciones arquitectónicas que regula las funciones/características de los elementos arquitectónicos y las relaciones permitidas entre estos dentro de cualquier arquitectura que se adapte a ese estilo [45].

2.4.1 Patrón arquitectónico

La arquitectura orientada a los servicios (SOA) es un tipo de diseño de software que permite reutilizar sus elementos gracias a las interfaces de servicios que se comunican a través de una red con un lenguaje común. Un servicio es una unidad autónoma de una o más funciones del software diseñada para realizar una tarea específica, como recuperar cierta información o ejecutar una operación. Contiene las integraciones de datos y código que se necesitan para llevar a cabo una función empresarial completa y diferenciada. Se puede acceder a él de forma remota e interactuar con él o actualizarlo de manera independiente. En otras palabras, la SOA integra los elementos del software que se implementan y se mantienen por separado, y permite que se comuniquen entre sí y trabajen en conjunto para formar aplicaciones de software en distintos sistemas [46].

Funcionamiento de la arquitectura orientada a los servicios

Dado que la SOA expone los servicios utilizando protocolos estándar de red para enviar solicitudes o acceder a los datos (p. ej., SOAP, JSON, ActiveMQ o Apache Thrift), no es necesario que los desarrolladores realicen las integraciones desde cero. De hecho, pueden utilizar los patrones llamados buses de servicios empresariales (ESB) para integrar un elemento centralizado y los sistemas de *backend*, y ponerlos a disposición de todos como interfaces de servicios. Asimismo, pueden reutilizar las funciones actuales, en lugar de tener que volver a crearlas.

En este tipo de arquitectura, los servicios se comunican por medio de un sistema "sin conexión directa". Se trata de un método para interconectar los elementos en un sistema o una red, de manera que puedan transmitir información o coordinar un proceso empresarial, mientras se reduce la dependencia entre ellos. Como consecuencia, se crea una nueva aplicación [46].

Ventajas frente al enfoque monolítico

- **Comercialización más rápida y mayor flexibilidad:** la posibilidad de reutilizar los servicios agiliza y simplifica el proceso de creación de las aplicaciones. Los desarrolladores no tienen que empezar siempre desde cero, como en el caso de las aplicaciones monolíticas.
- **Uso de la infraestructura heredada en los mercados nuevos:** la SOA permite que los desarrolladores tomen las funciones de una plataforma o un entorno y las amplíen e implementen en otros.
- **Reducción de los costos gracias a una mayor agilidad y un desarrollo más eficiente**
- **Mantenimiento sencillo:** dado que todos los servicios son autónomos e independientes, se puede modificar y actualizar cada uno cuando sea necesario, sin afectar al resto.
- **Escalabilidad:** la SOA permite que ejecutar los servicios en varios lenguajes de programación, servicios y plataformas, lo cual aumenta la escalabilidad de forma considerable. Además, utiliza un protocolo de comunicación estandarizado para que las empresas puedan disminuir la interacción entre los clientes y los servicios, lo cual permite ampliar las aplicaciones con menos presiones e inconvenientes.
- **Mayor confiabilidad:** la SOA genera aplicaciones más confiables, ya que es más fácil depurar servicios pequeños que un código de gran volumen.
- **Gran disponibilidad:** las instalaciones de la SOA están disponibles para todos [46].

Basado en la arquitectura SaaS (*Software as a Service*, en español Software como un Servicio) que se define en el patrón arquitectónico SOA, se decide estructura la propuesta de solución de la siguiente forma: En una computadora servidor estará desplegada la API junto al servidor de bases de datos que almacenará la base de datos del sistema y en otra computadora servidor estará desplegado el cliente web (el portal web) de la propuesta de solución.

2.4.2 Patrones de diseño

Un patrón de diseño se plantea debido a la aparición de un problema recurrente en un entorno web. Una definición más formal señala: “Un patrón de diseño nombra, abstrae e identifica los aspectos claves de una estructura de diseño común que le hacen útil para crear un diseño orientado a objetos reutilizable. El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se centra en un problema usualmente un problema de diseño orientado a objetos. Describe cuando se aplica, si puede ser aplicadas en vista de otras limitaciones de diseño, y las consecuencias y los compromisos de utilizar” [47].

Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)

En diseño orientado a objetos, **GRASP** son *patrones generales de software para asignación de responsabilidades*, es el acrónimo de "*General Responsibility Assignment Software Patterns*". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de *software*.

- **Experto:** es el principio básico de asignación de responsabilidades. Este principio se aplica en el servicio *RequestService* del *Frontend*, donde se manejan todas las peticiones hacia la API.
- **Controlador:** el patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa. Este patrón se aplica en cada componente definido en el *Frontend* en el que contiene una vista *.html* y un controlador *.ts*.
- **Alta Cohesión:** la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Este patrón se aplica en el *Backend*, en la definición de un módulo para gestionar cada entidad de la base de datos.
- **Bajo Acoplamiento:** con el Acoplamiento de Control, un módulo le envía a otro un elemento de control que determina la lógica de ejecución del mismo. Este patrón se emplea en el *Backend* en los módulos que gestionan las relaciones *many-to-many*, en los que se reciben entidades de otros módulos para concretar una relación.

2.4.3 Estándares de codificación

Los estándares de codificación definen la estructura y apariencia física del código, lo que facilita su comprensión, mantenimiento y lectura.

- Los nombres de variables, funciones o parámetros deben seguir la codificación *Camel Case*.

```
public setTitle = (title: string) => {  
  this.title.setTitle(`JAN | ${title}`);  
};
```

- Poner sufijo *Controller* a las clases que cumplan esta funcionalidad.

```
export class AuthController {  
  (. . .)  
}
```

- Poner el sufijo \$ a todas propiedades de clases o variables que almacenen una suscripción.

```
public subscription$: Subject<any> = new Subject<any>();
```

- Especificar los modificadores de acceso *public*, *private* y *protected* en cada caso que sea necesario.

```
public theme: 'light' | 'dark' = 'light';  
public size: 'compact' | 'normal' = 'normal';  
public lang = 'es';  
public type: 'image' | 'audio' = 'image';  
public useGlobalDomain = false;
```

2.5 Diagrama de clases de diseño con estereotipos web

A continuación, se relaciona el diagrama de clases del diseño correspondiente a las historias de usuarios descritas anteriormente. Se muestran las relaciones entre las clases participantes, teniendo como característica particular sobre los diagramas de clases tradicionales, el uso de estereotipos UML especiales que son específicos para los diagramas de clases web, los cuales permiten modelar de forma certera la arquitectura y funcionamiento de aplicaciones de este tipo [48].

2.5.1 HU Crear Publicación

Este diagrama de clases con estereotipos web muestra la interacción de los componentes arquitectónicos del sistema y la relación entre ellos, lo que permiten crear una publicación.

