



PLATAFORMA PARA LA FORMACIÓN DE HABILIDADES INVESTIGATIVAS EN LOS ESTUDIANTES UNIVERSITARIOS

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autora: Liz Carmen Leal Montejo

Tutores: MSc. Miguel Ángel Hernández de la Rosa
Ing. Yojahny Chávez Marrero

La Habana, septiembre de 2020
"Año 62 de la Revolución"

Agradecimientos

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo **Liz Carmen Leal Montejo**, con carné de identidad **97012109051** soy la autora principal de la presente tesis que tiene por título **“Plataforma para la Formación de Habilidades Investigativas en los Estudiantes Universitarios”** y por este medio autorizo a la Universidad de Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales de carácter exclusivo.

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

Liz Carmen Leal Montejo

Autora

MSc. Miguel A. Hernández de la Rosa

Tutor

Ing. Yojahny Chávez Marrero

Tutor

Agradecimientos

AGRADECIMIENTOS

Agradecer primeramente a mi familia, especialmente a mi madre, por siempre estar ahí, por saber guiarme y educarme durante cada proceso y etapa de mi vida, en esta ocasión, a la culminación de mis estudios, a este lindo proceso estudiantil. Agradecer también a mis amigos, a los más cercanos, a los que siempre han estado ahí dispuestos a ayudarme y apoyarme para lo que se presente; a esas personas que no lo piensan dos veces para ofrecer su mano sincera y su apoyo incondicional. A mis compañeros, a mis colegas, a mis hermanas de 5 años de vida, las que han transcurrido junto conmigo desde los primeros días, a las que llegaron después y se quedaron para siempre; por hacer honor a esos momentos de risas y alegrías, pero también, hasta los de mayor estrés. A mi novio que también ha formado parte de este proceso, a mis profesores, y de forma general a todos los que han participado de una manera u otra en esta hermosa experiencia.

Resumen

RESUMEN

El presente trabajo de diploma centra el estudio en la implementación de una plataforma de aprendizaje para contribuir a la formación de habilidades investigativas en el estudiantado universitario, específicamente para los estudiantes que afrontan la realización del trabajo de diploma. La presente tesis tiene como objetivo desarrollar una plataforma de aprendizaje basada en tecnología web para la formación de habilidades investigativas durante la realización del trabajo de diploma en el estudiante de Ingeniería en Ciencias Informáticas. Para la implementación de la plataforma, se utilizará el framework de desarrollo Laravel asistido por la consola Artisan. La utilización de UML como lenguaje para el modelado del sistema permitió visualizar y documentar el sistema. El análisis y diseño permitió obtener los artefactos correspondientes de la plataforma a partir de la adopción de la metodología de desarrollo AUP en la variante UCI Escenario 2. El lenguaje de programación utilizado HTML5 y SQL para la gestión de datos almacenados en la base de datos relacionales, también se utilizó Apache como servidor web para el manejo de las páginas web. El modelo conceptual de la plataforma se logró a partir del análisis orientado a objetos, para la realización de un correcto levantamiento de requisitos del sistema. La plataforma basa su implementación en el patrón Modelo-Vista-Controlador. El diseño de los diagramas de clases, facilitó el enfoque en cuanto a composición lógica y física de la propuesta de solución.

Palabras clave: formación de habilidades investigativas; plataforma virtual de formación; trabajo de diploma.

Índice general

TABLA DE CONTENIDOS

Introducción	1
Capítulo 1. Fundamentación Teórica	8
1. 1. Conceptos asociados de la investigación	8
1. 2. Análisis de soluciones existentes.....	21
1. 3. Lenguajes, herramientas, metodologías y tecnologías utilizadas.....	27
1.4.1. Metodologías de desarrollo de software	27
1.4.2. Modelado de software	29
1.4.3. Herramienta para control de versiones.....	30
1.4.4. Lenguaje de programación.....	30
1.4.5. Sistema Gestor de Base Datos	31
1.4.6. Servidor Web	32
1.4.7. Framework de desarrollo.....	33
1.4.8. Otras herramientas utilizadas para el desarrollo de la plataforma	33
1. 4. Conclusiones del capítulo	35
Capítulo 2. Descripción de la propuesta.....	37
2. 1. Características de la propuesta de solución	37
2. 2. Modelo Conceptual	39
2. 3. Especificación de requisitos de software	40
2.4. 1 Requisitos funcionales del sistema.....	42
2.4. 2 Requerimientos no funcionales del sistema.....	44
2. 4. Diagrama de Casos de Uso del Sistema	46
2. 5. Diseño de la propuesta de solución.....	47
2. 6. Patrón Arquitectónico:	47
2. 7. Patrones de Diseño.....	52
2. 8. Modelo de datos.....	55
2. 9. Verificación y validación	56
2. 10. Conclusiones del capítulo.....	57
Capítulo 3. Implementación y Pruebas.....	59
3. 1. Modelo de implementación. Diagrama de Despliegue	59
3. 2. Estándar de codificación y su justificación.....	60
3. 3. Pruebas de caja negra	66
3. 4. Validación por el método Criterio de Expertos.....	67

Índice general

3. 5. Conclusiones del capítulo.....	69
Conclusiones	71
Recomendaciones	72
Referencias bibliográficas	73
Anexos.....	79

Índice de tablas

INDICE DE TABLAS

Tabla 1	Aspectos aportados por los tutores y profesores en los cursos evaluados.	2
Tabla 2	Operacionalización de las variables.....	4
Tabla 3	Requisitos Funcionales del Sistema.	43
Tabla 4	Estándares de codificación a utilizar en la implementación del sistema.....	61

Índice de figuras

INDICE DE FIGURAS

Figura 1	Modelo conceptual.	40
Figura 2	Diagrama de Casos de Uso del Sistema.	47
Figura 3	Representación del MVC.	51
Figura 4	Diagrama de Clases del Diseño según MVC.	54
Figura 5	Modelo de Datos.	56
Figura 6	Modelo de Despliegue.	60
Figura 7	Representación del panel de expertos.	69

Introducción

INTRODUCCIÓN

Es necesario añadir la incorporación de las Tecnologías de la Información y las Comunicaciones al proceso de formación científica e investigativa de los futuros egresados de la carrera en Ciencias Informáticas, como al resto de los procesos formativos debe precederle la definición de estrategias para su uso y apropiación. No basta con disponer de computadoras y conexiones, ni con desarrollar capacidades técnicas para el uso instrumental de los recursos, sino que es necesario tener definidos los objetivos de su empleo, un cómo y un para qué se incorporan al proceso investigativo; lo que implica utilizarlas racionalmente como una herramienta para la solución de los problemas del entorno educativo.

La formación de habilidades investigativas constituye un factor clave en el estudiantado universitario. Desarrollar las capacidades para investigar problemas, efectuar evaluaciones, tomar decisiones racionales, son cuestiones fundamentales para la vida profesional.

Los estudiantes de la carrera de Ingeniería en Ciencias Informáticas, de acuerdo a su Plan de Estudio D, deben ser capaces de resolver problemas y proponer soluciones de informatización con rigor profesional, así como elaborar y presentar documentación científico-técnica con valor cultural, técnico y social. Para el logro de estos objetivos, se establece un sistema de influencias que incide sobre el estudiantado y que se articula desde las estrategias de formación integral establecidas por la institución universitaria.

Desde la asignatura Metodología de la Investigación Científica, correspondiente al primer semestre del tercer año, se imparten contenidos que potencian el desarrollo de habilidades para la redacción, comunicación y defensas de posiciones científico-técnicas, todo ello basado en actividades teórico-prácticas que contribuyen desde el aprendizaje presencial a la formación investigativa del Ingeniero en Ciencias Informáticas.

A partir del tercer año de la carrera, y con mayor énfasis en el quinto año como parte del trabajo de diploma, a cada estudiante se le asignan tareas, donde se requiera de consultar bibliografía científico-técnica y deben tener además resultados medibles, los cuales pueden ser: artefactos, documentos, informes, entre otros. Los resultados de estas tareas deben ser presentados en talleres, seminarios o reuniones de proyectos, donde se demuestren las habilidades investigativas que posee el estudiante.

Por otra parte, el Comité de Tesis de la Facultad 1, órgano metodológico responsabilizado de la planificación, organización y el control del proceso de trabajos de diploma, cada año desarrolla acciones formativas encaminadas a la sistematización de conocimientos en torno a la

Introducción

investigación científica y la ingeniería y gestión de software, además de los talleres de tesis, dirigidos por tribunales que integran profesores y especialistas de centros de desarrollo de software, responsabilizados con emitir una valoración de los avances presentados por los diplomantes.

Los informes de balance del Comité de Tesis de la Facultad 1, en los últimos dos cursos (2017-2018 y 2018-2019), se pueden evidenciar los resultados de la gestión de este proceso, donde se identifican debilidades relacionadas con la escritura académica, la comunicación científica, y la ingeniería y gestión de software. La consulta a tutores y profesores que forman parte de los tribunales evaluativos, también ha arrojado valoraciones que refuerzan los hallazgos anteriores.

El análisis documental de las actas de los tribunales de ejercicios de culminación de estudio para los cursos 2017-2018 y 2018-2019 permitieron identificar las principales debilidades detectadas en los informes o memorias escritas, en el proceder de ingeniería de software para el desarrollo de los sistemas propuestos que responden al problema científico, y por último se detectan las principales debilidades en la comunicación o presentación oral de los resultados en los actos de defensa.

A partir de lo anterior fue posible establecer un nivel de categorías teniendo en cuenta las semejanzas de los señalamientos realizados por los tribunales, lo cual constituye un criterio interno de evaluación, teniendo en cuenta las debilidades identificadas.

A continuación, se clasifican las valoraciones obtenidas por tutores y profesores que forman parte de los tribunales evaluativos.

Tabla 1 Aspectos aportados por los tutores y profesores en los cursos evaluados.

Criterio de agrupamiento	Aspectos aportados por tutores y profesores
Escritura Académica	<ul style="list-style-type: none">- Bajo número de referencias bibliográficas de los últimos 5 años.- Uso inadecuado de la norma bibliográfica.- Dificultades en la redacción de los componentes del diseño investigativo.- La relación problema de investigación-objeto de estudio no se encuentra bien determinada.
Comunicación Científica	<ul style="list-style-type: none">- Se identifican debilidades en la exposición oral.- La presentación digital utilizada no cumplía requisitos calidad.- No realiza acciones para la divulgación científica de los resultados logrados.
Ingeniería y Gestión de Software	<ul style="list-style-type: none">- El software aportado no respondió a todos los requisitos identificados inicialmente.- El modelo de datos lógicos del sistema presentaba deficiencias en su diagramación.- No se hace un adecuado proceso de gestión de requisitos de software.

Introducción

En los informes del cumplimiento de los Objetivos de Trabajo, específicamente en el área de resultado clave Formación de Pregrado, se constata que en los cursos 2017-2018 y 2018-2019 en la Facultad 1, se han desarrollado seminarios y talleres científicos, para aumentar la preparación de los estudiantes en proceso de trabajo de diploma. No obstante, estas actividades se han centrado básicamente en la orientación a los estudiantes en el uso de materiales para soportar su formación, sin embargo, el contexto actual del Ingeniero en Ciencias Informáticas demanda una preparación más eficaz, que implemente la tecnología como objeto de apoyo, para contribuir en el desarrollo de más habilidades investigativas en el estudiante.

Para disminuir el impacto negativo que producen estas condiciones sobre los estudiantes de quinto año, los métodos actuales de formación investigativa, que dependen en alto grado de la presencialidad, deben ser reforzados por esquemas sustentados en las TIC.

Problema de Investigación: ¿Cómo favorecer la formación de habilidades investigativas desde un entorno virtual, durante la realización del trabajo de diploma en el estudiante de Ingeniería en Ciencias Informáticas?

Objeto de Estudio: Diseño y desarrollo de plataformas de aprendizaje virtual.

Objetivo general: Desarrollar una plataforma de aprendizaje basada en tecnología web para la formación de habilidades investigativas durante la realización del trabajo de diploma en el estudiante de Ingeniería en Ciencias Informáticas.

Objetivos específicos:

- Caracterizar los fundamentos teóricos relacionados con el desarrollo de plataformas web para la formación de habilidades investigativas.
- Definir las tecnologías, las herramientas y la metodología para la implementación de la plataforma Web para la formación de habilidades investigativas.
- Diseñar la plataforma Web para la formación de habilidades investigativas.
- Validar la solución propuesta a partir del cumplimiento de los objetivos definidos en la investigación para la problemática existente.

Campo de acción: Diseño y desarrollo de aplicaciones web para la formación de habilidades investigativas.

Hipótesis: El desarrollo de una plataforma de aprendizaje basada en tecnología web, contribuirá a la formación de habilidades investigativas para la escritura académica, la comunicación

Introducción

científica, así como para la ingeniería y gestión de software durante la realización del trabajo de diploma en el estudiante de Ingeniería en Ciencias Informáticas.

Variables de Investigación:

Variable Independiente:

- 1) Plataforma web para la formación de habilidades investigativas

Variable dependiente:

- 1) Contribución a la formación de habilidades investigativas para la escritura académica durante la realización de la Tesis.
- 2) Contribución a la formación de habilidades investigativas para la comunicación científica durante la realización de la Tesis.
- 3) Contribución a la formación de habilidades investigativas para la ingeniería y gestión de software durante la realización de la Tesis.

Tabla 2 Operacionalización de las variables

Variable	Dimensiones	Indicadores	Índices/Instrumentos	
Plataforma web para la formación de habilidades investigativas	Procesos y requerimientos	Requerimientos funcionales	Documento de Especificación de requerimientos. Metodología de desarrollo	
		Requerimientos no funcionales		
		Flujo de proceso		
	Diseño Arquitectura web que cumpla con los requerimientos para la gestión de los procesos de diseño del sistema	Procesos del sistema	Vista despliegue	Vista lógica Vista física Vista procesos
		Vista lógica de procesos		
		Vista de desarrollo de procesos		
	Herramientas y tecnologías necesarias para la instalación y codificación de los requerimientos del sistema	Codificación de capa de presentación	Laravel Software HTML. PHP.	
		Codificación de capa de datos		
		Codificación de capa de negocio		
	Pruebas de verificación y validación de software	Cantidad de errores	Documento de validación y verificación. Pruebas unitarias. Prueba de caja negra. Prueba caja blanca. Pruebas de validación	
		Acuerdos de modificación		
		Resultados satisfactorios		
Medición del nivel de requerimientos				

Introducción

Variable	Dimensiones	Indicadores	Índices/Instrumentos
Contribución a la formación de habilidades investigativas para la escritura académica durante la realización de la Tesis	Redacción del título	PreTest	Escala valorativa de 0 (Nada) a 5 (Máximo)
	Redacción del resumen	Contenidos	
	Redacción de palabras clave	Elaboración propuesta personal	
	Elaboración del índice de contenidos	Coevaluación	
	Elaboración de la bibliografía	Autovaloración del aprendizaje	
	Redacción del problema de investigación	PosTest	
	Redacción de los objetivos		
Contribución a la formación de habilidades investigativas para la comunicación científica durante la realización de la Tesis	Comunicar resultados en eventos	PreTest	Escala valorativa de 0 (Nada) a 5 (Máximo)
	Diseñar presentaciones atractivas	Contenidos	
	Uso redes sociales para la investigación	Elaboración propuesta personal	
Contribución a la formación de habilidades investigativas para la ingeniería y gestión de software durante la realización de la Tesis	Requisitos de Software	Coevaluación	Escala valorativa de 0 (Nada) a 5 (Máximo)
	Modelación de Datos	Autovaloración del aprendizaje	
	Gestión de Software	PosTest	

Introducción

A continuación, se detallan los métodos utilizados en la presente investigación:

Métodos teóricos:

- **Analítico - Sintético:** a través de este método se realiza el análisis de la solución propuesta, mediante las funcionalidades que deben ser desarrolladas para cada módulo de la Plataforma de Formación de Habilidades Investigativas con el objetivo de mejorar, contribuir y soportar al trabajo investigativo del estudiante de Ingeniería en Ciencias Informáticas durante su proceso de culminación de estudios.
- **Inductivo - Deductivo:** permitió arribar a conclusiones generales sobre la gestión de la información en cuanto a los conocimientos presentes en los estudiantes para su futuro enfrentamiento a los procesos de tesis. Ayudó a definir la estructura funcional del sistema como propuesta de solución a partir del análisis de necesidades de estudiantes y profesores apoyado en el estudio de soluciones existentes.