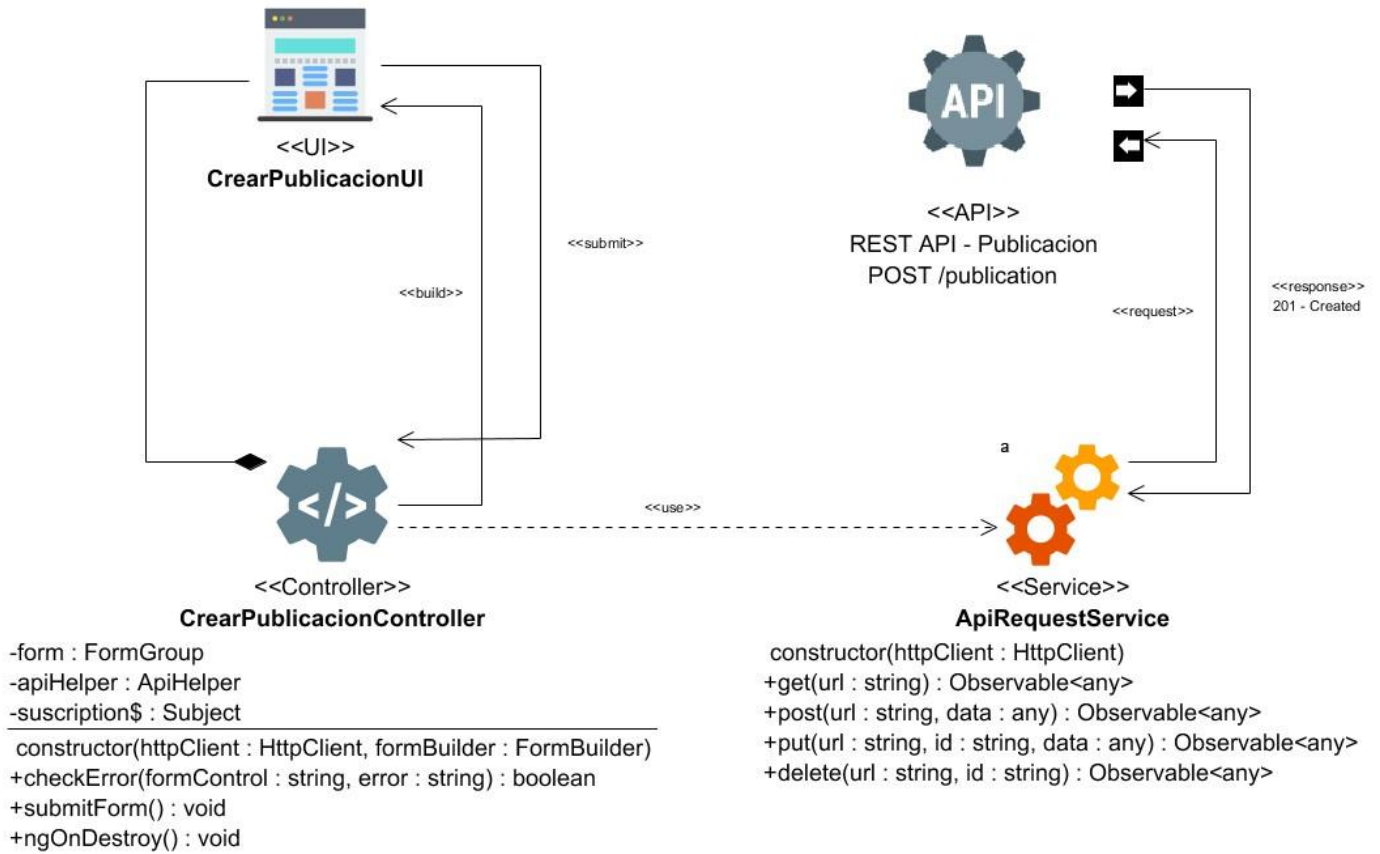


Figura 3: Diagrama de Clases con Estereotipos web de CU Crear Publicación

2.5.2 HU Buscar Institución

Este diagrama de clases con estereotipos web muestra la interacción de los componentes arquitectónicos del sistema y la relación entre ellos, lo que permiten realizar búsquedas en el sistema sobre instituciones que estén registradas.

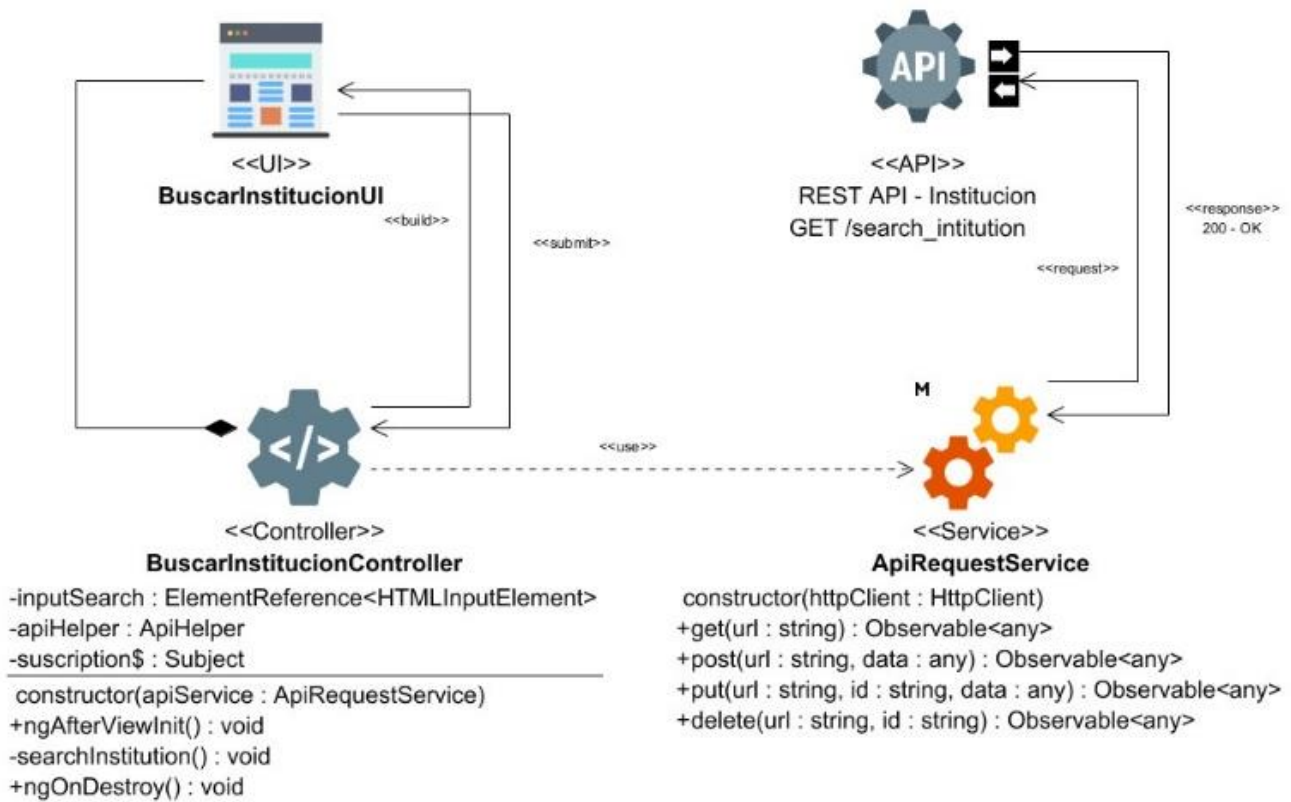


Figura 4: Diagrama de clases con Estereotipos web CU Buscar institución

2.5.3 HU Responder Inquietud

Este diagrama de clases con estereotipos web muestra la interacción de los componentes arquitectónicos del sistema y la relación entre ellos, lo que permiten responder a inquietudes que hayan sido registradas en el sistema por los usuarios.

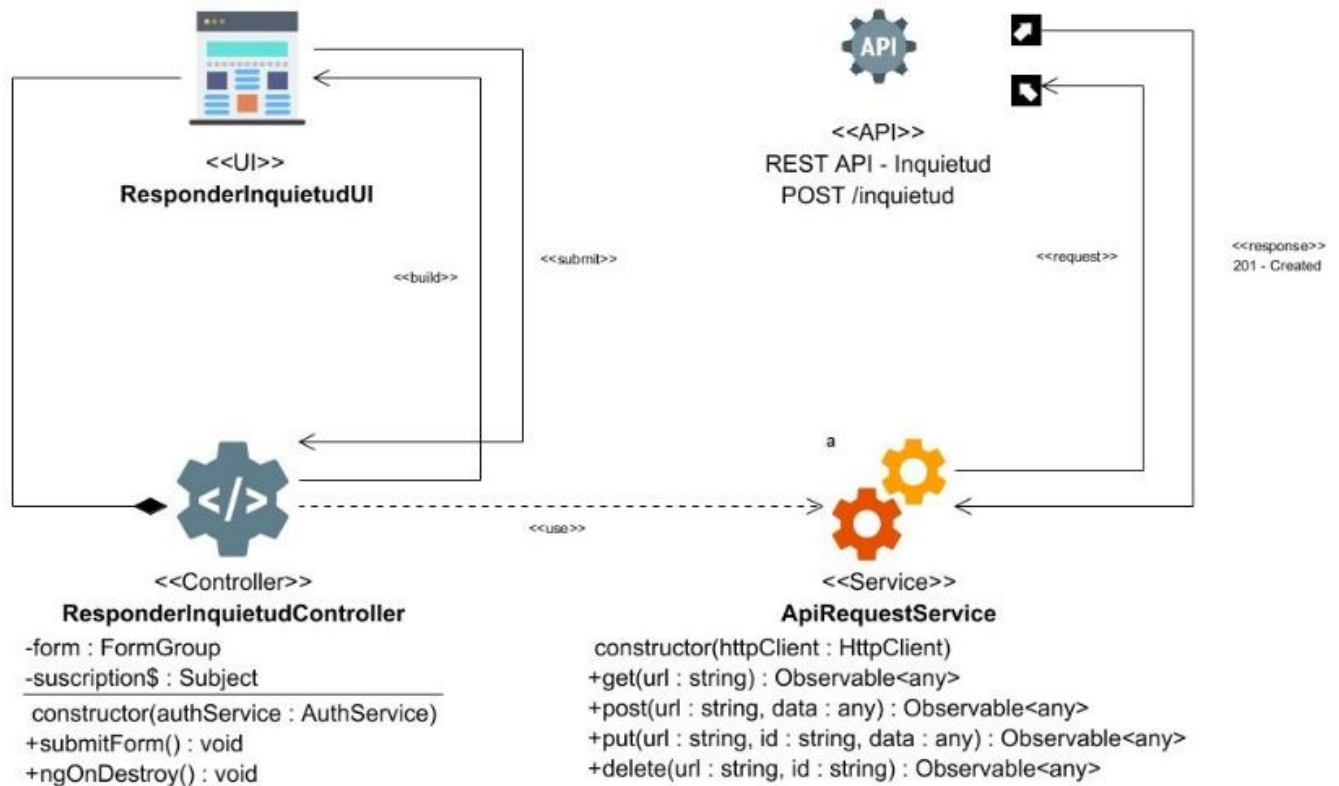


Figura 5: Diagrama de clases con estereotipos web de CU Responder inquietud

2.6 Diagramas de secuencia

Los diagramas de secuencia son una solución de modelado dinámico popular en UML porque se centran específicamente en *líneas de vida* o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine [49].