Métodos empíricos:

- **Modelación:** se utilizó para representar a partir de modelados el escenario de la problemática en cuestión, lo cual permite, a partir de los diseños de diagramas, una mayor comprensión de lo concerniente a la organización del flujo de la información de la solución propuesta.
- **Entrevista:** se empleó para recopilar información referente a los procesos de enseñanza y aprendizaje tanto de forma presencial como no asistida a estudiantes y profesores, los cuales aportaron información de valor apoyándose en su experiencia. Los profesores encargados de la formación investigativa aportaron valiosas ideas relacionadas con el diseño didáctico de la plataforma para asistir la formación de habilidades.
- **Observación:** se empleó en diferentes momentos y en diversos entornos y áreas de la enseñanza, se estudiaron los procesos involucrados, así como las deficiencias o carencias detectadas en la recopilación o gestión de la información para estos procesos.
- **Criterio de expertos:** con el objetivo de evaluar la relevancia e implicaciones prácticas, así como la utilidad metodológica de la plataforma de formación de habilidades investigativas, mediante una técnica de comunicación estructurada basada en un panel de expertos.

La presente investigación consta de una estructura capitular compuesta por tres capítulos:

Capítulo 1: Fundamentación teórica. En este capítulo se presentan los elementos teóricos que sirven de sustento a la investigación. Se fundamentan conceptos, se clasifican y analizan

Introducción

determinadas herramientas, plataformas y entornos que se enmarcan dentro del ambiente de la investigación. Se plasma el estudio realizado a los diferentes sistemas, gartuitos y comerciales, existentes tanto nacional como internacionalmente. Se caracteriza la metodología de desarrollo a utilizar para la implementación de la propuesta de solución.

Capítulo 2: Descripción de la solución propuesta. En este capítulo se describe el negocio. Se realiza la modelación del proceso del negocio, en donde se logra plasmar a partir de una representación abstracta el ambiente del problema. Se definen las técnicas para la obtención de los requisitos funcionales y no funcionales. Se describen la arquitectura y los patrones de diseño utilizados. También, se elabora el diagrama de casos de uso del sistema, el diagrama de clases del diseño y el modelo de base de datos.

Capítulo 3: Validación y pruebas. En el capítulo se tratan las futuras pruebas a aplicar para verificar el correcto funcionamiento de los módulos del sistema para el desarrollo de habilidades investigativas, para ello se realizará una validación funcional de la solución propuesta. Se describen los estándares de codificación, nomenclatura. Se hace referencia a las pruebas de partición de equivalencia, dentro de las correspondientes a caja negra. Además se propone, para implementar a futuro, el proceso de pruebas de validación Criterio de Expertos.

Luego las conclusiones, recomendaciones, bibliografía y anexos de la tesis.

Capítulo 1. Fundamentación teórica

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

El estudio de conceptos y características generales asociados a las plataformas de gestión para la formación investigativa en general y de sistemas para la formación de habilidades investigativas en los estudiantes universitarios, constituyen elementos clave en la elaboración del presente marco teórico. También, se realiza el análisis de algunas herramientas existentes para la gestión e informatización referentes al objetivo de estudio. Se caracteriza el proceso de desarrollo de software, lenguajes de programación y tecnologías que se utilizan para el desarrollo de software en la web.

1. 1. Conceptos asociados de la investigación

A continuación, se muestran algunos conceptos importantes relacionados con el problema planteado, los cuales permiten una mejor comprensión de los aspectos a tratar en la investigación.

Habilidades Investigativas:

(Rodríguez, y otros, 2014), quienes conceptualizaron las habilidades investigativas como dominio de acciones (psíquicas y prácticas) que permiten la regulación racional de la actividad, refieren que dichas habilidades representan el dominio del contenido de la formación para la investigación (sistema de conocimientos, habilidades y valores), además que permiten determinar las necesidades de la formación para la investigación.

Para (Machado, y otros, 2019), son conocimientos y hábitos que el sujeto posee para ir a la búsqueda del problema y a la solución del mismo, por lo que resaltan la importancia de desarrollarlas en la Educación Superior a partir de la socialización. Lo anterior se relaciona con lo expuesto por (Torres, y otros, 2018), cuando refieren que las habilidades investigativas permiten que los estudiantes aprenden, construyen y se preparen de manera integral para transformar su realidad, considerando que se ponen en juego acciones que favorecen el análisis, la búsqueda de soluciones a problemas a los que se enfrentan, la reflexión crítica, entre otros.

En el caso de (Martínez , y otros, 2018), detallan que dichas habilidades implican valores éticos profesionales y cualidades de la personalidad, al propiciar la independencia, la responsabilidad, la creatividad, el compromiso con la profesión seleccionada, así como una valoración adecuada de ella; además permiten el desarrollo de un pensamiento científico.

(Miranda , 2014), fundamenta la concepción pedagógica de las Habilidades Investigativas, a partir de un estudio que realizó con estudiantes de medicina donde recupera la relación triádica

Capítulo 1. Fundamentación teórica

entre modos de solución de problemas, fundamentos epistemológicos y el proceso de formación en habilidades investigativas como elementos articuladores para actividades curriculares y extracurriculares, lo que permite considerar a la investigación como herramienta para la solución de problemáticas de disciplinas como la Medicina, entre otras.

Clasificaciones existentes sobre las habilidades investigativas:

Dentro de las clasificaciones más generales de las habilidades investigativas se encuentran (Rodríguez , et al., 2014):

1. Habilidades básicas de investigación, habilidades propias de la ciencia particular y habilidades propias de la metodología de la investigación pedagógica (Balboa, 2001).
2. Habilidades para problematizar, teorizar y comprobar la realidad objetiva (Ramos, 2002).
3. Habilidades de percepción, instrumentales, de pensamiento, de construcción conceptual, de construcción metodológica, de construcción social del conocimiento y meta cognitivas (Moreno, 2005).
4. Habilidades investigativas de mayor integración para la enseñanza del pregrado tales como: solucionar problemas profesionales, modelar, ejecutar, obtener, procesar, comunicar información y controlar (Ramírez, et al., 2008).

Aprendizaje no presencial:

El profesor y el alumno no se encuentran en la misma dimensión-espacio temporal, proporcionando una gran autonomía geográfica y temporal. La enseñanza se realiza a través del trabajo exploratorio de los alumnos que adquiere un comportamiento activo en el proceso de aprendizaje. Es el proceso de aprendizaje que incluye la enseñanza apoyada en las Tecnologías de la Información y la Comunicación (TIC). Incorpora instrucción indirecta y aprendizaje asistido por computadora individualizado. Este aprendizaje requiere actitud correcta y estudiantes altamente motivados para su implementación exitosa.

Los entornos virtuales de formación:

El uso de las plataformas de tele formación es una tendencia bastante difundida a nivel mundial, las cuales son conocidas también como: entornos virtuales de enseñanza/aprendizaje, plataformas e-learning o plataformas virtuales de formación. Estos espacios virtuales están diseñados para la implementación de actividades formativas en las diversas modalidades de estudio, ya sea totalmente a distancia, presencial o mixto (Calzada, 2013).

Según Gisbert et al. (1998) —un entorno virtual de enseñanza/aprendizaje (EVE/A) es un

Capítulo 1. Fundamentación teórica

conjunto de facilidades informáticas y telemáticas para la comunicación y el intercambio de información en el que se desarrollan procesos de enseñanza/aprendizaje, en el que interactúan, fundamentalmente, profesores y estudiantes”.

Desarrollo del software educativo:

En la actualidad el auge alcanzado por las TIC ha impulsado el uso de la informatización en todas las esferas de la sociedad, lo cual no deja de lado el ámbito educacional. La utilización de dicha tecnología en la educación ha causado un gran impacto en la calidad del PEA, permitiendo una mayor interacción estudiante-profesor-tecnología. A las herramientas utilizadas para dichos fines se les conoce como software educativo.

El autor (Cataldi, et al., 2000) define estas herramientas como “los programas de computación realizados con la finalidad de ser utilizados como facilitadores del proceso de enseñanza y consecuentemente del aprendizaje, con algunas características particulares tales como: la facilidad de uso, la interactividad y la posibilidad de personalización de la velocidad de los aprendizajes”.

Por otra parte, el autor (Marquès, et al., 1995) plantea que la terminología software educativo es utilizada para designar genéricamente los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje. También tienen su propia definición el autor (Ledo, 2010) que plantea de forma genérica a los softwares educativos como aplicaciones o programas computacionales que faciliten el proceso de enseñanza aprendizaje.

Herramientas de gestión de aprendizaje

Dentro de los sistemas y plataformas interactivas que permiten la producción de contenidos digitales educativos se encuentran los sistemas de gestión de contenidos (Content Management System –CMS. Ej.: Drupal, Joomla, etc.), de administración y control de las actividades de formación (Learning Management System –LMS. Ej.: SumTotal, Librarian, etc.), así como las plataformas interactivas que integran ambas funciones (Learning Content Management System –LCMS. Ej.: Moodle, Dokeos, WebCT, etc.) (Rodríguez, y otros, 2015).

CMS

Un Sistema de Gestión de Contenido-CMS (Content Management System, en inglés) es un software que permite la creación y administración de los contenidos de una página Web, principalmente, de forma automática. Así, con él se puede publicar, editar, borrar, otorgar

Capítulo 1. Fundamentación teórica

permisos de acceso o establecer los módulos visibles para el visitante final de la página. El CMS está formado por 2 elementos:

- La aplicación gestora de contenidos (CMA): El elemento CMA permite al gestor de contenidos o autor realizar la creación, modificación y eliminación de contenido en un sitio Web sin necesidad de tener conocimientos de lenguaje HTML.
- La aplicación dispensadora de contenidos (CDA): El CDA usa y compila la información para actualizar el sitio Web.

El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. De esta manera, es posible manejar el contenido y variar en cualquier momento el diseño del sitio sin tener que darle formato a todo el contenido de nuevo. Además, permite de manera fácil y controlada la publicación en el sitio a varios editores –con la debida autorización previa– para que éstos puedan escribir artículos, proponer votaciones, definir la apariencia y el funcionamiento de los interfaces gráficos del sitio, entre otros aspectos.

En definitiva, hablar de un CMS es referirse a una interfaz que controla una o varias bases de datos donde se aloja el contenido de un sitio. Las características de un CMS pueden variar, pero la mayoría incluye publicación basada en Web, indexación, revisión, búsqueda y recuperación de la información. (Mayor, 2016)

LMS

Un sistema de gestión de aprendizaje LMS (Learning Management System) es un software que automatiza la administración de acciones de formación. Son variadas las funcionalidades de un LMS, registra a todos los actores que intervienen en el acto de aprendizaje (alumnos, profesores, administradores, etc.), organiza los diferentes cursos en un catálogo, almacena datos sobre los usuarios, realiza un seguimiento del aprendizaje y la temporización de los trámites y genera informes automáticamente para tareas de gestión específicas.

Habitualmente, el LMS lo emplean las organizaciones que poseen un volumen considerable de cursos diferentes y para ello se les hace necesario el uso de una herramienta tecnológica que les permita gestionarlos con efectividad y de forma práctica. La mayoría de LMS están basadas en la Web para facilitar en cualquier momento y en cualquier lugar y a cualquier ritmo el acceso a los contenidos de aprendizaje y administración.

Al igual que ocurre con los CMS, se puede encontrar LMS de diferente tipología, ya sea bajo licencias propietarias o LMS tipo *open source*. En definitiva, podríamos decir que un LMS puede

Capítulo 1. Fundamentación teórica

considerarse como un CMS de propósito específico, concretamente educativo, que potencia las posibilidades de colaboración e interactividad que puede ofrecer un espacio virtual, siempre que se empleen los recursos adecuados para tal finalidad. Por tanto, los LMS se adecuan a los propósitos específicos del denominado e-Learning (aprendizaje en línea) (Mayor, 2016).

Un LMS proporciona, por tanto:

- Tecnologías transmisivas (por ejemplo, mediante una presentación que se agregue al curso).
- Tecnologías interactivas (mediante cuestionarios, actividades, etc.).
- Tecnologías colaborativas (mediante foros, wikis, etc.)

LCMS

Un Sistema de Gestión de Contenidos de Aprendizaje LCMS (Learning Content Management System, en inglés) es una aplicación de software que combina las capacidades de gestión de cursos de un LMS con las capacidades de almacenamiento y creación de contenidos de un CMS. Los LCMS se acercan a la denominación en castellano de “campus virtuales”. Permite la creación y el desarrollo eficiente de contenidos para el aprendizaje proporcionando las herramientas necesarias a autores, diseñadores instruccionales y expertos del tema.

Normalmente se crean partes de contenido en forma de módulos que se pueden personalizar, manejar, y que se pueden usar en diferentes ocasiones (son los denominados objetos de aprendizaje). Así, en lugar de apoyar el desarrollo totalizador de cursos, lo que hace un LCMS es ayudar a diseñadores instruccionales a crear módulos o bloques de contenido reutilizable, que luego se distribuirán según convenga en cada caso, por los diseñadores de los cursos (Mayor, 2016).

Diferencias entre un CMS y un LMS:

Con el objetivo de precisar de forma simple cuáles son las diferencias fundamentales, desde el punto de vista de gestión de contenidos, entre un CMS y un LMS, se puede decir que el LMS administra y gestiona contenidos formativos y el CMS de cualquier ámbito, otra diferencia es que el LMS permite herramientas de comunicación para la actividad docente, mientras que el CMS no las incluye, ya que su finalidad está más orientada a la gestión de contenidos, más que a la gestión del aprendizaje (Mayor, 2016).

Diferencias entre un LMS y un LCMS:

Capítulo 1. Fundamentación teórica

Un LMS y un LCMS, aunque complementarios, son dos sistemas muy diferentes que sirven para distintos propósitos. Se puede decir que un LMS es un software que planifica y gestiona los eventos de aprendizaje dentro de una organización, incluyendo el aula online o virtual y los cursos dirigidos por un instructor. Por el contrario, un LCMS es un software para la gestión de contenidos de aprendizaje de los diversos programas de capacitación que se configuran en el desarrollo en toda la organización. En este segundo caso, se proporciona a los desarrolladores, autores, diseñadores instruccionales y expertos en la materia los medios para crear y reutilizar el contenido de aprendizaje y reducir la duplicación de los esfuerzos de desarrollo, ya que un LCMS crea, almacena, ensambla y entrega de forma personalizada el contenido en forma de objetos de aprendizaje específicos. Un LMS nos ayuda a gestionar la administración de una enseñanza o de un curso en todos sus aspectos: gestionar usuarios, recursos, actividades, módulos, permisos, generar informes, evaluaciones, calificaciones, comunicación de foros, videoconferencias, chats y demás. Un LCMS es lo mismo que un LMS, pero con una particularidad, que es la propia de los CMS: el hecho de poder administrar todos los contenidos del sistema (Mayor, 2016).