2.6.1 HU Crear Publicación

Este diagrama de interacción muestra el flujo de información e interacción entre los actores del caso de uso dando como resultado la creación de una publicación en el portal web.

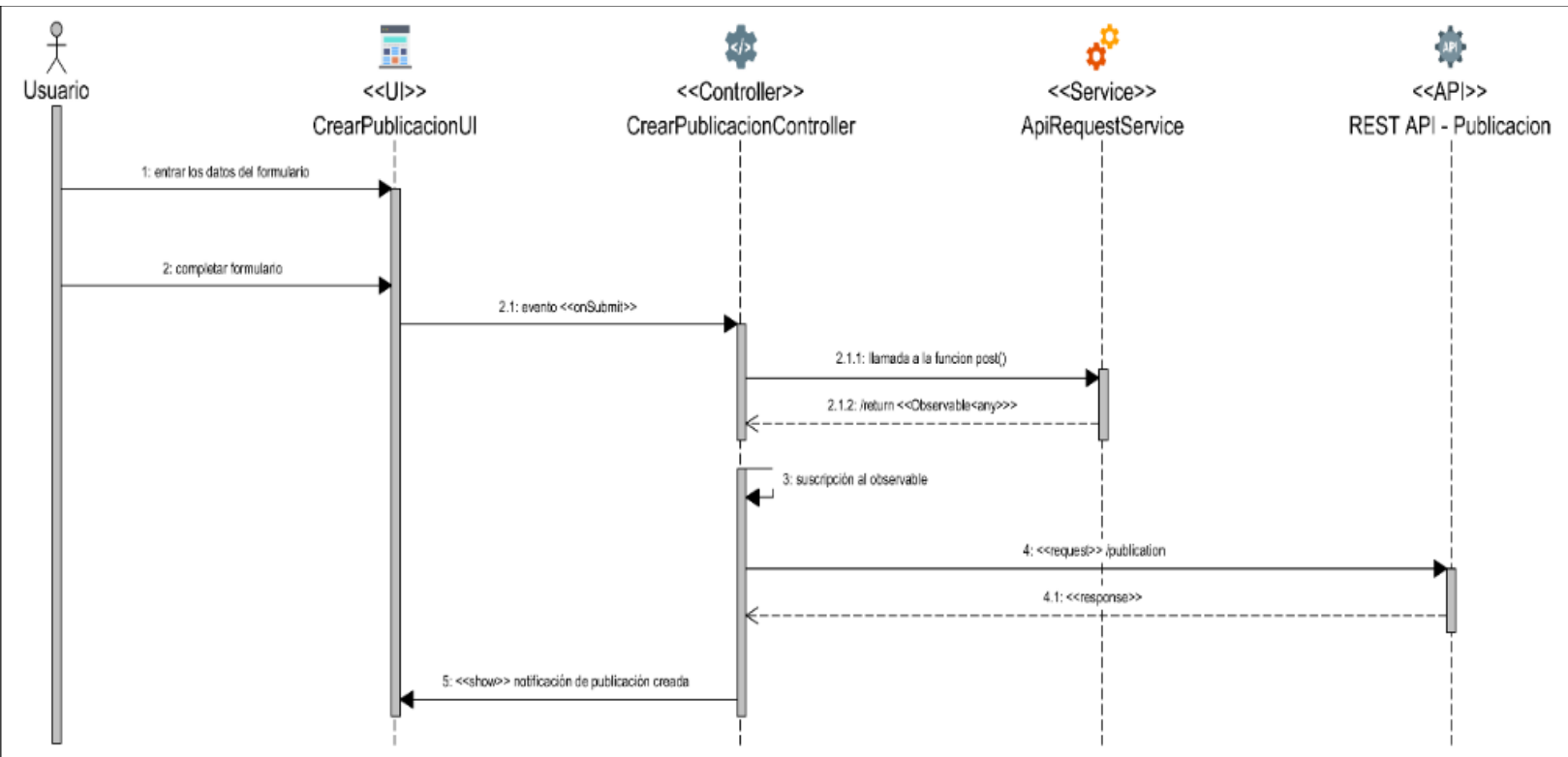


Figura 6: Diagrama de Secuencia de CU Crear publicación

2.6.2 HU Buscar Institución

Este diagrama de interacción muestra el flujo de información e interacción entre los actores del caso de uso dando como resultado la búsqueda de las instituciones registradas en el sistema.

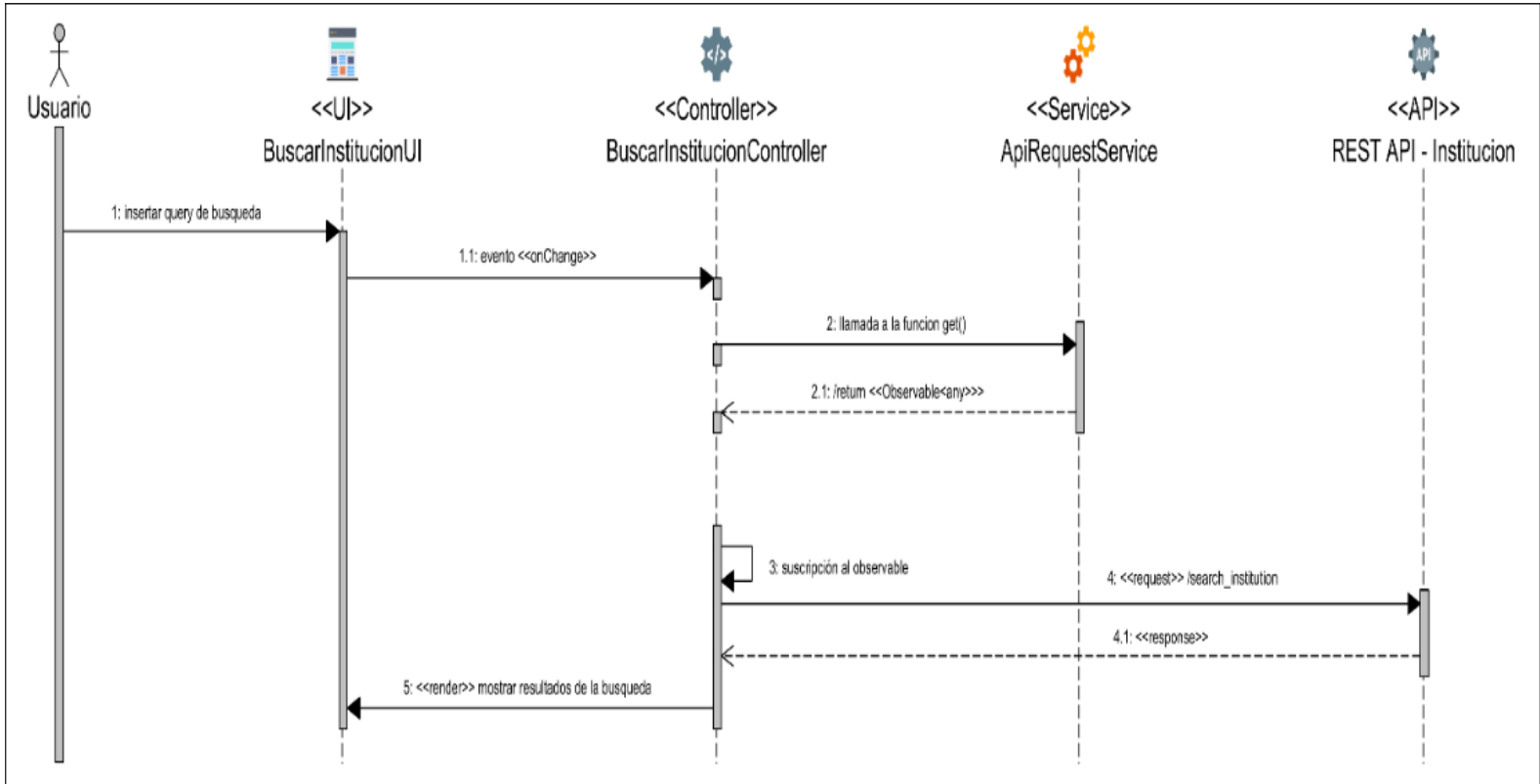


Figura 7: Diagrama de Secuencia de CU Buscar Institución

2.6.3 CU Responder Inquietud

Este diagrama de interacción muestra el flujo de información e interacción entre los actores del caso de uso dando como resultado dar respuesta a inquietudes registradas por los usuarios del portal web.