Principales herramientas tanto para la gestión del aprendizaje como para la gestión de contenidos del aprendizaje (LMS/LCMS):

Estas herramientas se pueden clasificar por *gratuitos*, *comerciales* y en la *nube*, por ello a continuación se presentan las principales de cada clasificación con una breve descripción:

Plataformas educativas gratuitas:

Moodle

MOODLE es un Entorno Virtual de Enseñanza Aprendizaje, un programa informático que permite diseñar, estructurar y realizar diversos procesos formativos a través de Internet. Fue creado en 2002 por Martin Dougiamas, quien era en ese momento administrador de otro EVEA (Entorno Virtual de Enseñanza Aprendizaje) privado, WebCT, para la Universidad Tecnológica de Curtin. La primera versión de MOODLE como herramienta apareció el 20 de agosto de 2002 y, a partir de allí han aparecido nuevas versiones de forma más o menos regular. La aparición de dichas versiones se ha visto estimulada por el hecho de que MOODLE se distribuye gratuitamente como software libre, bajo la denominada licencia GNU; es decir, sí tiene derechos de autor, pero viene con autorización para que cualquiera pueda usarlo, copiarlo y distribuirlo, ya sea literal o con modificaciones, gratis o mediante una gratificación. Además, cualquier usuario está autorizado no sólo a usar el programa con cualquier propósito, sino a estudiar cómo funciona y adaptarlo a

Capítulo 1. Fundamentación teórica

sus necesidades, distribuir copias y mejorar el programa, siempre que haga públicas las mejoras a los demás usuarios actuales y potenciales (Quintero , et al., 2007). Otra de sus ventajas es que es compatible con estándares como SCORM, IMS entre otros. También se encuentra disponible para cualquier sistema operativo, solo necesita PHP y MySQL (Lagunes, et al., 2015).

[\(http://moodle.com/\)](http://moodle.com/)

Atutor

El portal de esta plataforma indica que es de código abierto y que se basa en la gestión de contenidos de aprendizaje. Atutor permite el ingreso a estudiantes, profesores, administradores y también a personas con capacidades diferentes. En la parte técnica fue desarrollado en PHP, un poco de Java y como sistema manejador de base de datos utilizan MySQL (Lagunes, et al., 2015).

Chamilo

Chamilo fue creado en el 2010, su comunidad ha crecido mucho, lo cual significa que ha tenido una buena aceptación entre instituciones educativas y profesores. Tiene la ventaja de poderse instalar casi en todos los sistemas operativos, Windows, Linux y OSX y fue desarrollado también con PHP y MySQL (Lagunes, et al., 2015).

Dokeos

Dokeos no es solo una plataforma sino una compañía que ofrece servicios de hospedaje, soporte y otros servicios de e-learning. Esta plataforma es de código abierto, permite administrar contenidos, chat, audio, video y también cuenta con un proceso de entrenamiento, es utilizado en casi 10,000 organizaciones (Lagunes, et al., 2015).

Claroline

Claroline es una plataforma de Gestión del Aprendizaje de libre distribución, desarrollada por la Universidad Católica de Louvain. Permite a los diferentes usuarios administrar cursos virtuales desde un navegador de internet. Las características más importantes son:

La publicación de documentos en diferentes tipos de formatos.

- Posibilidad de creación de foros públicos y/o privados.
- Creación de grupos de alumnos.
- Generación de exámenes/pruebas de evaluación.

Capítulo 1. Fundamentación teórica

- Elaboración de una agenda con diferentes tareas y fechas claves en el curso.
- Posibilidad de publicación de noticias en un tablón de anuncios, además del uso del correo electrónico.

Desde un punto de vista técnico, Claroline, necesita un entorno concreto para su correcto funcionamiento. Este entorno deberá estar formado por los siguientes elementos:

- Servidor Web: Se recomienda la instalación y configuración del servidor web Apache.
- Gestor de base de datos, se usará MySQL.
- Instalación del entorno de programación PHP como lenguaje de servidor.

Claroline es una herramienta viva y en constante cambio, a través de la dirección web, se reciben iniciativas a través de los mensajes y foros de discusión, para cambiar y mejorar la aplicación. Otro tema importante, es la documentación ofrecida, ya que se encuentra disponible en ocho idiomas, además de dividir la propia documentación en una parte para el profesor y otra parte para el alumno. Ni que decir tiene, que es una de las herramientas de teleformación más extendidas y más utilizadas en la actualidad (Roberto , et al., 2006).

[\(http://www.claroline.net/\)](http://www.claroline.net/)

Plataformas educativas Comerciales:

WebCT

La plataforma de enseñanza virtual WebCT pone a disposición de toda su comunidad un conjunto de utilidades para la docencia a través de Internet, que permite complementar a la enseñanza presencial, y que facilita además la educación a distancia. Representa una gran interactividad y sencillez de uso, que simplifica la tarea de comunicación entre profesores y alumnos. Asimismo, WebCT ofrece al profesor un conjunto de herramientas para la monitorización de los estudiantes, creación de exámenes, facilidades para el control del acceso al curso, asignación de claves, transferencias de ficheros, etc (Martín, et al., 2014).

<http://www.webct.com>

Blackboard

Esta plataforma de paga es una de las más utilizadas, en México la utilizan las universidades privadas más importantes del país. En otros países como Colombia también es de las más utilizadas. La estructura de la plataforma es módulo de contenidos, herramientas de comunicación, herramientas de evaluación y herramientas de seguimiento y gestión de

Capítulo 1. Fundamentación teórica

aprendizaje. Una ventaja que tiene es que es muy flexible y dentro de esto permite la integración de las redes sociales y el uso de dispositivos (Lagunes, et al., 2015).

Edu2.0

Esta es de las pocas plataformas que desde inicio indican que son más útiles para B-learning, esto es, sirve de apoyo para las clases presenciales de los profesores, desde luego que no se descarta para cursos 100% en línea. Una de las bondades de Edu2.0 es que es fácil de utilizar y no requiere de instalación, se pueden tener los cursos de forma remota sin la necesidad de tener un servidor (Lagunes, et al., 2015).

E-educativa

"E-learning más simple, más fácil ..." así es como se promociona esta plataforma que dicho sea de paso, sí cumple con su lema. También es posible ver en su sitio web que son más de 6,000 instalaciones alrededor del mundo y tiene dentro de sus usuarios a empresas muy grandes como Toyota, Bayer y algunas universidades. E-educativa es muy completa, ofrece paquetes de Campus, ejecutiva, escuela suite, videoconferencias, y contenidos. Entre sus servicios tiene escritorio, aulas virtuales, blog, página web entre otros (Lagunes, et al., 2015).

FirstClass

Esta es una plataforma comercial que en estos momentos se encuentra en la versión 12, entre sus servicios ofrece blog, almacenamiento en la nube, correo electrónico, mensajería, almacenamiento y compartición de archivos, entre otros servicios. Al igual que las plataformas anteriores tiene una interfaz amigable y al ser de pago cuenta con un buen soporte técnico (Lagunes, et al., 2015).

Nixty

La última plataforma de esta clasificación es Nixty, la cual ofrece blog, foros, libro de calificaciones en línea, mensajes, wikis entre otros. En cuanto a su precio, ofrece cursos desde casi 5 dólares hasta 10 dólares. La Plataforma de encarga de anunciar los cursos, de cobrar el curso y por ello te cobra el 20% del cobro, es una buena alternativa si se desea vender un curso (Lagunes, et al., 2015).

Plataformas educativas en la nube:

Ecaths

Esta es una plataforma argentina con una idea similar a Edu2.0 en su orientación, esta pensada para complementar las clases presenciales con una plataforma versátil, aunque en su sitio web

Capítulo 1. Fundamentación teórica

se presenta como versión beta. Tiene la ventaja de ser gratuita y contiene información del curso, cronograma, bibliografía, texto digitalizado, y trabajos prácticos en línea entre otras actividades. También cabe hacer mención que su interfaz es sencilla (Lagunes, et al., 2015).

Edmodo

Esta plataforma tiene la virtud de ser dos en una, es una plataforma educativa combinada con las funciones de una red social, y su principal ventaja es ser gratuita. Otra de sus ventajas es que promueve la comunicación, la colaboración y la compartición de recursos entre estudiantes y profesores, lo cual la hace una plataforma muy poderosa en la comunicación. Quizás una de sus principales ventajas es que es muy simple e intuitiva, algunos usuarios indican que es similar a Facebook, cuenta con un muro y los profesores pueden crear grupos y subgrupos (Lagunes, et al., 2015).

Schoology

Una de las plataformas más utilizadas dentro de esta clasificación es Schoology, y tiene su parte gratuita para profesores y la parte comercial vendida a empresas. Dentro de sus servicios tiene integración con Google, sus calendarios se sincronizan con Outlook y Google Calendar, permite mensajería y correo, soporta SCORM 2004, entre otros servicios interesantes. Al igual que Edmodo funciona como una red social, cuenta con un muro para la publicación, todos tienen su perfil personal y también pueden crear discusiones, es una interfaz diferente para quienes han trabajado con otras plataformas (Lagunes, et al., 2015).

Udemy

Nació en Turquía y poco a poco fue creciendo y migrando sus actividades para Estados Unidos, después de muchos esfuerzos mejoró sus funcionalidades y ahora permite a los instructores crear cursos permitiendo subir archivos PDF, archivos de audio, presentaciones Power Point y Archivos Zip. Udemy es una base de datos de cursos creados por profesores de todas partes del mundo y entre sus ventajas esta que la mayoría de los cursos son gratuitos, habiendo otros que son mediante pago. Al igual que otras plataformas si el profesor desea cobrar por el curso, la plataforma cobra el 30% y ofrece las bondades y seguridad de la plataforma (Lagunes, et al., 2015).

Estas plataformas se relacionan entre sí, pues están encaminadas de forma particular a la gestión del aprendizaje, ambiente por el que se realiza el estudio de la investigación; no obstante, como bien se pudo analizar, son pocas las gratuitas, pues la mayoría están desarrolladas para

Capítulo 1. Fundamentación teórica

empresas y organizaciones dentro del sector privado, con fines y atributos meramente comerciales.

Herramientas modernas para la gestión de la información científica

Cuando se realiza investigación la primera problemática es encontrar información confiable, por ello la primer herramienta de apoyo a la investigación a analizar son las Bases de datos electrónicas y entre ellas esta EBSCOhost® una de las bases de datos más grande del mundo, con miles de revistas en línea y millones de artículos (EBSCO, 2020).

Una segunda opción es ScienceDirect quien también cuenta con miles de revistas científicas y más de 9 millones de artículos en las áreas científica, tecnológica y médica (ELSEVIER, 2020). Otra base de datos es Google Académico el cual ayuda a encontrar el material más relevante dentro del mundo de la investigación académica y la ventaja que tiene es que todas las personas tienen acceso a él.

La segunda problemática es la administración de las fuentes de información, para ello, se recurre a herramientas como EndNote, la cual administra las fuentes de información y puede sincronizar con dispositivos móviles, computadora o directamente en línea. Creada por Thomson Reuters, lo cual ya da muestras de su calidad, su único inconveniente es que es comercial (Thomson Reuters, 2020).

Una segunda herramienta para administrar las fuentes de información es Mendeley® la cual es un gestor de referencias gratis y red social académica, permite hacer una biblioteca propia completa en muy poco tiempo, citar a medida que escribe, y leer y anotar sus archivos PDF en cualquier dispositivo (Mendeley, 2020).

Otra herramienta de este tipo es Reference Manager®, la cual es de paga y tiene muy buenos servicios como buscar en las bases de datos en línea, organizar referencias fácilmente, y ver la bibliografía aparecen a medida que escribe (Thomson Reuters, 2020).

Finalmente existen dos opciones para administrar las fuentes, una es Word el cual tiene incluido dentro de su procesador la opción de referencias y allí se puede llevar el control de las fuentes y la otra es Zotero que permite recolectar y administrar las fuentes de información, funciona mediante el navegador Firefox y puede ser agregado como un complemento a Word (Zotero, 2020).

Estas herramientas, tienen la particularidad de soportar a la investigación científica, pues ellas son las encargadas de gestionar y administrar fuentes confiables, ya que están integradas a

Capítulo 1. Fundamentación teórica

bases de datos con grandes volúmenes de información de impacto, las mismas poseen atributos distintivos que permiten al investigador obtener un resultado de calidad, más acertivo y cercano a sus intereses, por ende también son motivo de estudio, pues se hacen evidentes las deficiencias de estudiantes para realizar una adecuada investigación que esté respaldada por documentación veraz, por lo que sería posible además el acceso al conocimiento para la correcta utilización de estas herramientas que se enmarcarían a la estructura evaluativa de la futura plataforma, dirigida específicamente a favorecer la escritura académica en los propios estudiantes.

Framework y herramientas de gestión del aprendizaje

Ventajas de usar un Framework:

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Una de las principales razones de usar esta tecnología, además, es por el hecho de poder implementar de forma efectiva la actividad coevaluativa dentro de este sistema.

Ventajas de los Frameworks MVC

El uso de los frameworks basados en este patrón permite tener una separación lógica y física de los componentes de la aplicación, ya que por un lado se tienen los modelos, por otro las vistas y por otro los controladores. De esta forma, los desarrolladores de la aplicación pueden centrarse en la parte que les toca, ya sea como diseñadores en las vistas, o como programadores de los modelos del negocio. Los frameworks ofrecen una elevada organización en el trabajo, ya que todo parece tener un sitio, aunque siempre existen cosas que son difíciles de acomodar, pero generalmente se obtiene mucha más organización que cuando se hace el layout de carpetas y la organización de los archivos manualmente. Generalmente estos frameworks poseen generadores que crean los archivos base de los modelos o vistas, para no tener que crear cada archivo relacionado a mano (Romero, y otros, 2012).

Ventajas de usar Laravel

- Laravel es un framework PHP. Es uno de los frameworks más utilizados y de mayor comunidad que cuenta con muy buena documentación en el mundo de Internet.

Capítulo 1. Fundamentación teórica

- Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web.
- Un sistema de rutas, mediante las cuales es fácil crear y mantener todo tipo de URLs amigables a usuarios y buscadores, rutas de API, etc.
- Un sistema de abstracción de base de datos, con un ORM potente pero sencillo de manejar, mediante el que podemos tratar los datos de la base de datos como si fueran simples objetos.
- Un sistema para creación de colas de trabajo, de modo que es posible enviar tareas para ejecución en background y aumentar el rendimiento de las aplicaciones.
- Varias configuraciones para envío de email, con proveedores diversos.
- Un sistema de notificaciones a usuarios, mediante email, base de datos y otros canales.
- Una abstracción del sistema de archivos, mediante el cual podemos escribir datos en proveedores cloud, y por supuesto en el disco del servidor, con el mismo código.
- Gestión de sesiones.
- Sistema de autenticación, con todo lo necesario como recordatorios de clave, confirmación de cuentas, recordar un usuario logueado, etc.
- La posibilidad de acceder a datos en tiempo real y recibir notificaciones cuando estos se alteran en la base de datos.
- Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas.