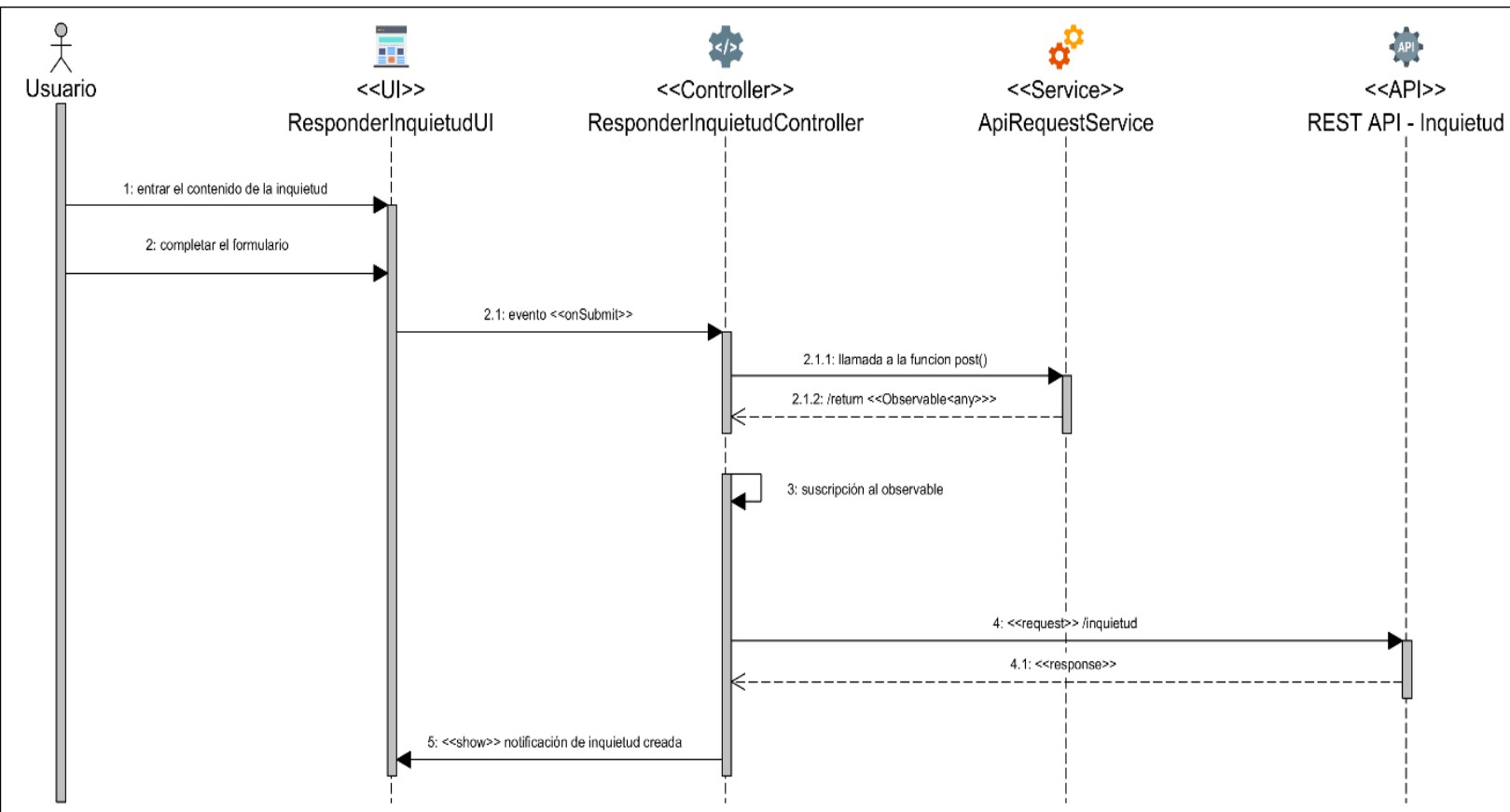


Figura 8: Diagrama de Secuencia de CU Responder Inquietud

2.7 Modelo de datos

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades.

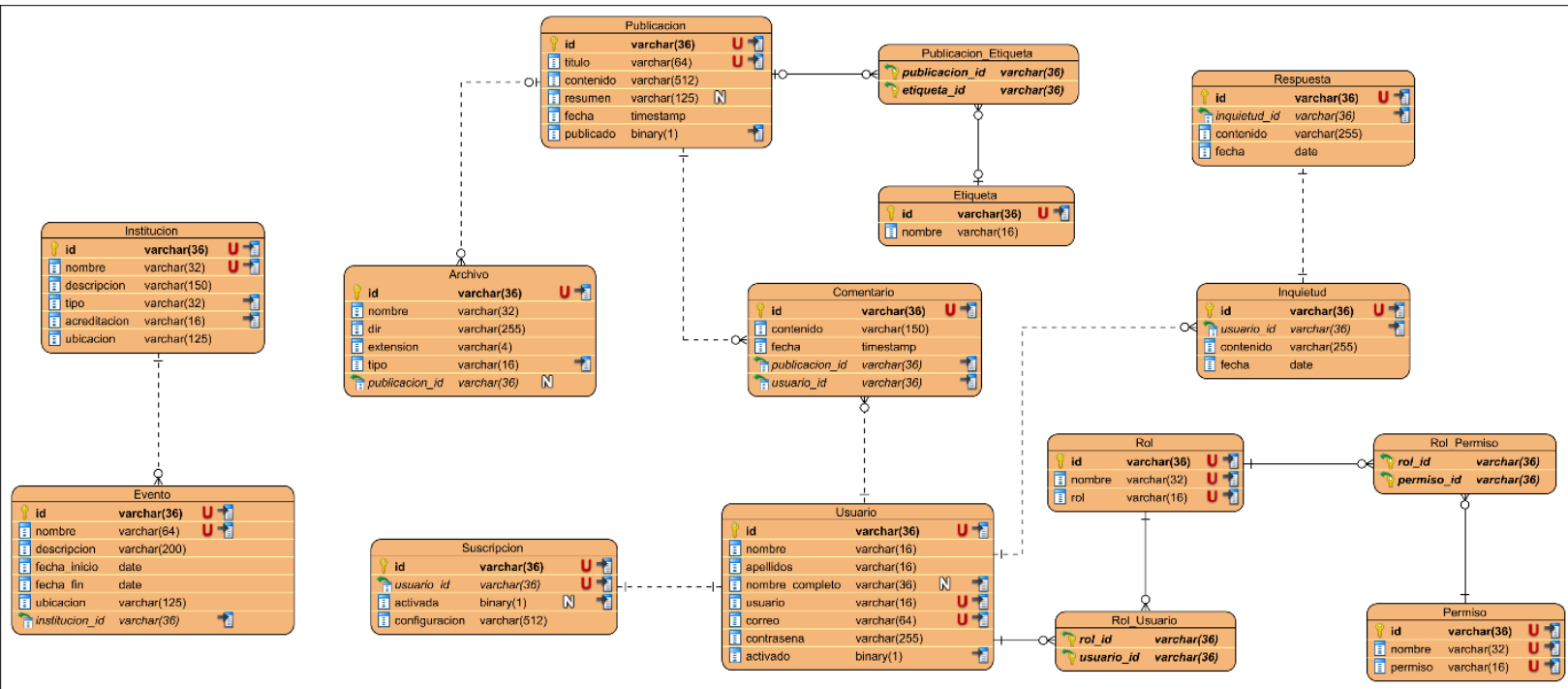


Figura 9: Modelo de Datos

El modelo de datos de la propuesta de solución tiene 15 tablas y 9 relaciones entre ellas. Los usuarios podrán tener 1 o varios roles, tendrán un registro de suscripción en el sistema que podrá estar activo o no, además podrán registrar inquietudes en el portal las cuales se darán respuesta por vía correo electrónico. Las publicaciones que se hagan en el portal podrán tener una o varias etiquetas y una imagen adjunta, además de poder tener comentarios realizado por usuarios. Las instituciones registradas en el sistema podrán estar asociadas a eventos que podrán ser de diferentes índoles, estos eventos los promueve el organismo acreditador, que en este caso es la JAN. Además, las instituciones contarán con un registro de programas y carreras que han sido acreditadas con alguna de las categorías que defina la JAN. Los documentos adjuntos podrán guardarse sin necesidad de estar relacionados con alguna publicación realizada en el portal web.

2.8 Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos.

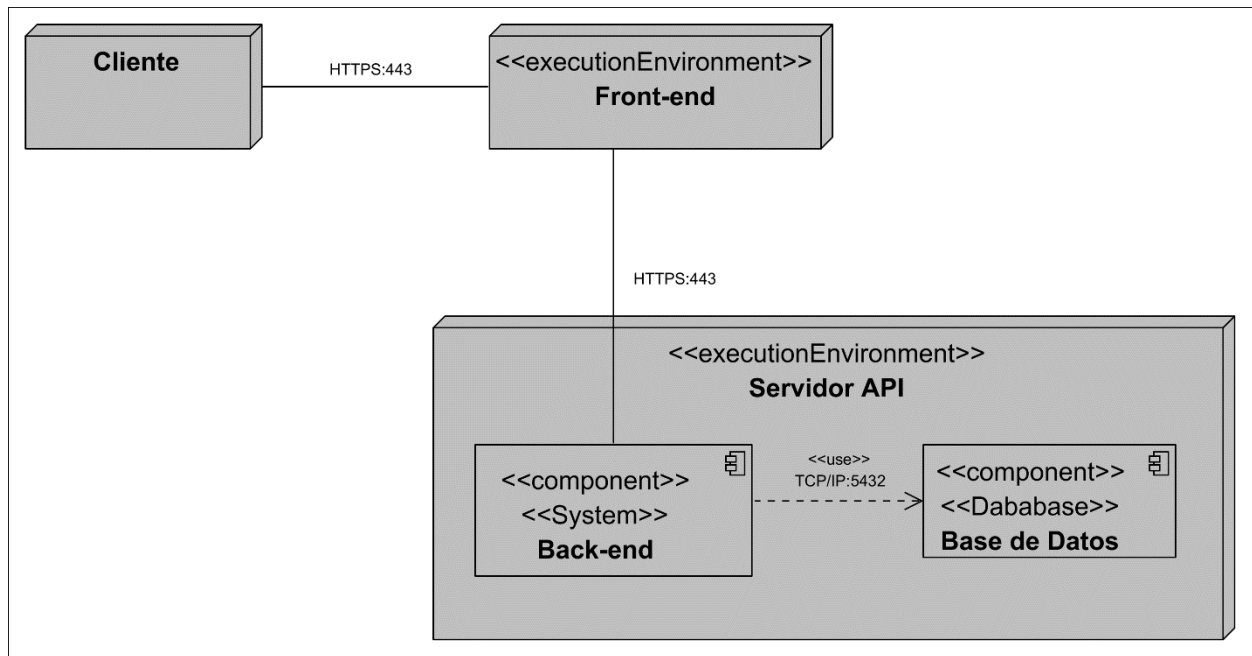


Figura 10: Diagrama de Despliegue

Especificación de despliegue:

- **Sistema Operativo:** Windows 10 Home 21H1 19043.1237
 - **Microprocesador:** Intel(R) Core (TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
 - **Memoria RAM:** 12 GB
 - **Almacenamiento:** 250 GB SSD M.2
- **Servidor Web:** Apache 2.14
- **Servidor de Base de Datos:** PostgreSQL 13
- **Motor:** Node >= 14.18.1
- **Gestor de paquetes:** Yarn >= 1.22.17, NPM >= 8.1.3
- **Interfaz de Comando:** @nestjs/cli >= 8.1.2, @angular/cli ~12.2.9

2.9 Conclusiones del capítulo

Tras el desarrollo del presente capítulo se arriba a las siguientes conclusiones:

- La obtención de los requisitos funcionales y no funcionales permitió identificar las funcionalidades que el sistema presentará, así como las restricciones en las que operará el sistema.
- La elaboración de todos los artefactos ingenieriles permitirá disminuir el impacto de las nuevas funcionalidades o cambios en el código fuente del sistema, en correspondencia de la especificación de los requisitos funcionales y no funcionales.
- La definición del modelo de datos provee de un resumen de la estructura de datos general de la aplicación, así como la interrelación de las clases que lo componen.

Capítulo 3: Implementación y Pruebas del Sistema

En el desarrollo de este capítulo se cuantifica el nivel de fiabilidad de los resultados obtenidos y la calidad en el desarrollo de la propuesta de solución. Para lograr esta valoración se definen estrategias de pruebas de software en consonancia con los estándares actuales, donde se decide aplicar distintas disciplinas de pruebas internas y de funcionalidad que se definen en la metodología que se empleó como guía durante todo el desarrollo de la propuesta de solución, con la única finalidad de verificar la calidad del producto final antes de la entrega al cliente.

3.1 Pruebas de software para la evaluación de la propuesta de solución

Las pruebas funcionales del sistema permiten verificar que el sistema en desarrollo satisface sus requisitos funcionales. Existen una amplia cantidad de trabajos y capítulos de libros que proponen cómo obtener objetivos de prueba a partir de requisitos funcionales expresados como casos de uso [51].

En los últimos años se ha propuesto una serie de principios que permiten establecer unas pautas comunes para que las empresas de desarrollo de software las conozcan y adapten a sus procesos de pruebas. Las pruebas son unas herramientas que permiten identificar la presencia de defectos; sin embargo, no garantizan que no haya defectos ocultos en el software, y el hecho de que no se identifiquen defectos no es una evidencia de que el software esté totalmente correcto [52].

Las pruebas de software permiten la ejecución de un sistema con el objetivo principal de encontrar errores presentes en el mismo; así conseguimos reducirlos y mejorar la calidad del sistema, haciéndolo más robusto frente a fallos.

3.2 Estrategia de pruebas

Una estrategia de pruebas, no es una serie de instrucciones que detallan el paso a paso de cómo se debe hacer las pruebas. La estrategia define los objetivos a alcanzar, a menudo con recursos limitados y en condiciones de incertidumbre, donde se construye un plan de alto nivel para lograr los objetivos específicos de la prueba. La estrategia que se adopte dependerá de principalmente del proceso de desarrollo que se utiliza al interior de una organización TI y de las capacidades existentes del equipo de prueba [53].

Prueba del software a implementar:

- **Pruebas funcionales:** es una prueba de tipo caja negra basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software.
- **Pruebas de Usabilidad:** Este tipo de prueba se refiere a asegurar de que la interfaz de usuario (o GUI) sea intuitiva, amigable y funcione correctamente.
- **Pruebas de rendimiento:** Esta prueba es llevada a cabo por un grupo de pruebas independientes para asegurar que el rendimiento del sistema está dentro de los parámetros definidos. Se puede requerir herramientas para simular clientes y cargas pesadas, y se realizan las mediciones precisas de desempeño.
- **Pruebas de seguridad:** es un conjunto de pruebas objetivas con el fin de detectar las vulnerabilidades de un sistema, teniendo muy claro que ningún sistema es 100% seguro o inviolable.
- **Pruebas de aceptación:** pertenecen a las últimas etapas previas a la liberación en firme de versiones nuevas a fin de determinar si cumplen con las necesidades y/o requerimientos de las empresas y sus usuarios.

3.2.1 Pruebas funcionales

“Se basan en funciones, prestaciones y en su interoperabilidad con sistemas específicos, y pueden llevarse a cabo en todos los niveles de prueba”. Es importante mencionar que se orientan en el comportamiento externo de un producto o aplicativo software, en las pruebas de caja negra.

La técnica de **Caja Negra**: “Puede utilizarse para lograr objetivos de cobertura de entrada y salida, con entradas humanas, vía interfaces a un sistema, o parámetros de interfaz de las pruebas de integración” [52].

Se emplearán las pruebas de caja negra con el objetivo de encontrar errores en las siguientes categorías:

- 1) Funciones incorrectas o faltantes.
- 2) Errores de interfaz.
- 3) Errores en las estructuras de datos o en el acceso a bases de datos externas.
- 4) Errores de comportamiento o rendimiento.