Laravel es un framework para aplicaciones web con una elegante sintaxis que ofrece componentes relativamente básicos para intentar disminuir determinada parte del desarrollo, facilitando las tareas más comunes que se utilizan en cierta medida para algunos proyectos web, además de la ligereza en su instalación mediante Composer. Cuenta con una arquitectura MVC (Model-View-Controller) que es una de las más utilizadas. Nos permite separar nuestra aplicación en tres capas: Modelo (la parte más baja del sistema, encargada de gestionar los datos), Controlador (el encargado de toda la lógica y procesamiento de la aplicación) y Vista (la parte visual e interactiva de la aplicación). Otra ventaja de Laravel, es que usa su sistema de templates Blade, lo que permite tener una mejor estructuración de las vistas, pudiendo extender vistas unas

Capítulo 1. Fundamentación teórica

de otras, y usar codeblocks (Laravel, 2019). Por lo antes mencionado sobre este framework Laravel, se puede afirmar que hace el entorno de trabajo web adecuado para la implementación de la solución propuesta.

1. 2. Análisis de soluciones existentes

Las fuentes de información consultadas permitieron identificar varias plataformas en la enseñanza universitaria dirigidas al aprendizaje de diversas disciplinas. Sin embargo, no se identificaron soluciones específicas encaminadas a la formación de habilidades investigativas para la escritura académica, la comunicación científica y la Ingeniería y Gestión de Software para el estudiantado universitario que transita por el proceso de realización del trabajo de diploma. En el contexto de la presente investigación, se tomaron en cuenta las plataformas que gestionan el aprendizaje y permiten desarrollar de manera específica materiales digitales, personalizar cursos en línea y desarrollar evaluaciones alternativas como apoyo a la enseñanza.

A continuación, se relacionan las principales plataformas utilizadas para gestionar el aprendizaje en los centros de educación cubanos:

CubaEduca

CubaEduca es una comunidad educativa en red, del Sistema Nacional de Educación. Constituye un espacio virtual, como plataforma integradora, que sirve de puerta a todo un conjunto de contenidos y servicios afines a la educación cubana. Es un canal de comunicación interactivo, sistémico y dinámico, gestionado por la Empresa de Informática y Medios Audiovisuales, CiNESOFT.

Esta plataforma cuenta además con un apartado dedicado a la familia, para estimular y ayudar al estudiante a través de varias herramientas. Atendiendo a las limitantes de conectividad, CubaEduca creó multimedias portables de contenidos offline con las herramientas, animaciones, juegos y el hiperentorno educativo disponible desde la web. Desde su concepción, CubaEduca está basada en la filosofía de web 2.0 o web social. No se trata solo de presentar contenidos, sino de recibir retroalimentación para los profesores que trabajan en el portal. Esta es una de las riquezas que posee, cuyo personal es de basta experiencia. Además cuenta con foros, actividades interactivas y aulas virtuales que forman parte del proceso de retroalimentación. (Guevara, 2017).

www.cubaeduca.cu

Capítulo 1. Fundamentación teórica

Repasador Virtual

Servicio digital que proporciona asistencia a todos los estudiantes cubanos, se realizó en el preuniversitario Aristides Viera, del municipio habanero de Playa. El nombrado Repasador Virtual, es una novedosa plataforma digital que contó con el apoyo en su realización, de expertos de la Empresa CiNESOFT, así como del Ministerio de Educación, entre otras instituciones cubanas. La novedosa iniciativa se integra al Portal Educativo cubano CubaEduca como un servicio web, que brinda asistencia y asesoría a estudiantes de diversas enseñanzas. Los servicios de la plataforma abarcan todas las asignaturas que se incluyen en los planes de estudio, desde la primera infancia hasta el bachillerato, e incluye la preparación para las pruebas de ingreso a la educación superior. (Radio Cubana, 2019)

Procesos de Uso e Implementación

Un equipo reúne a profesores con más de 25 años de experiencia vinculados al grupo organizativo del Ministerio de Educación. El gremio incluye a autores de los libros de texto y la mayoría colabora con el proyecto online CubaEduca y con CiNESOFT.

Esta plataforma se concibe como canal de consulta, asesoría y orientación curricular en línea para la comunidad estudiantil hasta el 12mo. grado, educadores en ejercicio y formación, y la familia. Por esta vía se responderán dudas sobre los contenidos de cada una de las asignaturas. El programa brinda además orientación y asesoría para las tareas docentes con apoyo de contenidos nacionales y el empleo de la información digital. Los recursos virtuales del repasador establecen una actividad de intercambio entre sus actores: usuario (estudiante, docente o familia), que plantea una duda, y el profesor en línea, que brinda la orientación. El sistema fluye en función de las demandas de los usuarios. Ayuda a identificar las necesidades educativas y busca hacer partícipe a la familia del ciclo enseñanza-aprendizaje. (Expósito, 2019)

Tecnología Utilizada

El sitio se creó como producto sobre la web, montado en un servidor con el uso del proyecto de software libre Symfony (framework PHP diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador). (Expósito, 2019)

<https://repasador.cubaeduca.cu/login>

CursAD

CursAD es el sitio formativo de los Joven Club de Computación y Electrónica, dirigido a la capacitación y preparación en contenidos relacionados con las tecnologías de la información y la

Capítulo 1. Fundamentación teórica

electrónica. Los cursos, foros y juegos instructivos ofertados en este sitio se encuentran soportados sobre la plataforma educativa Moodle.

<https://cursad.jovenclub.cu>

Aula GESTA

AulaGesta es una comunidad virtual de enseñanza y aprendizaje del Ministerio de Industrias MINDUS, donde se ofrecen servicios de formación profesional, con una visión armonizada a los procesos de actualización del Modelo Económico Cubano. Los servicios de educación en línea que se ofrecen están orientados a programas de formación de postgrados a directivos y sus reservas, utilizando modernos enfoques de gestión del aprendizaje.

<http://plataforma-educativa.gesta.cu/>

Aula Virtual CENED

El Aula Virtual CENED está soportada sobre la plataforma educativa Moodle, a través de la cual se puede acceder a variadas actividades de formación. La misma permite múltiples opciones, entre ellas la resolución por parte del estudiante, de las actividades planteadas por los profesores y la consulta a través de chat y foros sobre temas del curso, facilitando la construcción colaborativa del conocimiento.

<https://aulacened.uci.cu/>

EvaPostgrado

Entorno Virtual de Aprendizaje para estudios de Postgrado, es un sistema de formación a distancia desarrollado en la Universidad de Ciencias Informáticas con el objetivo de crear un espacio virtual para la formación de profesionales de la información y ramas relacionadas. Soportado en la plataforma Moodle, ofrece cursos de superación de posgrado, así como acciones formativas de posgrado académico conducentes a maestrías presenciales y a distancia, doctorados curriculares y colaborativos. (Egbert, et al., 2016)

evapostgrado.uci.cu

XAUCE ZERA 2.0

XAUCE ZERA 2.0 es una plataforma educativa desarrollada por el Centro de Tecnologías para la Formación (FORTES) de la UCI. Esta plataforma posee un diseño flexible capaz de guiar el proceso de enseñanza-aprendizaje en empresas e instituciones educativas a partir de un conjunto de herramientas centradas en los ambientes de aprendizaje colaborativo, que le dan

Capítulo 1. Fundamentación teórica

poder, tanto a la enseñanza como al aprendizaje. Posee una interfaz moderna y fácil de usar, diseñada para ser responsiva y accesible. Los educadores y los educandos pueden monitorear el progreso y el grado de finalización con un conjunto de opciones para monitoreo de actividades individuales o recursos.

<https://eva.uci.cu>

Las plataformas nacionales para la gestión del aprendizaje que han sido estudiadas, brindan servicios para la personalización de actividades formativas, diseños didácticos de los contenidos, así como facilidades para la elaboración de recursos de aprendizaje en diversas disciplinas o áreas de conocimiento. En tanto, actividades formativas encaminadas a la formación investigativa, son identificadas en la plataforma ZERA que entre sus cursos, se encuentra el de Metodología de la Investigación Científica que reciben los estudiantes del Tercer Año de Ingeniería en Ciencias Informáticas. Este curso posee una estructura ajustada a la planificación y control del proceso docente de esta asignatura y no concibe una estructura didáctica orientada a formar habilidades en el ámbito investigativo para realizar proyectos de investigación conducentes a trabajos de diploma.

A continuación, se referencian algunas plataformas internacionales para gestionar el aprendizaje:

QuizStar

QuizStar es una plataforma que incluye un creador de cuestionarios adaptado a los diferentes niveles educativos. Es adecuado para cualquier asignatura porque cuenta con distintas dinámicas de preguntas y respuestas y, además, ofrece contenido multimedia para complementar los exámenes.

<http://quizstar.4teachers.org/>

Socrative

Socreative posee un plan gratuito que permite incluir hasta 50 estudiantes en una clase y añadir preguntas en directo, que se corrigen de forma automática. Docentes y estudiantes tienen acceso a esta alternativa desde una aplicación instalada en sus dispositivos móviles o con una extensión de Chrome, siendo diferentes para cada uno de ellos. Las versiones de pago ofrecen más capacidad para estudiantes, mayor número de clases o poder exportar los datos de respuestas en Excel.

<https://www.socrative.com/>

Gnowledge

Capítulo 1. Fundamentación teórica

Knowledge destaca porque incluye estadísticas para medir el progreso académico del alumnado. Gracias a ella también es posible conocer quiénes han obtenido las mejores calificaciones en un mismo test o publicar los resultados en la red social de la propia plataforma.

[\(https://www.gnowledge.com/\)](https://www.gnowledge.com/)

Quibblo

Quibblo tiene una interfaz simple que facilita la creación de cualquier examen. Las preguntas se corrigen automáticamente proporcionando una calificación y una gráfica en la que se aprecia el rendimiento de la clase durante la prueba.

[\(https://www.quibblo.com/\)](https://www.quibblo.com/)

Gexcat

Gexcat permite preparar y corregir exámenes tipo test y generar también los de desarrollo de manera rápida. Su funcionamiento se basa en tres pasos: importar las preguntas a su base de datos; generarlas (hasta nueve modelos diferentes) en torno a las preguntas que se han incluido en la herramienta y, por último, corregirlas de manera automática. Con esta última opción se pueden conocer datos estadísticos de las preguntas que más aciertan o fallan los estudiantes.

www.gexcat.com

Plataforma virtual de Sistemas Tutores Inteligentes para la enseñanza de las matemáticas

Este sistema fue desarrollado para uso interactivo de los estudiantes de matemáticas, como una plataforma virtual con tutores inteligentes para el apoyo al proceso de aprendizaje. A partir del uso del sistema de tutoría inteligente, la plataforma genera enseñanza en tiempo real y por demanda de las necesidades individuales, con base en herramientas gratuitas como PHP, jQuery, Python y un Sistema Algebraico Computacional, también conocido por sus siglas en inglés CAS (Computer Algebra System). Permite hacer gráficas para manipularlas a través de botones, con el fin de ilustrar el concepto de la derivada, representar el lanzamiento de un proyectil para enseñanza o investigación o simular modelos físicos y matemáticos. A medida que el usuario va respondiendo correctamente, se van desplegando las preguntas que formula el tutor inteligente hasta llegar a la solución. Si el usuario no puede contestar, el tutor le hace sugerencias hasta determinado número de intentos. Mediante estas herramientas se realizan también exámenes en línea, a través de script gratuitos y controlando el tiempo (Manizales, 2014).

Valoración de los sistemas identificados

Capítulo 1. Fundamentación teórica

Una vez realizada la caracterización de las plataformas para la gestión del aprendizaje antes mencionadas, tanto en el ámbito nacional e internacional, se concluye lo siguiente:

- No todos los sistemas están diseñados o encaminados para el entorno universitario.
- Algunos de estos sistemas son privativos.
- No todas apoyan la política y los lineamientos de informatización del país sobre el uso del software libre para garantizar la soberanía tecnológica.
- Algunas requieren licencias.
- Algunos son sistemas de pago.
- La mayoría de los sistemas gestionan la información de los estudiantes de manera parcial.
- No todas son capaces de generar reportes.
- No todos estimulan al desarrollo del conocimiento dentro de la Escritura Académica, la Comunicación Científica o la Ingeniería y Gestión de Software.
- Se encuentra totalmente ausente el estímulo al desarrollo del análisis crítico de los estudiantes sobre el conocimiento de otros partiendo de una actividad coevaluativa y estando sustentada o dirigida a mejorar habilidades para la investigación científica, con enfoque directo a brindar mayor calidad a las tesis de pregrado dentro de la carrera de Ingeniería en Ciencias Informáticas.
- No todos se apoyan en el uso de materiales didáctico educativos para facilitar la calidad del aprendizaje.
- Se encuentra ausente el estímulo a la utilización de redes sociales académicas para la divulgación de resultados y la integración a una comunidad profesional.
- No todos se integran al análisis de resultados en base a los conocimientos adquiridos por los estudiantes hacia características claves para el desarrollo de un profesional en el ámbito de la Investigación Científica que involucre a profesores.

Cada uno de estos sistemas aportan un valor real y objetivo a necesidades puntuales de estudiantes y profesores para los procesos de enseñanza y aprendizaje, pues han partido de necesidades reales, las cuales proporcionaron la obtención de sus respectivas soluciones, pero por otra parte ninguna de estas herramientas aportan una solución global en cuanto a los aspectos antes analizados, sino que tratan de manera dispersa algunos de estos criterios de

Capítulo 1. Fundamentación teórica

forma individual, por lo que es válido afirmar la necesidad de una nueva propuesta de solución que generalice y englobe de forma práctica cada uno de estos criterios conformados que serán la base principal de la plataforma para contribuir a la formación de habilidades investigativas.

De acuerdo con los razonamientos que se han venido realizando, estas herramientas se descartan para sufragar las necesidades objetivas que sustentan esta investigación, no obstante, dada la experiencia moderada que poseen aportan pautas en el desarrollo a soluciones similares, tales como:

- La implementación de un sistema multiusuario con permisos favorece al control, acceso y modificación de la información.
- La generación de listados y reportes permite el ahorro de tiempo y contribuye a la toma de decisiones.
- El desarrollo de la aplicación sobre plataforma web permite mayor flexibilidad a los usuarios, al poder ser capaz de gestionar el aprendizaje.
- El diseño de una arquitectura escalable, ofrece mejor respuesta de adaptación a posibles cambios en el negocio.

1. 3. Lenguajes, herramientas, metodologías y tecnologías utilizadas

1.4.1. Metodologías de desarrollo de software

Una metodología de desarrollo de software se refiere a un framework (entorno o marco de trabajo) que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. Las metodologías de desarrollo de software tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistemas. Las metodologías de desarrollo se pueden enmarcar en dos grandes grupos, las metodologías tradicionales y las metodologías ágiles. Las tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán; son recomendadas para los proyectos de grandes dimensiones y con grandes equipos de desarrollo. Las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas, respondiendo a los cambios que puedan surgir a lo largo del desarrollo del proyecto. La selección depende de qué producto

Capítulo 1. Fundamentación teórica

se desee desarrollar, las dimensiones que tendrá el proyecto, el tiempo que se disponga y el equipo de trabajo, entre otros factores (Hernández, 2020).