Para desarrollar el método de caja negra se utilizan las técnicas [54]:

- Partición de equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Análisis de valor de frontera: prueba la habilidad del programa para manejar datos que se encuentran en los límites o fronteras aceptables.

Para poner en práctica este método de pruebas, según las técnicas que se describieron anteriormente, se toma en cuenta los Diseños de Casos de Pruebas (DCP). El diseño de casos de prueba para la **partición de equivalencia** se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Por su parte, el **análisis de valor de frontera** es una técnica de diseño de casos de prueba que complementan la partición de equivalencia. En lugar de seleccionar algún elemento de una clase de equivalencia, éste conduce a la selección de casos de prueba en los “bordes” de la clase. En lugar de enfocarse exclusivamente en las condiciones de entrada, también deriva casos de prueba a partir del dominio de salida [41].

Para llevar a cabo las pruebas de caja negra se confeccionarán CP para los requisitos de mayor impacto en la funcionalidad del sitio web. Esto garantizará un mayor nivel de fiabilidad en el producto final.

3.2.2 Pruebas de usabilidad

Las pruebas de usabilidad son procedimientos con los que se puede probar la usabilidad de una web. Utilizando métodos empíricos, una prueba de usabilidad ofrece posibilidades para optimizar la experiencia del usuario. El resultado de tal prueba puede llevar a cambios en el diseño o en la redacción de la web.

Con las pruebas de usabilidad, se desarrollan métodos que proporcionan una situación fáctica clara. La implementación de las pruebas tiene la ventaja de que pueden implementarse online. Esto beneficia al cliente en términos de coste, especialmente en comparación con los métodos más tradicionales, ya que las pruebas se pueden realizar en gran medida en sus propias webs.

Las pruebas de usabilidad se pueden realizar para muchas áreas y métodos diferentes. Los posibles objetivos de las pruebas son:

- Comprobación de la usabilidad diaria del software en la fase de desarrollo.
- Análisis de escenarios definidos en el marco de una compra.

- Análisis de una web para explicar fenómenos como un alta tasa de rebote o un bajo tiempo de retención.

Se utilizan los siguientes métodos:

- Encuestas in situ
- Pruebas A/B
- *Eye Tracking*
- Pruebas *think aloud*
- Prueba de daltonismo
- Evaluación heurística
- Escala de Usabilidad del Sistema

Para llevar a cabo las pruebas de usabilidad a la propuesta de solución se empleará la Evaluación Heurística: Consiste en valorar si los elementos de la interfaz del usuario de un sitio web o aplicación se corresponden con los principios de usabilidad establecidos a partir de una lista de principios o heurísticas. Esta técnica consigue detectar un gran número de problemas de usabilidad en sitios web con un menor coste que las técnicas que involucran usuarios. Para la evaluación general de sitios web existen criterios muy establecidos y probados que proporcionan la seguridad de que se está evaluando todo lo que es necesario evaluar [55]:

- Tipografía
- Navegabilidad
- Diseño, Estructura y Elementos de la Página
- Imágenes y contenido multimedia
- Contenido escrito para la web

Para la aplicación de esta técnica, se utilizará la lista de chequeo establecida por el Departamento de Calidad de la UCI para aplicaciones web, en ella se describen cuáles son los indicadores a evaluar por cada una de las heurísticas descritas anteriormente.

3.2.3 Pruebas de rendimiento

El objetivo de las pruebas de rendimiento es determinar el rendimiento del sistema bajo una carga de trabajo definida utilizando diferentes tipos de pruebas de rendimiento tales como pruebas de carga, de estrés y de estabilidad.

- **Pruebas de rendimiento baja intensidad:** Durante la fase de evaluación, cuando ejecuta pruebas funcionales a pequeña escala (pruebas de integración de unidad o componente), es útil supervisar los tiempos de respuesta de transacciones individuales.
- **Pruebas de rendimiento de alta intensidad:** Las pruebas de rendimiento de alta intensidad se pueden considerar como pruebas de rendimiento instrumentadas. Esta fase de las pruebas se realiza en escenarios completos, tales como pruebas de estrés y pruebas de estabilidad realizadas en procesos de negocio globales.
- **Visualizar restricciones del rendimiento:** Si el entorno de prueba consta de interacciones complejas entre componentes basadas en mensajes, las pruebas de rendimiento comprobarán que estos mensajes fluyen por el sistema a la velocidad necesaria. Para obtener una mejor comprensión de este concepto, es importante visualizar cómo los componentes procesan estos mensajes [56].

Durante las pruebas de rendimiento se empleará como herramienta auxiliar para el proceso de pruebas de carga y estrés la aplicación JMeter para facilitar el procedimiento de validación de la aplicación. El resultado de estas pruebas determinase la capacidad real de la aplicación para enfrentar en entorno de explotación.

3.2.4 Pruebas de seguridad

Existen diferentes tipos de Pruebas de Seguridad, las más comunes y aceptadas son las Pruebas de Penetración de Caja Negra, las Pruebas de Penetración de Caja Blanca y las Pruebas de Penetración de Caja Gris.

- **Prueba de Caja Negra:** En este tipo de auditoría el equipo de consultores no recibe ningún tipo de información sobre los sistemas informáticos y activos pertenecientes a la infraestructura de TI de la organización. En este caso, el equipo de consultores sólo recibe el nombre de la institución, por lo que se trabaja con la información que se puede recolectar a través de medios públicos. Este tipo de pruebas simula el ataque de un cracker, por lo que permite medir el alcance e impacto que tendría un evento real.
- **Prueba de Caja Blanca:** Este enfoque de auditoría se utiliza cuando el cliente necesita realizar un análisis de seguridad a profundidad en los sistemas informáticos. En este tipo de auditoría, el equipo consultor recibe información con mayor detalle sobre los activos y servicios de la infraestructura tecnológica, tal como: versiones de los servicios

que se ejecutan, listas de los sistemas operativos instalados en los servidores, código fuente de aplicaciones, entre otros.

- **Prueba de Caja Gris:** Este tipo de auditoría es una combinación de los tipos anteriores, en donde el cliente entrega cierta información, pero no toda al equipo de consultores, tal como: segmentos de red, direcciones IP de servidores pertenecientes a la infraestructura de TI, diagramas con la topología de la organización, entre otros. Otra manera de clasificar el tipo de pruebas es respecto al lugar desde el que el equipo de consultores realiza la ejecución de las mismas [57].

Con la implementación de las pruebas de seguridad se logra una perspectiva detallada de las vulnerabilidades encontradas en los sistemas, lo cual es de gran ayuda al momento de aplicar medidas correctivas para elevar la confiabilidad de la aplicación.

3.2.5 Pruebas de aceptación

Prueba de aceptación del usuario es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Las pruebas de aceptación son aquellas que son diseñadas por el propio equipo de desarrollo en base a los requisitos funcionales especificados, y ejecutadas por el propio usuario de manera que este dé validez y conformidad al producto que se les está entregado en base a lo que se acordó inicialmente.

De forma general las pruebas de aceptación pueden afrontarse mediante los siguientes tipos [58]:

- Pruebas de Aceptación del Usuario (UAT)
- Pruebas de Aceptación Operativa (OAT)
- Pruebas de Aceptación de la Normativa
- Pruebas de Aceptación Alfa y Beta
- Pruebas de Aceptación Contractual

Para el caso de la propuesta de aceptación se someterá su funcionamiento a un proceso de pruebas de aceptación Alfa y Beta. Durante el estado de alfa se dará a probar a un grupo reducido de personas con el objetivo de tener el control de los entonos en los cuales se pruebe el sitio web; una vez resueltos los posibles errores que se encuentren durante esta etapa se

pasará a un estado de prueba beta en la que se dará acceso a probar el sitio a un grupo mayor de personas. La ejecución de este tipo de prueba validará que la aplicación esta lista para ser desplegada en un entorno de producción.

3.3 Conclusiones del capítulo

Tras el desarrollo del presente capitulo se arriba a las siguientes conclusiones:

- La definición de una estrategia de pruebas será fundamental para validar la calidad de la aplicación.
- La ejecución de varios tipos de pruebas a la propuesta de solución posibilita validar la calidad desde varias artistas.

Conclusiones

Tras el desarrollo del presente documento se arriba a las siguientes conclusiones:

- La definición de un marco teórico referencial de la investigación asociada a instituciones acreditadoras de la enseñanza superior, dio lugar a la comprensión total del presente trabajo de diploma, así como sentar las bases teórico-metodológicas y conceptuales del mismo.
- La definición de la metodología de desarrollo de software, definió las pautas para el proceso ordenado y detallado de la propuesta de solución.
- La definición de las tecnología y herramientas que formaron parte del entorno de desarrollo, están acorde con los requerimientos actuales del mercado y apuntan a la soberanía tecnológica del país.
- Con la elaboración de todos los artefactos ingenieriles que define la metodología seleccionada para el desarrollo, y tomando como referencia la arquitectura de los marcos de trabajo seleccionados, se obtuvo una representación abstracta de la propuesta de solución.
- La definición de la estrategia de pruebas es de vital importancia en la validación de la calidad de la aplicación.

Referencias Bibliográficas

1. Moreno Campdesuñer, Ileana. LA CALIDAD EDUCATIVA EN LA ENSEÑANZA UNIVERSITARIA. Villa Clara: Universidad Central "Marta Abreu" de Las Villas, 2009. ISBN: 978-959-250-359-5.
2. Rodríguez Espinar, Sebastián. LA EVALUACIÓN DE LA CALIDAD EN LA EDUCACIÓN SUPERIOR. ISBN: 978-84-995880-3-2.
3. Pérez Porto, Julián; Merino, María. Definición de calidad educativa. Definición.de. 2021. [consulta: 20-06-2021]. Disponible en: <https://definicion.de/calidad-educativa/>.
4. Torres Hernández, Alfonso. Evaluación y calidad de la educación. Milenio: Grupo Milenio. 22 de junio de 2016. [consulta: 19-06-2021]. Disponible en: <https://www.milenio.com/opinion/alfonso-torres-hernandez/apuntes-pedagogicos/evaluacion-y-calidad-de-la-educacion>.
5. Carrión, Juan. Diferencia entre dato información y conocimiento. 2017. [consulta: 19-06-2021]. Disponible en: http://www.gestiondelconocimiento.com/conceptos_diferenciaentredato.htm
6. Sánchez, J.; Milena, A.; Rodríguez, L. Gestión del conocimiento. TIA, Tecnología, Investigación y Académica, 2018, p. 46-51. ISSN: 2344-8288.
7. Real Academia Española, 2021. [consulta: 18-06-2021]. Disponible en: <https://dle.rae.es/conocimiento>.
8. Educación, Argentina. Concepto.de: Etecé Editorial. [consulta: 21-06-2021]. Disponible en: <https://concepto.de/educacion-4>.
9. Pérez Porto, Julián; Gardey, Ana. Definición de Educación, 2021. Definicion.de. [consulta: 21-06-2021]. Disponible en: <https://definicion.de/educacion/>.
10. Educación, 2019. Significados.com. [consulta: 21-06-2021]. Disponible en: <https://www.significados.com/educacion/>
11. La calidad de la educación: principales conceptos, 2017. ISOTools.org. [consulta: 21-06-2021]. Disponible en: <https://www.isotools.org/2017/03/25/la-calidad-en-la-educacion-principales-conceptos/>.
12. Reyes Jaimes, Gabriel Renato. La Calidad Educativa: Un concepto Construido, 2017. Revista: Acta Educativa (06). 01 – 10.2018.
13. Evaluación, 2018. Significados.com. [consulta: 2021-06-21]. Disponible en: <https://www.significados.com/evaluacion/>.

14. COPAES – Glosario, 2021. [consulta: 21-06-2021]. Disponible en: <https://www.copaes.org/glosario.html>.
15. Real Academia Española, 2021. [consulta: 01-09-2021]. Disponible en: <https://dle.rae.es/enseñanza>.
16. Real Academia Española, 2021. [consulta: 01-09-2021]. Disponible en: <https://dpej.rae.es/lema/educación-superior>.
17. ABET | ABET Accreditation. [consulta: 24-06-2021]. Disponible en: <https://www.abet.org/>.
18. COPAES. Consejo para la Acreditación de la Educación Superior, A.C. [consulta: 24-06-2021]. Disponible en: <https://www.copaes.org/>.
19. AQU: AQU site principal. *Agència per a la Qualitat del Sistema Universitari de Catalunya* [consulta: 25-06-2021]. Disponible en: <https://www.aqu.cat/es/>.
20. aaq – swiss agency of accreditation and quality assurance. [consulta: 25-06-2021]. Disponible en: <https://aaq.ch/en/>.
21. About Us. *The Quality Assurance Agency for Higher Education*. [consulta: 25-06-2021]. Disponible en: <https://www.qaa.ac.uk/about-us>.
22. Quality Code. *The Quality Assurance Agency for Higher Education*. [consulta: 25-06-2021]. Disponible en: <https://www.qaa.ac.uk/quality-code>.
23. Zuñiga Vazquez, Luis Ramon. Desarrollo de aplicaciones web utilizando Angular como framework, 2020. Universidad Politécnica de Sinaloa. Mazatlán, Sinaloa.
24. Acosta, Víctor Manuel. *Frontend con Angular: Todo lo que debes saber sobre esta herramienta*, 2019. Revista Digital INESEM. [consulta: 31-08-2021]. Disponible en: <https://revistadigital.inesem.es/informatica-y-tics/frontend-con-angular-todo-lo-que-debes-saber-sobre-esta-herramienta/>.
25. NESTJS. Documentation | NestJS - A progressive Node.js framework. *Documentation | NestJS - A progressive Node.js framework* [online]. [consulta: 5-10-2021]. Disponible en: <https://docs.nestjs.com>.
26. PHAM, Anh Duc. *Developing Backend of a web application with NestJS framework: Case: Integrify Oy's student management system*. [online]. 2020. [consulta: 5-10-2021]. Disponible en: <http://www.theseus.fi/handle/10024/353200>.
27. Buenaventura Correa, Carlos José. *Dx Design: herramienta gráfica para la creación de prototipos web*, 2019. Bogotá, Colombia. Universidad de los Andes, Facultad de Arquitectura y Diseño, Departamento de Diseño.

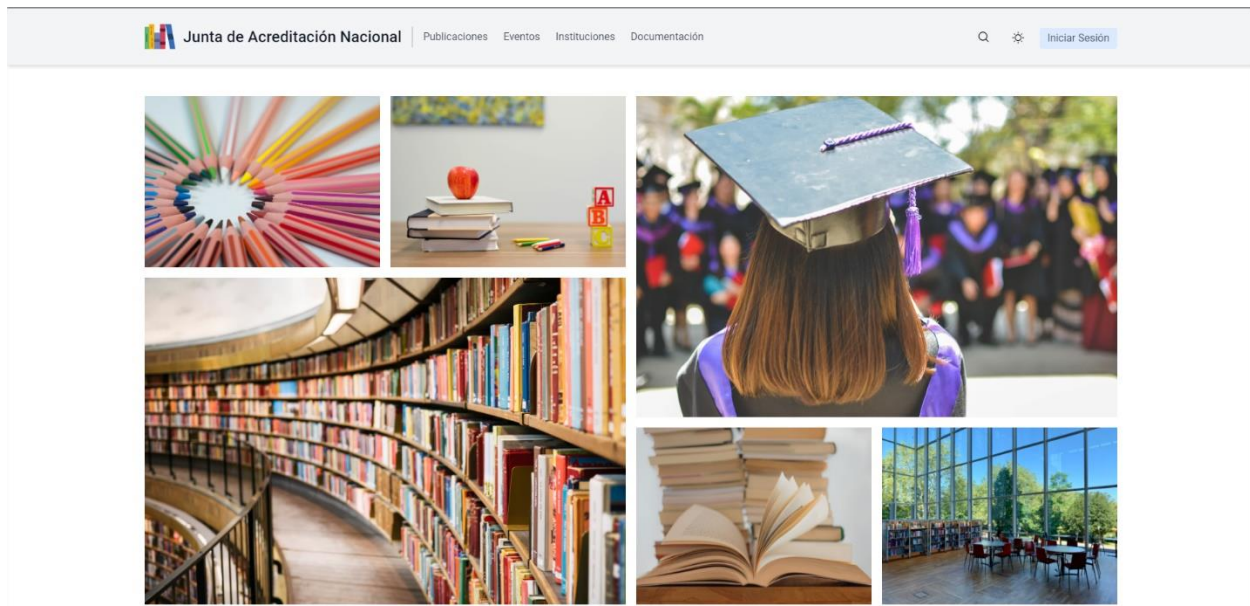
28. Beati, Hernán. HTML5 y CSS3 Para diseñadores, 2016. Alfaomega Grupo Editon, Argentina. ISBN: 978-987-1609-67-3.
29. Hernández Sanz, Javier. Desarrollo completo de un sitio web con buenas prácticas, 2018. *Universitat Politècnica de València*. Departamento de Sistemas Informáticos y Computación.
30. Pérez Jiménez, Juan Diego. Qué es Sass: ventajas, desventajas y ejemplos de desarrollo, 2019. OpenWebinars S.L. [consulta: 31-08-2021]. Disponible en: <https://openwebinars.net/blog/que-es-sass-ventajas-desventajas-y-ejemplos-de-desarrollo/>.
31. Cabrera Barrios, José. Qué es TypeScript, 2020. OpenWebinars S.L. [consulta: 31-08-2021]. Disponible en: <https://openwebinars.net/blog/que-es-typescript/>.
32. ¿Qué es el lenguaje unificado de modelado (UML)? Lucidchart. [consulta: 23-06-2021]. Disponible en: <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>.
33. ¿Qué es una API de REST? RedHat, Inc. [online]. [consulta: 6 -10-2021]. Disponible en: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>.
34. WebStorm: *The Smartest JavaScript IDE by JetBrains*. JetBrains s.r.o. [consulta: 23-06-2021]. Disponible en: <https://www.jetbrains.com/webstorm/>.
35. Flores, E. Endurecimiento de navegadores web más utilizados. Caso práctico: Implementación de navegadores endurecidos en un paquete integrado para Microsoft Windows. (Tesis de maestría no publicada). 2015. Universidad Don Bosco, San Salvador, El Salvador, C.A.
36. API Platform: API Tools & Solutions | Postman. Postman, Inc. [consulta: 23-06-2021]. Disponible en: <https://www.postman.com/api-platform/>.
37. Pérez Ibarra, Susana Graciela; Quispe, José Rolando; Mullicundo, Felipe Fernando; Lamas, Daniel Alberto. Herramientas y tecnologías para el desarrollo web desde el *FrontEnd* al *BackEnd*. En XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja). Facultad de Ingeniería - Universidad Nacional de Jujuy. ISBN: 978-987-24611-3-3.
38. GROUP, PostgreSQL Global Development. PostgreSQL. PostgreSQL [online]. [consulta: 6-10-2021]. Disponible en: <https://www.postgresql.org/>. The world's most advanced open-source database.