La metodología de desarrollo AUP-UCI tiene como objetivo aumentar la calidad del software que se produce. Este modelo constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora de software. Estas prácticas se centran en el desarrollo de productos y servicios de calidad. En el caso de la variación de la metodología AUP, definida para la actividad productiva de la UCI, propone para el ciclo de vida de los proyectos, entre sus especificaciones, las fases: Inicio, Ejecución y Cierre). Propone 7 disciplinas: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación (Hernández, 2020). Esta metodología propone 4 escenarios posibles en los proyectos de desarrollo de software (Universidad de las Ciencias Informáticas, 2019):

- Escenario 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los casos de uso del negocio muestran como los procesos son llevados a cabo por personas y los activos de la organización. Modelan el negocio con casos de uso del negocio (CUN), solo pueden modelar el sistema con casos de uso del sistema (CUS).
- Escenario 2: 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. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información. Modelan el negocio con modelo conceptual (MC) solo pueden modelar el sistema con casos de uso del sistema (CUS).
- Escenario 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados. Modelan el negocio con descripción de proceso de negocio (DPN) solo pueden modelar el sistema con descripción de requisitos de procesos (DRP).

Capítulo 1. Fundamentación teórica

- Escenario 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una historia de usuario no debe poseer demasiada información. Proyectos que no modelen negocio, solo pueden modelar el sistema con historias de usuario (HU).

Teniendo en cuenta las características del presente proyecto de informatización para la gestión de la formación de habilidades investigativas, el escenario adoptado para su desarrollo fue el **Escenario 2**.

1.4.2. Modelado de software

Para el desarrollo de La Plataforma para la Formación de Habilidades Investigativas se utilizaron herramientas, tecnologías, lenguajes, arquitectura y metodología que guían el desarrollo de software. En este epígrafe se describe cada una de ellas y se indica la función que realizan dentro del sistema.

Lenguaje Unificado de Modelación UML

El lenguaje unificado de modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML es un lenguaje de representación visual que permite combinar diversos elementos gráficos y crear diagramas. Se usa para modelar sistemas y usa tecnología orientada a objetos. El lenguaje unificado de modelado describe lo que hará un sistema, pero no dice cómo implementarlo. Su objetivo es visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo. Involucra todo el ciclo de vida del proyecto y está pensado para varios lenguajes y plataformas. Es aplicable en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso utilizar (Ferré Grau, y otros, 2018).

Visual Paradigm

Visual Paradigm, versión 8.0, es una herramienta UML CASE (Computer Aided Software Engineering, por sus siglas en inglés) aplicable en todo el ciclo de vida del desarrollo de software, que admite UML (Unified Modeling Language, por sus siglas en inglés), BPMN (Business Process Modeling Notation, por sus siglas en inglés), SysML y notación de modelado de procesos empresariales del grupo de gestión de objetos, entre otras tecnologías. Además del soporte de

Capítulo 1. Fundamentación teórica

modelado, proporciona capacidades de generación de informes e ingeniería de código, incluida la generación de código y dibujar todos los tipos de diagramas de clases. Es una herramienta multiplataforma de diseño y gestión fácil de usar para sistemas de tecnología informática. La herramienta proporciona abundantes tutoriales UML, demostraciones interactivas de UML y proyectos UML. Es fácil de instalar y actualizar y compatible entre ediciones (Visual Paradigm, 2018).

1.4.3. Herramienta para control de versiones

Git

Versión 2.17 .Es un sistema de control de versiones distribuido de código abierto y gratuito, diseñado para manejar tanto proyectos pequeños como proyectos muy grandes, con rapidez y eficiencia. Es fácil de aprender y con un rendimiento rápido. Supera a las herramientas de SCM (Source Code Management, Gestión de Código Fuente) como Subversión, CVS, Perforce y ClearCase con características como bifurcaciones locales baratas, áreas de preparación convenientes y múltiples flujos de trabajo.

1.4.4. Lenguaje de programación

HTML

HTML (HyperText Markup Language, Lenguaje de Marcado de Hipertexto), versión 5, es un lenguaje descriptivo que especifica la estructura de una página web, documento de texto plano estructurado con elementos rodeados por etiquetas de apertura y cierre coincidentes. Cada etiqueta comienza y termina con corchetes angulares (<>), algunas de ellas son vacías, es decir, no pueden encerrar ningún texto, por ejemplo . Las etiquetas HTML se pueden extender con atributos, los que proporcionan información adicional que afecta la forma en que el navegador interpreta el elemento. Un archivo HTML puede ser procesado por cualquier navegador web.

CSS

CSS (Cascading Style Sheets, por sus siglas en inglés), versión 2, es un lenguaje de hojas de estilos en cascada creado para controlar la presentación de los documentos electrónicos definidos con XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas. La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además,

Capítulo 1. Fundamentación teórica

mejora la accesibilidad del documento y reduce la complejidad de su mantenimiento. Si el lenguaje XHTML se utiliza para designar lo que es un párrafo o lo que es un titular, el lenguaje CSS se utiliza para definir su aspecto, es decir, el color, tamaño y tipo de letra del texto y la separación entre titulares y párrafos (Eguíliz Pérez, 2008).

JavaScript

Java Script , versión 1.8.5, es un lenguaje de programación ligero interpretado o compilado en JIT (Just In Time, justo a tiempo) con funciones de primera clase. Es conocido como el lenguaje de scripting para páginas web, pero también es utilizado en el servidor. Además, es un lenguaje dinámico basado en prototipos, multi-paradigma y soporta estilos orientados a objetos, imperativos y declarativos (por ejemplo, programación funcional).

PHP

PHP: Hypertext Preprocessor (Preprocesador de Hipertexto PHP), versión 7.3.3, es un lenguaje de código abierto interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor web. Este no se encuentra limitado a resultados en HTML, también, puede presentar otros resultados, como XHTML y archivos XML. Asimismo, puede auto-generar estos archivos y almacenarlos en el sistema en lugar de presentarlos en la pantalla. La característica más potente y destacable es su soporte para una gran cantidad de base de datos, por lo que escribir una interfaz vía web para acceder a diferentes sistemas gestores de bases de datos, es una tarea viable, efectiva y rápida con dicho lenguaje.

SQL

Structured Query Language (Lenguaje de consultas estructurado), es un lenguaje de programación diseñado para administrar datos almacenados en bases de datos relacionales. Opera a través de declaraciones simples declarativas. Esto mantiene los datos precisos y seguros, y ayuda a mantener la integridad de las bases de datos, independientemente del tamaño.

1.4.5. Sistema Gestor de Base Datos

MySQL

MySQL, versión 8.0, ofrece un servidor de base de datos SQL (lenguaje de consulta estructurado) muy rápido, multiproceso, multiusuario y robusto. (MySQLOficial, 2020). El uso conjunto del lenguaje PHP y el sistema gestor de bases de datos MySQL permite la construcción, de una manera sencilla y eficiente verdaderos sitios webs dinámicos. (Cobo, 2005)

Capítulo 1. Fundamentación teórica

MySQL Workbench

MySQL Workbench, versión 8.0, este software da al usuario la posibilidad de modelado de datos, desarrollo de SQL y herramientas de administración completas para la configuración de servidores, administración de usuarios, copias de seguridad y mucho más. MySQL Workbench permite diseñar visualmente, modelar, generar y administrar bases de datos. Incluye todo lo que necesita un modelador de datos para crear modelos complejos de ingeniería de requisitos, ingeniería directa e inversa y también ofrece características clave para realizar tareas difíciles de gestión de cambios y documentación que normalmente requieren mucho tiempo y esfuerzo.

1.4.6. Servidor Web

Un servidor web es el encargado de manejar páginas web y enviarlas a través de la red a quienes lo requieran y tengan los permisos para dichas páginas. Son los principales encargados de generar tráfico en Internet; toda página web está almacenada en uno de estos servidores, los cuales en su mayoría pertenecen a empresas de hosting que arriendan sus servicios para que los clientes almacenen sus páginas web en ellos, teniendo acceso a los mismos durante las veinticuatro horas del día. Para cuestiones técnicas, el servicio web no es más que un software con procesos en espera de que algún usuario conectado a él solicite el contenido que almacene, al llegar una solicitud, el servidor web, que en este caso se hace uso de apache, genera los procesos propios para entregar el contenido por medio de protocolo http, al solicitante según los permisos que este tenga sobre los archivos, por lo cual si el sitio que es solicitado solo está disponible para usuarios que se identifiquen ante el sistema, quien no esté en la lista de esos usuarios, no podrán alcanzar el contenido solicitado. Cabe destacar que por lo general el software más utilizado por los centros de datos para esta tarea es apache, pues tiene la característica de poseer un usuario propio dentro del sistema, el cual no puede ser alcanzado por otro, ningún usuario puede identificarse como apache, creando un usuario virtual independiente dando más seguridad a sus operaciones y trabajo dentro del servidor (Gisela, 2019).

Apache

Versión 3.2.3. Apache es un software especializado en ofrecer servicios de servidor web. Es versátil, ligero y muy útil, además de ser completamente gratuito y de código abierto. Su popularidad es tal que, actualmente, cerca del 50% de las páginas web de todo el mundo se ejecutan en un servidor de este tipo. Aunque se le conoce así, su nombre completo es Apache HTTP Server, y sus responsables tienen también un nombre similar: Apache Software

Capítulo 1. Fundamentación teórica

Foundation. Lleva en activo desde el año 1995, tiempo más que suficiente para erigirse como el estándar que es en la actualidad. Fiable, robusto y muy flexible, permite al dueño de cualquier web publicar el contenido que desea en esta, como también gestionar todos sus ficheros de forma fácil y sencilla. Es una comunidad de usuarios la que sigue ofreciendo soporte para él y mejorándolo en todo lo posible. Actualmente se utiliza en plataformas Unix, Windows y Macintosh, de ahí que esté presente en la gran mayoría de páginas web de todo el mundo (Morelos, 2020).

1.4.7. Framework de desarrollo

Laravel

Version 6. El framework Laravel trabaja con una arquitectura de carpetas avanzada, de modo que promueve la separación de los archivos con un orden correcto y definido, que guiará a todos los integrantes del equipo de trabajo y será un estándar a lo largo de los distintos proyectos. Por supuesto, dispone también de una arquitectura de clases también muy adecuada, que promueve la separación del código por responsabilidades (Laravel, 2019).

Artisan

La consola es una herramienta potente para el desarrollador, con Artisan, Laravel abre una cantidad de opciones de comandos en consola para ejecutar diferentes tareas, este potente sistema de comandos de consola permite resumir muchas tareas tediosas y repetitivas. Artisan además ofrece la posibilidad de desarrollar comandos propios para ser utilizados en consola. Esta herramienta ha conseguido que la experiencia de desarrollo con Laravel sea muy atractiva para las personas que se han decidido por este framework (Laravel, 2019).

1.4.8. Otras herramientas utilizadas para el desarrollo de la plataforma

XAMPP Control Panel

Versión 3.2.3. Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los interpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, PERL.

Composer

Versión 1.9.1. Laravel depende de un número de paquetes externos para funcionar correctamente, así que la mejor forma de administrar los paquetes que necesita Laravel o esos paquetes extras que podemos integrar a nuestros desarrollos es utilizar Composer que será

Capítulo 1. Fundamentación teórica

administrador de dependencias. Realiza tareas manualmente engorrosas de forma automática y centralizada.

Axure RP

Versión 8. Es una herramienta muy completa, orientada a diseñar wireframes y prototipos básicos o avanzados de forma fácil, dirigido a aplicaciones web y de escritorio. Ofrece capacidades que se encuentran típicamente en las herramientas de creación de diagramas, como la colocación de arrastrar y soltar, el cambio de tamaño y formato de los widgets. Además, cuenta con funciones para anotar los widgets y la definición de las interacciones como la vinculación, la vinculación condicional, simulando controles de pestañas, mostrar / ocultar elementos.

El programa tiene por defecto una amplia gama de widgets con los cuales se pueden diseñar fácil y rápidamente interfaces para proyectos de sitios web o de aplicaciones móviles. Igualmente el arquitecto de información puede crear otros widgets personalizados y reutilizables, los cuales pueden simular funcionalidades o módulos que se pueden necesitar para describir mejor un proceso, interacción o contenido. Además, gracias a que la comunidad de diseñadores UX y arquitectos de información que usan Axure cada vez es más grande, se pueden encontrar en internet diversas librerías de gadgets, módulos, íconos, grillas y demás elementos de diseño que le pueden aportar muchos elementos al proyecto a desarrollar.

Firefox Developer Edition

Versión 2019.1 Firefox Developer Edition es una versión de Firefox adaptada para desarrolladores, ofreciendo características y herramientas de desarrollo experimental. Permite examinar, explorar y depurar sitios y aplicaciones web, así como diagnosticar problemas de rendimiento. A continuación, se presentan algunos de sus componentes:

- Consola web: posibilita visualizar mensajes de registro (logs) en una página web e interactuar con la página usando JavaScript.
- Inspector de página: permite revisar y modificar el HTML y CSS de la página.
- Depurador JavaScript: permite paso a paso, examinar y modificar el JavaScript que está ejecutándose en una página.
- Monitor de red: muestra las solicitudes de red hechas cuando una página está cargada.
- Herramienta de rendimiento: analiza la capacidad de respuesta del sitio web, del JavaScript y el rendimiento del diseño.

Capítulo 1. Fundamentación teórica

- Desempeño de red: muestra cuánto tardan en cargar las partes del sitio web.

PhpStorm

PhpStorm es un editor para PHP, HTML, CSS y JavaScript, desarrollado por JetBrains. PhpStorm es un IDE de PHP ligero y inteligente centrado en la productividad del desarrollador que entiende profundamente su código. El editor realmente 'obtiene' su código y entiende profundamente su estructura, soportando todas las características del lenguaje PHP para proyectos modernos y legados. Proporciona mejor terminación de código, prevención de errores en la marcha y más. Realiza muchas tareas de rutina directamente desde el IDE, gracias a la integración de Sistemas de Control de Versiones, soporte para implementación remota, bases de datos / SQL, herramientas de línea de comandos, Vagrant, Composer, REST Client y muchas otras herramientas. Cientos de inspecciones se encargan de verificar su código mientras escribe, analizando todo el proyecto.

Estas herramientas, poseen utilidades objetivas para el desarrollo e implementación de cualquier producto de software, las mismas desempeñan en este caso un papel fundamental para la producción del sistema en cuestión; MySQL proporciona el servicio de las bases de datos, que posibilita ser administrado gracias a la herramienta MySQLWorkbench, que es precisamente un herramienta especializada para la gestión del servicio de bases de datos; VisualParadigm ha permitido representar de forma visual toda la modelación y diseño en el entorno de los diferentes procesos del negocio; Git, permite un eficaz control y manejo de versiones para las etapas de desarrollo del sistema; el Xampp es primordial para gestionar y proveer el servicio Web de Apache; Axure ha recaído su utilidad en la maquetación y prototipado en cuanto al diseño del sistema. Se ha utilizado Firefox Developer Edition como navegador principal para visualizar la interactividad del diseño, y además por poseer particularidades que lo destacan del resto de navegadores en cuanto a la comodidad de su uso para el desarrollo. PhpStorm por su parte ha propiciado el ambiente idóneo, por ser un IDE especializado y dedicado para entornos de trabajo en Php, muy conjugado con el propio uso de la tecnología Laravel, que también se sustenta por el lenguaje Php.

1. 4. Conclusiones del capítulo

A partir de todo lo anteriormente planteado se pueden arribar a las siguientes conclusiones:

- Se implementó una base teórica relacionada con los ambientes virtuales de enseñanza y aprendizaje, posibilitó la adquisición de conocimientos sobre las diferentes herramientas encaminadas a la gestión del contenido y el aprendizaje, se clasificaron cada una de ellas en

Capítulo 1. Fundamentación teórica

privativas o gratuitas, a partir además de su despliegue y las necesidades que cubren tanto para el panorama nacional como internacional.