39. Navicat Premium | Conecta múltiples bases de datos en una única interfaz gráfica de usuario. PremiumSoft™ CyberTech Ltd. Disponible en: <https://www.navicat.com/es/products/navicat-premium>. [consulta: 6-10- 2021].
40. About Visual Paradigm. Visual Paradigm. [consulta: 23-06-2021]. Disponible en: <https://www.visual-paradigm.com/aboutus/>.
41. Rodríguez Sánchez, Tamara. Metodología de desarrollo para la Actividad productiva de la UCI. Serie Científica de la Universidad de las Ciencias Informáticas. Vol. 10, No. 8, Mes Agosto, 2017. ISSN: 2306-2495 | RNPS: 2343. Pág. 16-31. Disponible en: <http://mejoras.prod.uci.cu/>.
42. Jacobson, Ivar; Booch, Grady; Rumbaugh, James. El proceso unificado de desarrollo de software. La Habana: Félix Varela, 2004.
43. Menzinsky, Alexander. Historias de Usuario. Ingeniería de requisitos ágil, 2018.
44. Valero Orea, Sergio. Estimación de Proyectos de Software con Puntos de Casos de Uso. 2010, vol. 2.
45. Rodríguez Peña, Alina Dolores; Silva Rojas, Luis Guillermo. Arquitectura de software para el sistema de visualización médica Vismedic. Revista Cubana de Informática Médica. June 2016. Vol. 8, no. 1, p. 75–86.
46. ¿Qué es la arquitectura orientada a los servicios? RedHat, Inc. [consulta: 4-10-2021]. Disponible en: <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture>.
47. Castañeda Rojas, Edson Bryan. Propuesta de patrón de diseño de software orientado a prevenir la extracción automatizada de contenido web. Pontificia Universidad Católica del Perú [online]. 26 November 2016. [consulta: 24-09-2021]. Disponible en: <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/7513>.
48. Alemán Jiménez, Yulio; Sosa, Yoniel. Módulo de configuración para el mecanismo de rastreo del buscador Orión. 2014.
49. Lucid Software inc. Tutorial de diagrama de secuencia UML. Lucidchart. [consulta: 5-10-2021]. Disponible en: <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>.
50. Cordero Escobar, Idoris. La acreditación de los Programas por la Junta de Acreditación Nacional. Revista Cubana de Anestesiología y Reanimación [online]. August 2019. Vol. 18, no. 2. [consulta: 10-10-2021]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1726-67182019000200001&lng=es&nrm=iso&tlng=en.

51. Figueroa, Mauricio, Diez, Andrés E., CASTRILLÓN, Luis E. and MANRIQUE, Edison. Simulación, Implementación y Pruebas de un Sistema de Compensación en Corriente Directa basado en Ultra Capacitores, para Sistemas Ferroviarios. Información tecnológica. 2017. Vol. 28, no. 1, p. 179–188.
52. Paz, Mera; Andrés, Julián. Análisis del proceso de pruebas de calidad de software. [online]. 19 de octubre, 2016. [consulta: 13-11- 2021]. Disponible en: <https://repository.ucc.edu.co/handle/20.500.12494/962>
53. Copeland, Lee. *What Are the Key Components of an Effective Test Strategy?* [en línea], [Consulta: 15-11-2021] Disponible en: <https://www.stickyminds.com>.
54. Pressman, Roger S. 2010. *Ingeniería del software, UN ENFOQUE PRÁCTICO*. Séptima. México : The McGraw-Hill Companies, Inc., 2010. ISBN: 978-607-15-0314-5.
55. WIKI, Ryte. ¿Qué son las Pruebas de Usabilidad? - Ryte Wiki. [online]. [consulta: 13-11- 2021]. Disponible en: https://es.ryte.com/wiki/Prueba_de_Usabilidad
56. PRESSMAN, Roger. 2009. *Ingeniería del Software. Un enfoque práctico*. s.l.: McGraw-Hill/Interamericana, 2009.
57. Ortiz Castillo, Ana Milena. Introducción a las pruebas de penetración. Especialización en Seguridad Informática. Universidad Piloto de Colombia – Bogotá, Colombia.
58. IBM Docs. IMB. [online]. 6 March 2021. [consulta: 16-11- 2021]. Disponible en: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/rtw/9.0.0?topic=phases-performance-testing>.

Anexos

Anexo A. Página de Inicio del Portal de la Junta de Acreditación Nacional



Anexo B. Página de una publicación en el Portal de la Junta de Acreditación Nacional



Anexo C. Página de una Institución en el Portal de la Junta de Acreditación Nacional



The screenshot shows the website for the Universidad de las Ciencias Informáticas (UCI) on the National Accreditation Board portal. The header includes the logo of the Junta de Acreditación Nacional and navigation links for Publicaciones, Eventos, Instituciones, and Documentación. A search bar and a 'Iniciar Sesión' button are also visible. The main content area features a large image of the UCI building with the Cuban flag and the UCI logo. Below the image, the university's name and accreditation level are displayed, along with a 'Visitar' button. The text provides details about the university's location, founding year, and accreditation status.

Junta de Acreditación Nacional | Publicaciones | Eventos | Instituciones | Documentación

Q ✨ Iniciar Sesión

Universidad de las Ciencias Informáticas Nivel de Acreditación: Certificada [Visitar](#)

Dirección: Carretera a San Antonio Km 2 ½, Torrent, La Lisa, La Habana, Cuba

La Universidad de las Ciencias Informáticas (UCI), fundada en el año 2002 al calor de la Batalla de Ideas, es una institución adscrita al Ministerio de Educación Superior de la República de Cuba. Es un centro docente-productor que forma profesionales altamente comprometidos y calificados en la rama de la Informática, mediante un modelo de formación atípico y flexible basado en la vinculación estudio – trabajo. La actividad de desarrollo de software está evaluada con la categoría CMMI Nivel 2.

La Universidad de las Ciencias Informáticas es hoy, junto a la Universidad de Artemisa, el Centro de Estudios Superiores más joven del Cuba. La concepción y puesta en marcha de esta institución se desarrolló de forma distinta a como habían sido concebidas las otras 63 universidades cubanas y esto le confiere especiales características que la distinguen en el concierto de la Educación Superior cubana.

La UCI cuenta con un plan de estudio diferente al del resto de las universidades del país, siguiendo el principio martiano de la vinculación Estudio-Trabajo como modelo de formación, se hace énfasis en la producción (desarrollo de software) como parte del proceso de aprendizaje. A partir del quinto semestre de la carrera, los estudiantes pueden vincularse a diferentes proyectos productivos que aportan a la economía del país.