- Se caracterizó el marco de trabajo, metodología, lenguaje de programación, herramientas y tecnología de desarrollo, que permitieron formar las bases propicias para la futura implementación de la Plataforma de Formación de Habilidades Investigativas.

Capítulo 2. Descripción de la propuesta

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA.

El presente capítulo tiene como objetivo describir la propuesta de Plataforma para la Formación de Habilidades Investigativas en los estudiantes universitarios. A partir de ello se extraen las características que deben cumplir los módulos a desarrollar y se listan los requerimientos funcionales y no funcionales, mostrándose una breve caracterización de cada uno. También se muestra el diseño de la base de datos, se describe el patrón arquitectónico y los patrones de diseño utilizados para esta Plataforma.

2. 1. Características de la propuesta de solución

Para dar respuesta a la problemática planteada se propone la realización de un sistema interactivo de evaluación que será integrado a la Universidad de Ciencias Informáticas para aportar al proceso de enseñanza y aprendizaje no presencial, una herramienta que contribuya a la formación de habilidades investigativas durante el proceso de realización de los trabajos de diploma en los estudiantes de Ingeniería en Ciencias Informáticas, estimulando el aprendizaje de forma autodidacta, para así contribuir a la eficiencia y eficacia de la elaboración y presentación de los trabajos de diplomas en el estudiantado. En este ambiente participan dos actores fundamentales siendo estos estudiantes y profesores que interactuarán especialmente en la plataforma para dar cumplimiento a estos objetivos. El entorno web estará estructurado por tres módulos principales con los cuales estarán interactuando los estudiantes para su proceso de evaluación.

El Módulo Escritura Académica está intencionado para que el estudiante desarrolle sus habilidades lingüísticas en el ámbito académico, en especial, la producción y comprensión de textos escritos, desarrollar un conjunto de rasgos comunes de vocabulario, sintaxis y gramática que configuren un estilo propio cuyas cualidades principales sean la precisión, la concisión, la claridad y una tendencia a la neutralidad e impersonalidad. Este módulo contará con 8 unidades de contenido en donde cada una va a ser contenedora de tres estructuras de contenidos: un Pre-Test el cual pondría a prueba el dominio actual del estudiante sobre el tema induciéndolo a ofrecer su propuesta y evaluar a otro estudiante en ese aspecto; un apartado para Contenido Informativo que tendría como finalidad principal informar al estudiante sobre la temática abordada y por último un Post-Test que nuevamente evaluaría al estudiante para validar el cumplimiento de su objetivo que es solidificar el nuevo conocimiento.

El Módulo Comunicación Científica estaría encaminado a contribuir al eficaz desempeño en el proceso de presentación, transmisión e intercambio de información científica. Este contaría con

Comentado [w1]: No debes hablar de sistema pq todavia no teins un sistema

Capítulo 2. Descripción de la propuesta

tres unidades de contenido que aborden temas desde comunicar resultados en eventos hasta el tratamiento de redes sociales para la investigación. Estaría en consonancia con la misma estructura evaluativa expuesta en el módulo anterior.

El Módulo Ingeniería y Gestión de Software es la última unidad con la cual el estudiante interactuaría, tiene como finalidad ofrecer una noción general de los principios y metodologías para el desarrollo y mantenimiento de un software abarcando lo concerniente al modelado de datos, la extracción de requisitos y la gestión de software. Esta unidad se entornará en el mismo modelo de evolución antes mencionado.

Para cada módulo de los anteriormente mencionados se desplegará una página de contenidos que brinda la posibilidad de mantener actualizado cada módulo, incluyendo textos, imágenes, enlaces, videos, etc. Para los estudiantes, esta página de contenidos aporta los conocimientos que han sido diseñados de manera específica para cada unidad temática. La presencia de material audiovisual combinado con las actividades de aprendizaje hace que el contenido sea motivador y se produzcan interacciones en la plataforma.

El estudiante deberá realizar una tarea que está directamente relacionada con el contenido previamente consultado. Una vez elaborada la tarea, el sistema dispondrá de un enlace para subir la evidencia conformada por el estudiante y proceder a enviarla. Seguidamente el estudiante debe evaluar una propuesta aportada por uno de sus compañeros (coevaluación), totalmente anónima, otorgando una valoración según la escala aportada. Finalmente, el estudiante emite una autovaloración sobre el aprendizaje que ha experimentado en esta unidad temática.

El profesor evaluará, a partir de una escala preestablecida de cumplimiento de criterios, el desempeño de los estudiantes dentro de cada módulo para cada unidad de contenido. Además, el profesor, será el usuario con privilegios de gestionar las valoraciones percibidas por los estudiantes.

La interfaz referida a la evaluación del profesor permitirá hacer un seguimiento de los estudiantes, organizados por grupos docentes. Desde esta vista es posible conocer las evaluaciones recibidas y otorgadas de cada estudiante, en cada unidad temática, se valoran las evidencias aportadas por los estudiantes y el profesor emite una valoración individual del aprendizaje demostrado por el estudiante a partir de su desempeño en la plataforma, esta valoración se realiza a partir de la escala valorativa disponible en cada caso.

Capítulo 2. Descripción de la propuesta

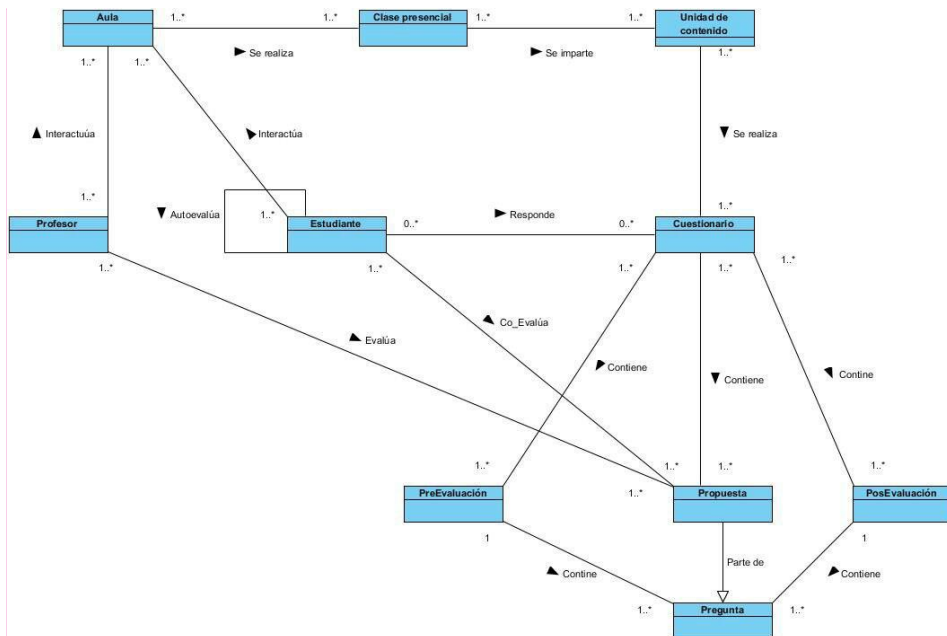
La solución será desarrollada teniendo en cuenta las reglas de negocio definidas, se usarán tecnologías actualizadas, de código abierto y libres de licencia.

2. 2. Modelo Conceptual

Un modelo conceptual es una representación de un sistema, hecho de la composición de conceptos que se utilizan para ayudar a las personas a conocer, comprender o simular un tema que representa el modelo. También es un conjunto de conceptos. Es un modelo visual de un sistema que ilustra las interconexiones de los componentes del modelo. Tiene como objetivo identificar y explicar los conceptos significativos del dominio de un problema. Es un artefacto útil que permite obtener conocimiento acerca de cómo se desenvuelve el problema en el contexto real, asociando para ello los conceptos que en la problemática se tratan. También se le denomina modelos conceptuales, modelos de objetos del dominio y modelos de objetos de análisis. Es el artefacto que se crea durante el análisis orientado a objetos. Se centra en las abstracciones relevantes, vocabulario del dominio e información del dominio. (Peñalvo, 2018).

Para la realización de un correcto levantamiento de requisitos se realiza el modelo conceptual, detallando cada uno de los conceptos asociados al entorno del problema y las relaciones entre ellos.

Capítulo 2. Descripción de la propuesta



Comentado [w2]: No se ve, uso excivo de mayúsculas

Figura 1 Modelo conceptual.

En el modelo conceptual se pueden apreciar las relaciones existentes. Concretamente el Profesor y el Estudiante como principales actores interactúan en la Clase presencial donde se imparte la Unidad de Conetenido. El Cuestionario estructurado por preguntas, para cada Unidad de Contenido es respondido por el Estudiante, basado en los resultados del Pre_Test, mientras que la Propuesta consite en entregar una evidencia conformada por el Estudiante que luego es evaluada por el Profesor. Seguidamente el Estudiante debe evaluar una Propuesta aportada por uno de sus compañeros (coevaluación), totalmente anónima, luego el Estudiante emite una autovaloración sobre el aprendizaje que ha experimentado en dicha Unidad de Contenido. Finalmente en el Pos_Test se recogen las respuestas a las preguntas para un análisis comparativo de conocimientos.

2. 3. Especificación de requisitos de software

Recopilar requisitos es el proceso de determinar, documentar y gestionar las necesidades y los requisitos de los interesados para cumplir con los objetivos del proyecto. El beneficio clave de

Capítulo 2. Descripción de la propuesta

este proceso es que proporciona la base para definir y gestionar el alcance del proyecto, incluyendo el alcance del producto (PMBOK, 2013).

El éxito del proyecto depende directamente de la participación activa de los interesados en el descubrimiento y la descomposición de las necesidades en requisitos, y del cuidado que se tenga al determinar, documentar y gestionar los requisitos del producto, servicio o resultado del proyecto. Los requisitos incluyen condiciones o capacidades que el proyecto debe cumplir o que deben estar presentes en el producto, servicio o resultado para satisfacer un acuerdo u otra especificación formalmente impuesta. Los requisitos incluyen las necesidades y expectativas cuantificadas y documentadas del patrocinador, del cliente y de otros interesados (PMBOK, 2013).

En cualquier proyecto software (Agut, 2001) los requisitos son las necesidades del producto que se debe desarrollar. Por ello, en la fase de análisis de requisitos se deben identificar claramente estas necesidades y documentarlas. Como resultado de esta fase se debe producir un documento de especificación de requisitos en el que se describa lo que el futuro sistema debe hacer. Por tanto, no se trata simplemente de una actividad de análisis, sino también de síntesis. Asimismo, se define requisito como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado (Piattini, 1996). Esta definición se extiende y se aplica a las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación.

Objetivos de la Especificación de Requisitos de Software:

Los principales objetivos que se identifican en la especificación de requisitos software son (Chalmeta, 2000):

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software: El cliente debe participar activamente en la especificación de requisitos, ya que éste tiene una visión mucho más detallada de los procesos que se llevan a cabo. Asimismo, el cliente se siente participe del propio desarrollo.
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente: En muchas ocasiones el cliente no sabe exactamente qué es lo que quiere. La especificación de requisitos de software (ERS) permite al cliente definir todos los requisitos que desea y al mismo tiempo los desarrolladores tienen una base fija en la que trabajar. Si no se realiza una buena especificación de requisitos, los costes de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación.

Capítulo 2. Descripción de la propuesta

- Servir de base para desarrollos de estándares de ERS particulares para cada organización: Cada entidad puede desarrollar sus propios estándares para definir sus necesidades.

Una buena especificación de requisitos software ofrece una serie de ventajas entre las que destacan el contrato entre cliente y desarrolladores, la reducción del esfuerzo en el desarrollo, una buena base para la estimación de costes y planificación, un punto de referencia para procesos de verificación y validación, y una base para la identificación de posibles mejoras en los procesos analizados (Agut, 2001).

Para el desarrollo de la propuesta de solución se definieron requisitos funcionales y no funcionales. Para la captura de los funcionales se realizaron una serie de técnicas que facilitan la interacción y comunicación cliente - desarrollador.

Definición de las técnicas de obtención de requisitos:

El proceso de obtención de requisitos, cuyo objetivo es definir las características o particularidades que debe llegar a cumplir el sistema no es solo un proceso estrictamente técnico, sino que también está estrechamente vinculado con el aspecto social, la interacción interpersonal, en este sentido las habilidades de comunicación deben estar desarrolladas para una eficaz captura de estos requisitos. Para la identificación de los requisitos de la propuesta de solución se utilizaron las técnicas mencionadas a continuación:

- Entrevista: Esta técnica permitió analizar en detalle las necesidades del cliente en su ambiente de trabajo, posibilitó el estudio de las expectativas del mismo respecto a la propuesta de solución, se realizó una serie de preguntas y se definieron las motivaciones y aspiraciones que lo impulsan para que se lleve a cabo el desarrollo de la solución.
- Tormenta de ideas: Esta técnica es una de las sesiones de trabajo que ha sido orientada a obtener la mayor cantidad de ideas posibles con el objetivo de luego ser evaluadas.

2.4.1 Requisitos funcionales del sistema

Los requerimientos funcionales de un sistema son capacidades o condiciones que el sistema debe cumplir, de cómo debería reaccionar el mismo a entradas particulares y de cómo debería comportarse en situaciones específicas. Además, surgen de la razón fundamental de la existencia de un producto. (Sommerville, 2011)

En algunos casos, los requisitos funcionales también, pueden indicar explícitamente lo que el sistema no debe hacer, por lo que puede ser también una declaración negativa. Luego del

Capítulo 2. Descripción de la propuesta

análisis del negocio y las actividades a informatizar identificadas se pueden definir los siguientes requisitos funcionales:

Tabla 3 Requisitos Funcionales del Sistema.

No.	Requisito	Descripción	Complejidad	Prioridad
RF1	Autenticar Usuario	El sistema debe permitir que un usuario se pueda autenticar.	Media	Alta
RF2	Crear Usuario	El sistema debe permitir insertar usuario.	Media	Alta
RF3	Listar Usuario	El sistema debe permitir mostrar un listado de usuarios con permisos dentro del sistema.	Media	Media
RF4	Modificar Usuario	El sistema debe permitir modificar usuario.	Media	Media
RF5	Eliminar Usuario	El sistema debe permitir eliminar usuario.	Media	Media
RF6	Crear Preguntas	El sistema debe permitir crear preguntas.	Media	Alta
RF7	Modificar Preguntas	El sistema debe permitir modificar preguntas.	Media	Alta
RF8	Eliminar Preguntas	El sistema debe permitir eliminar preguntas.	Media	Media
RF9	Mostrar Preguntas	El sistema debe permitir mostrar preguntas.	Media	Media
RF10	Insertar Respuestas Preguntas	El sistema debe permitir insertar las repuestas de los estudiantes a las preguntas.	Media	Alta
RF11	Asignar permisos	El sistema debe permitir asignar permisos.	Media	Alta
RF12	Registrar Evaluaciones	El sistema debe permitir registrar las evaluaciones del profesor a las respuestas dadas por los estudiantes y las evaluaciones dadas por los estudiantes a las propuestas de otros estudiantes.	Media	Alta
RF13	Insertar Contenido Informativo	El sistema debe permitir insertar contenido informativo de la unidad de contenido (sub temática) en particular, que incluye recursos educativos, tutoriales, documentos informativos o imágenes.	Media	Alta
RF14	Modificar Contenido Informativo	El sistema debe permitir modificar contenido.	Media	Media
RF15	Mostar Contenido Informativo	El sistema debe permitir mostrar contenido.	Media	Alta
RF16	Eliminar Contenido Informativo	El sistema debe permitir eliminar contenido.	Media	Media

Capítulo 2. Descripción de la propuesta

RF17	Insertar rol	El sistema debe permitir Insertar un rol a cada usuario.	Media	Alta
RF18	Listar roles	El sistema debe permitir mostrar una lista de roles de usuario.	Media	Media
RF19	Modificar rol	El sistema debe permitir modificar los roles.	Media	Media
RF20	Eliminar rol	El sistema debe permitir eliminar roles.	Media	Media
RF21	Insertar grupo	El sistema debe ser capaz de insertar grupo.	Media	Alta
RF22	Mostrar grupo	El sistema debe ser capaz de mostrar grupo.	Media	Alta
RF23	Modificar grupo	El sistema debe ser capaz de modificar grupo.	Media	Media
RF24	Eliminar grupo	El sistema debe ser capaz de eliminar grupo.	Media	Media
RF25	Generar Reportes	El sistema debe permitir realizar un reporte estadístico brindando un listado de la información seleccionada.	Media	Media

2.4.2 Requerimientos no funcionales del sistema

Los requisitos no funcionales son restricciones en los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, en el proceso de desarrollo e impuestas por las normas. Definen propiedades emergentes del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad. Pueden especificar también la utilización de una herramienta CASE en particular, un lenguaje de programación o un método del desarrollo. Estos requisitos se aplican al sistema en su totalidad, en lugar de a las funciones o servicios individuales (Laguna, 2018).

Clasificación de los requisitos no funcionales:

Requisitos del producto:

Especifican el comportamiento del producto obtenido: velocidad de ejecución, memoria requerida, porcentaje de fallos aceptables.

Requisitos organizacionales:

Son una consecuencia de las políticas y procedimientos existentes en la organización: procesos estándar utilizados, de fechas de entrega, documentación a entregar.

Requisitos externos:

Presentan factores externos al sistema y a su proceso de desarrollo: interoperabilidad del sistema con otros, requisitos legales, éticos.

Capítulo 2. Descripción de la propuesta

La solución propuesta cuenta con los siguientes requisitos no funcionales:

Usabilidad:

Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder mostrando en la vista mediante íconos y menús el acceso a la misma.

Los términos visibles para el usuario en la aplicación serán los utilizados por los usuarios en la rama abordada con vista a una mayor comprensión por parte de los mismos de la herramienta de trabajo.

Las vistas del sistema deben indicar en cada momento la acción que se está realizando, así como los íconos deben estar representados por una imagen acorde a la acción que se realiza mediante el mismo.

Confiabilidad:

El sistema debe ser tolerante a fallos y debe mostrar la información necesaria para poder mantener orientado al usuario en base a la ocurrencia de errores.

El sistema de autenticación recogerá los datos necesarios de los usuarios y tendrá control de identidades.

Eficiencia:

El sistema deberá soportar la interacción concurrente de los usuarios al mismo de forma estable.

El sistema deberá tener por cada transacción un tiempo de respuesta promedio de 2 segundos y un máximo de 4 segundos.

Soporte:

El sistema se registrará por un estándar de código debidamente documentado en el expediente de proyecto.

Todos los productos de trabajo generados en el desarrollo del software se registrarán por unas pautas de configuración debidamente documentadas en el expediente de proyecto.

Seguridad:

El acceso a los datos estará restringido por usuario y contraseña, garantizando la disponibilidad de la información a los usuarios autorizados, mediante sus respectivos roles y permisos.

Interfaz:

Capítulo 2. Descripción de la propuesta

La comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTPS.

El sistema se integrará con el directorio activo de la institución cliente mediante el protocolo LDAP.

La interfaz de usuario se guiará por la vista de arquitectura de presentación.

Portabilidad:

La propuesta de desarrollo debe ser capaz de ejecutarse en los navegadores empleados en la UCI tales como Safari, Chrome, Mozilla Firefox, Ópera.

2. 4. Diagrama de Casos de Uso del Sistema

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). Un diagrama de casos de uso consta de actores, casos de uso, relaciones de uso, herencia y comunicación (Universidad Nacional Abierta y a Distancia, 2016).

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

Los diagramas de casos muestran la relación entre los actores y los casos de uso en un sistema.

Capítulo 2. Descripción de la propuesta

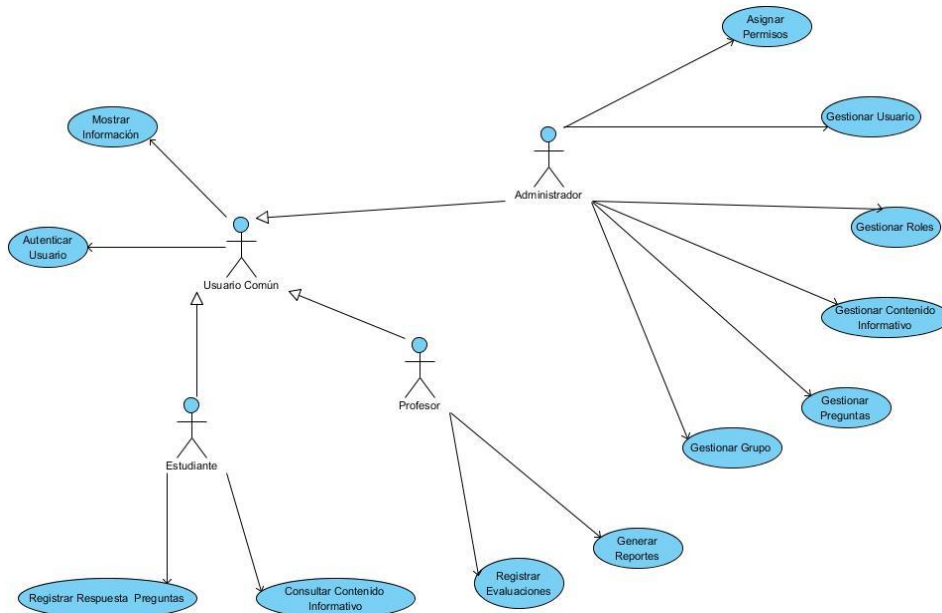


Figura 2 Diagrama de Casos de Uso del Sistema.

2. 5. Diseño de la propuesta de solución

Arquitectura del software: La arquitectura de software se refiere a la estructura general de éste y a las formas en las que brinda integridad conceptual a un sistema. Contiene conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software. Una correcta definición de la arquitectura permite facilitar el diseño y construcción de los componentes y sus relaciones. En su forma más sencilla, es la estructura de organización de los componentes de un programa (módulos), la forma en la que éstos interactúan y la estructura de datos que utilizan (Muñoz , y otros, 2013).

2. 6. Patrón Arquitectónico:

Un patrón arquitectónico se puede considerar como una descripción abstracta estilizada de buena práctica, que se ensayó y puso a prueba en diferentes sistemas y entornos. Debe describir una organización de sistema que ha tenido éxito en sistemas previos. Debe incluir información sobre cuándo es y cuándo no es adecuado usar dicho patrón, así como sobre las fortalezas y debilidades del patrón. (Sommerville, 2011)

Capítulo 2. Descripción de la propuesta

La plataforma para la formación de Habilidades Investigativas basa su implementación en el patrón Arquitectura en Capas, específicamente en el patrón **Modelo-Vista-Controlador** (MVC). Laravel, así como la mayoría de frameworks en PHP implementan este patrón de diseño en donde cada capa maneja un aspecto de la aplicación.

Para Laravel los controladores estarán situados en la carpeta `app/Http/Controllers`, los modelos directamente en `app` y las vistas en `resources/view`.

El MVC es un patrón de arquitectura de software que separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Modelo maneja los datos del sistema y las operaciones asociadas a esos datos. El componente Vista define y gestiona cómo se presentan los datos al usuario. El Controlador dirige la interacción del usuario (por ejemplo, teclas oprimidas, clics del mouse, etcétera) y pasa estas interacciones a Vista y Modelo.

El Modelo: Debe representar la parte de la aplicación que implementa la lógica de negocio. Esto significa que debe ser responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos, controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo (Romero, y otros, 2012).

Laravel incluye un sistema de mapeo de datos relacional llamado Eloquent ORM que facilita la creación de modelos. Este ORM se funda en un patrón active record. Es opcional el uso de Eloquent, pues también dispone de otros recursos que facilitan la interacción con los datos, o específicamente la creación de modelos.

La forma de crear **Modelos** en Laravel usando Eloquent ORM, es:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Estudiante extends Model
{
    //defiendo el nombre de la tabla con la info de los estudiantes
    public $table = 'tbd_estudiantes';
}
```


Capítulo 2. Descripción de la propuesta

La Vista: La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de cualquier petición que se presente (Romero, y otros, 2012).

Laravel incluye de paquete un sistema de procesamiento de plantillas llamado Blade. Este sistema de plantillas favorece un código mucho más limpio en las Vistas, además de incluir un sistema de Caché que lo hace mucho más rápido. El sistema Blade de Laravel, permite una sintaxis mucho más reducida en su escritura. Por ejemplo, en vez pintar la vista usando el código PHP:

```
<?php echo $mi_nombre; ?>
En Blade se escribiría:
{{ $mi_nombre }}
```

Plantillas

Las plantillas en Blade son archivos de texto plano que contiene todo el HTML de la página con etiquetas que representan elementos o zonas a incluir en la plantilla, o vistas parciales como se conocen en otros frameworks PHP. Sin embargo, en Blade estos elementos incrustados se organizan en un solo archivo. Esta es una idea muy interesante de Laravel que mejora la organización de las vistas y su rendimiento. Sobre todo, cuando las vistas pueden llegar a ser muy complejas incluso con elementos anidados. En el render de una Vista completa en Laravel se usan dos archivos: la plantilla definiendo el HTML global y las zonas a incluir.

Un ejemplo típico para una plantilla HTML5 de Blade:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>@yield('titulo')</title>
</head>
<body>
  @yield('navegacion')
</body>
</html>
```

Capítulo 2. Descripción de la propuesta

En la plantilla presentada, el código `@yield()` identifica al método donde como parámetro se indica el nombre de la zona desplegar. Por otro lado, el código de la vista, donde se define la plantilla a usar y la información de las distintas zonas a desplegar:

```
<!-- identificando la plantilla a utilizar -->
@extends('template')

<!-- definiendo una zona -->
@section('titulo')

@endsection

<!-- definiendo otra zona -->
@section('navegacion')

@endsection
```

El Controlador: Gestiona las peticiones de los usuarios, es el objeto que proporciona significado a sus órdenes. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista. Los controladores pueden ser vistos como administradores, espera peticiones de los clientes, comprueba su validez de acuerdo a las normas de autenticación o autorización, delega la búsqueda de datos al modelo y selecciona el tipo de respuesta más adecuado según las preferencias del cliente. Finalmente delega este proceso de presentación a la capa de la Vista (Romero, y otros, 2012).

Los controladores contienen la lógica de la aplicación y permiten organizar el código en clases sin tener que escribirlo todo en las rutas. Todos los controladores deben extenderse de la clase *BaseController*.

```
class UserController extends BaseController {
    public function mostrarPerfil($id)
    {
        $user = User::find($id);
        return View::make('user.profile', array('user' => $user));
    }
}
```

Estos pueden ser llamados en las rutas de diferentes maneras, pero la más común usando rutas es:

```
Route::get('user/{id}', 'UserController@mostrarPerfil');
```

Capítulo 2. Descripción de la propuesta

Este modelo de arquitectura presenta varias ventajas:

Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.

Interfaz de Programación de Aplicaciones API (Application Programming Interface) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.

Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Utilidad del patrón:

El patrón Modelo-Vista-Controlador se utiliza para el diseño de aplicaciones con interfaces complejas. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad de las partes. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos (Romero, y otros, 2012).

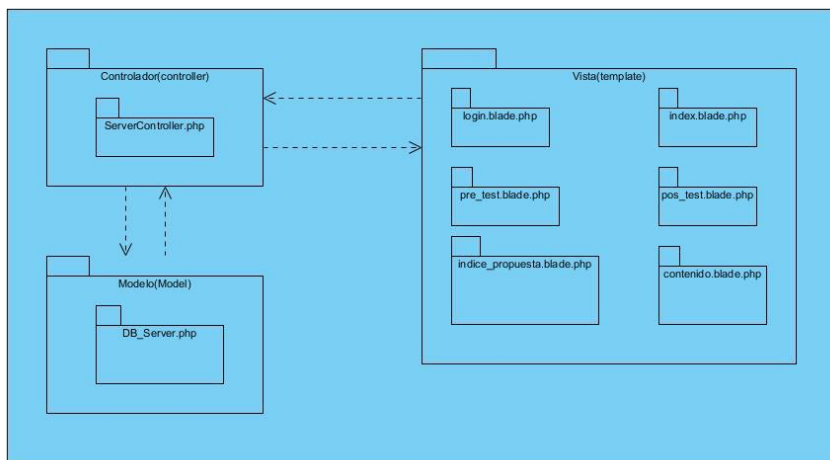


Figura 3 Representación del MVC.

Capítulo 2. Descripción de la propuesta

2.7. Patrones de Diseño

Un patrón de diseño se caracteriza como “una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución”. Para el diseño de software, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución. (Pressman, 2009)

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

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para indicar la importancia de captar (grasping) estos principios, si se quiere diseñar eficazmente el software orientado a objetos (Gamma, y otros, 2009).

Según plantea (Astudillo, et al., 2011) lo patrones GRASP son:

Experto: El patrón garantiza que la responsabilidad de la creación de un objeto o la implementación de un método, recaiga sobre la clase que conoce toda la información necesaria para crearlo lo que contribuye a un adecuado encapsulamiento, favoreciendo la robustez y fácil mantenimiento del sistema. Este patron se verá reflejado en la clase *Login()*, pues este se encarga de encapsular la lógica referente al manejo de usuarios y contraseñas.

Creador: Se asigna la responsabilidad a una clase de crear cuando contiene, agrega, compone, almacena o usa otra clase, lo que brinda una alta posibilidad de reutilizar la clase creadora. Se aplica por su responsabilidad en clase controladora <<PreguntasController.php>>, la cual crea instancias de la entidad Preguntas (Cabrera, 2019).

Controlador: Asignar la responsabilidad de administrar un mensaje de evento del sistema a una clase que representa el sistema global, dispositivo, subsistema o representa un escenario de caso de uso en el que tiene lugar el evento del sistema.

Este patrón se desplegará en todo el sistema debido a que cada uno de los eventos generados por el usuario es redirigido a una clase o función controladora que realiza las operaciones solicitadas, en la clase controladora *ServerController ()*, que se encarga de atender y ofrecer respuesta a cada una de las solicitudes del usuario relacionadas con el proceso de visualización

Capítulo 2. Descripción de la propuesta

y de generar reportes. Otros ejemplos son en las clases: <<UsuariosController.php>>, <<GruposController.php>>, entre otras.

Diagrama de Clases del Diseño:

Los diagramas de clase (DC) pueden usarse cuando se desarrolla un modelo de sistema orientado a objetos para mostrar las clases en un sistema y las asociaciones entre dichas clases. (Sommerville, 2011)

Los tipos de clases utilizados en estos diagramas son:

- Página Servidora (Server Page) para la programación lógica del servidor.
- Página Cliente (Client Page) que constituyen el código cliente.
- Formulario (Frm) que representan los formularios presentes en el código cliente.

La siguiente Figura muestra el diagrama de clase del diseño correspondiente al caso de uso Gestionar Preguntas, acorde a la arquitectura MVC. Los restantes casos de uso se comportan de igual manera partiendo del framework seleccionado por lo que se plasma como una muestra genérica.

Capítulo 2. Descripción de la propuesta

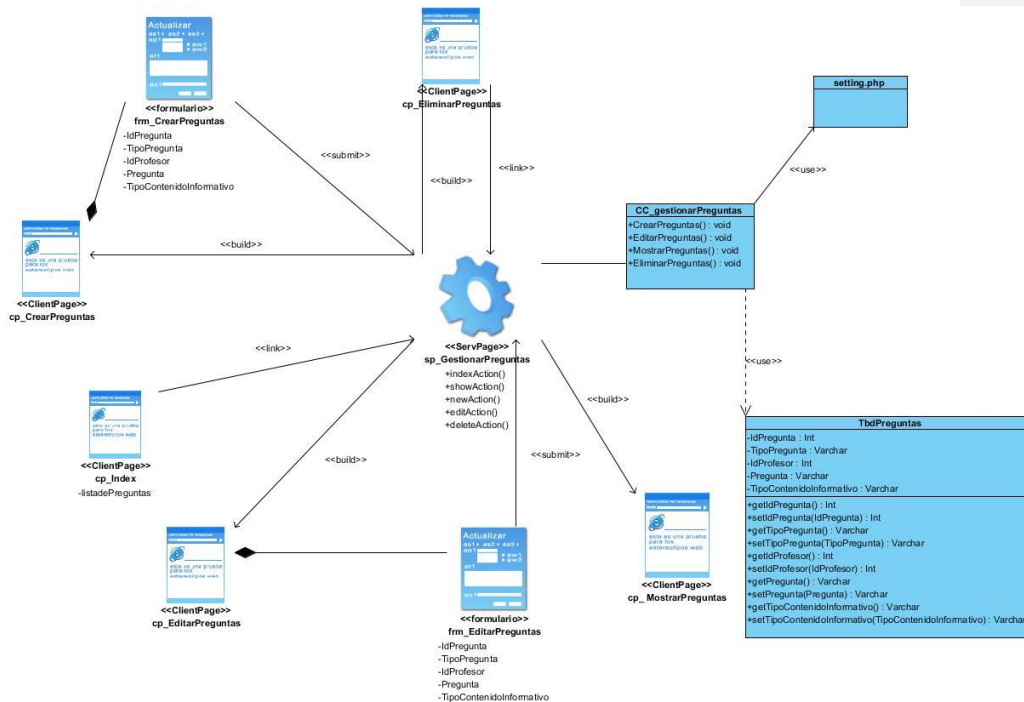


Figura 4 Diagrama de Clases del Diseño según MVC.

Descripción de las clases

Para crear preguntas la página servidora `sp_GestionarPreguntas` construye la página cliente `cp_CrearPreguntas` la cual está compuesta por el formulario `frm_CrearPreguntas` que permite el ingreso de los datos requeridos y que va a ser ejecutado por la página servidora. Para modificar las preguntas la página servidora crea la página cliente `cp_EditarPreguntas` que contiene el formulario `frm_EditarPreguntas` donde se modifican los datos y que será posteriormente ejecutado por la página servidora `sp_GestionarPreguntas`. Para acceder a los datos se hace a través de la clase controladora `CC_gestionarPreguntas`. En el caso de mostrar preguntas se accede a la página cliente `cp_MostrarPreguntas` construida por la página servidora que permite la visualización del contenido; y en el caso de eliminar preguntas, la página servidora crea la página cliente `cp_EliminarPreguntas`, donde luego interactuaría con la clase controladora, para luego esta actualizar la información dentro de la base de datos. La página cliente `cp_Index`

Capítulo 2. Descripción de la propuesta

representa la página principal encargada de mostrar el listado de las preguntas. La tabla TbdPreguntas permite el acceso a los datos almacenados.

2. 8. Modelo de datos

Modelo: Conjunto de conceptos que permite construir una representación organizacional de una institución.

Universo de discurso: Visión del mundo real del diseñador de la Base de Datos. Su definición, constituye el primer paso del diseño para poder conseguir los objetivos del diseñador.

Modelo de Datos: Conjunto de conceptos, reglas y convenciones que permiten describir, a distintos niveles de abstracción, los datos del Universo de Discurso o la estructura de una base de datos, la cual es denominada esquema, o, con otras palabras, que permiten construir una representación organizada de un sistema real. Por lo tanto, se considera un modelo de datos como una herramienta que facilita la interpretación del Universo de Discurso y la representación de un sistema de información.

El modelo de datos determina la estructura lógica de una base de datos y el modo de almacenar, organizar y manipular los datos. Dentro de sus propósitos se encuentra definir los datos persistentes que serán almacenados (PRESSMAN, 2006).

A continuación, se muestra en la Figura 6 el modelo de datos para el sistema, el cual ofrece una descripción abstracta sobre la representación de los datos en un Sistema Gestor de Base de Datos (SGBD). Los componentes que lo conforman, son entidades existentes en el propio sistema. Este modelo contiene a su vez, características propias de estos objetos y la forma en que se relacionan entre sí.

Capítulo 2. Descripción de la propuesta

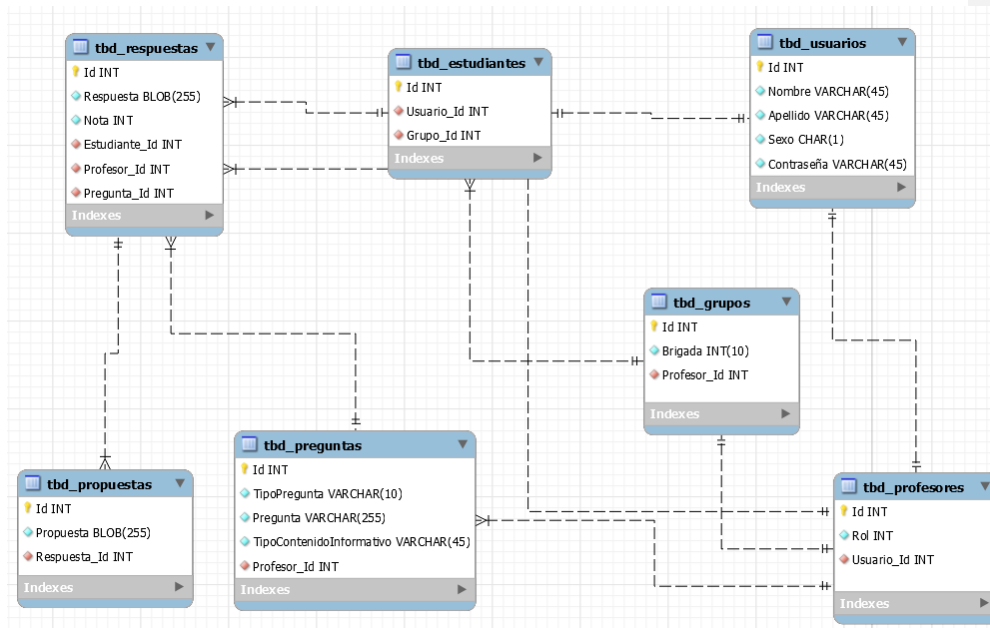


Figura 5 Modelo de Datos.

2.9. Verificación y validación

Las pruebas de software pertenecen a una actividad o etapa del proceso de producción de software denominada Verificación y Validación, usualmente abreviada como V&V, no es más que nombre genérico dado a las actividades de comprobación que aseguran que el software respeta su especificación y satisface las necesidades de sus usuarios. El sistema debe ser verificado y validado en cada etapa del proceso de desarrollo utilizando los documentos producidos durante las etapas anteriores. En este sentido, la verificación consiste en corroborar que el programa respeta su especificación, mientras que validación significa corroborar que el programa satisface las expectativas del usuario. En otras palabras, la verificación es una actividad desarrollada por ingenieros teniendo en cuenta un modelo del programa y el programa en sí, en tanto que la validación la debe realizar el usuario teniendo en cuenta lo que él espera del programa y el programa en sí (Cristiá, 2009).

Capítulo 2. Descripción de la propuesta

Validación de los requisitos funcionales

La validación de requisitos, tiene como misión, demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es el proceso de verificarlos en cuanto a validez, consistencia, integridad, realismo y certidumbre.

Técnicas de validación

Revisiones de requerimientos: comprueban consistencia, integridad, verificabilidad, comprensibilidad y rastreabilidad. Proceso manual que involucra a varios lectores, tanto del cliente como del contratista. Puede ser formal o informal. En la revisión formal, el equipo de desarrollo conduce al cliente a través de los requerimientos, explicándole las implicaciones de cada uno.

Seguridad

La seguridad en las aplicaciones informáticas es un factor que no debe ser olvidado por los desarrolladores. El registro de cada una de las actividades realizadas por el usuario autenticado en el sistema, mediante el almacenamiento de trazas de usuarios, permite mantener un control sobre las acciones realizadas en el sistema. Se establecen como políticas de seguridad la autenticación de las personas autorizadas con la cuenta del dominio uci.cu. Para garantizar la integridad de la información es utilizado el modelo de control de acceso basado en rol, el cual describe a grupos de usuarios que actúan bajo un conjunto de roles. Por tales motivos fueron definidos varios roles, asignándoles solamente los permisos de acceso a la información de acuerdo al rol que este desempeñe una vez registrado en la aplicación. Entre los roles definidos se encuentran estudiante, profesor y administrador.

2. 10. Conclusiones del capítulo

Luego de haber realizado el análisis y diseño de la Plataforma Para la Formación de Habilidades Investigativas en los Estudiantes y de haber generado los artefactos que dispone la metodología *AUP en la variante UCI Escenario 2*, se puede concluir lo siguiente:

- El análisis de las características del negocio, permitió identificar los principales requisitos funcionales y no funcionales del sistema para la gestión de los procesos de enseñanza y aprendizaje de forma no presencial.

Capítulo 2. Descripción de la propuesta

- Con la realización de modelo de datos se obtuvo una descripción abstracta sobre la representación de los datos en el Sistema Gestor de Base de Datos referente a los módulos de la plataforma.
- Se definió como patrón arquitectónico la arquitectura modelo-vista-controlador para la implementación de la propuesta de solución, lo cual permitió comprender el marco de trabajo a nivel de implementación, se definió, y por ende se pudo comprender la separación lógica y física de los componentes de la aplicación, dado que cada capa que los conforma maneja un aspecto fundamental dentro del mismo.
- La identificación de los patrones de diseño permitió disminuir el impacto de los cambios futuros en el código fuente de la misma.
- El diseño de los diagramas de clases, facilitó el enfoque en cuanto a composición lógica y física de la futura Plataforma para la Formación de Habilidades Investigativas.

Capítulo 3. Implementación y pruebas

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS.

En este capítulo se aplicarán técnicas y procedimientos para la validación del sistema informático, lo cual es un requerimiento para cumplir con las buenas prácticas y es clave a la hora de garantizar la calidad de los procesos y la comprobación de la eficiencia y la eficacia de la Plataforma para la Formación de Habilidades Investigativas.

También se presentará la estrategia para la comprobación objetiva de la hipótesis planteada en la presente tesis. Para ello se aplicará el método empírico Criterio de Expertos que busca evaluar la relevancia e implicaciones prácticas, así como la utilidad metodológica de la Plataforma de Formación de Habilidades Investigativas.

3. 1. Modelo de implementación. Diagrama de Despliegue

El modelo de despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto (Muñoz , et al., 2013). Sobre el modelo de despliegue es válido aclarar que:

- Cada nodo representa un recurso de cómputo, puede ser un procesador o un dispositivo hardware.
- La comunicación entre las PC clientes y el servidor web se establece utilizando el protocolo de comunicación segura HTTPS.
- El servidor de aplicaciones web Apache y el servidor de bases de datos MySQL, establecen una comunicación mediante la familia de protocolos TCP.
- Se establece una conexión con el LDAP utilizando el protocolo de comunicación segura HTTPS.

Capítulo 3. Implementación y pruebas

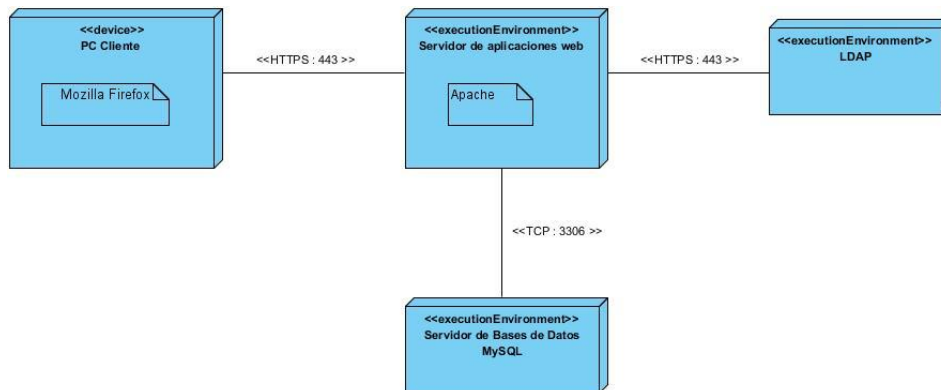


Figura 6 Modelo de Despliegue.

PC Cliente: Computadora que cuenta con un navegador actualizado y que siga los estándares web (Mozilla Firefox v 2.x o superior).

Servidor de aplicaciones Web: representa una estación donde estará montado el Servidor Apache sobre el cual correrá la aplicación.

Servidor Base de Datos: representa el servidor donde estará el SGBD MySQL que dará respuesta a las peticiones hechas por la aplicación.

3. 2. Estándar de codificación y su justificación

Implementación:

La implementación es un proceso que comienza cuando se crea una aplicación en el equipo de un desarrollador y termina cuando está instalada y lista para ejecutarse en el equipo de un usuario. Describe cómo los elementos del modelo de diseño se implementan en términos de componentes, en otras palabras, toma el resultado del modelo de diseño para generar el código final del sistema. Dicho código está determinado por el lenguaje de programación y tiene como objetivo llevar a cabo la implementación de cada una de las clases significativas del diseño.

Técnicas de programación:

Las técnicas de programación constituyen una parte fundamental en el proceso de desarrollo e ingeniería del software dentro del ámbito informático. Cada técnica tiene sus propias características, y distintos métodos de resolución de problemas, así como la implementación de estándares de ciertas compañías o instituciones.

Capítulo 3. Implementación y pruebas

Durante el desarrollo del software se utilizaron como técnicas de programación, la Programación Orientada a Objetos (POO).

Programación Orientada a Objetos:

La POO es una técnica que aumenta la velocidad de desarrollo de los programas, gracias a la reutilización de los objetos. El elemento principal de la POO es el objeto, el cual es un conjunto complejo de datos que poseen estructura y forman parte de una organización. Un objeto, contiene los datos bien estructurados y pueden ser visibles o no, dependiendo del programador y las acciones del programa en ese momento. El polimorfismo y la herencia, son unas de sus principales características.

Estándar de codificación:

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación, con vistas a generar un código de alta calidad, es de gran importancia para la calidad del software, y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido caben muchas posibilidades de que un proyecto de software, se convierta en un sistema fácil de comprender y de mantener.

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento (Microsoft, 2017).

A continuación, se define el estándar de codificación basado en el empleo de la plataforma para el Desarrollo de Habilidades Investigativas.

Tabla 4 Estándares de codificación a utilizar en la implementación del sistema

Tipo de estándar	Descripción
Organización del código	*El código se organizará por bloques *El formato del código se ajustará por la opción Reformat Code del IDE de desarrollo PhpStorm.

Capítulo 3. Implementación y pruebas

	<p>*La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios.</p>
Máxima longitud de las líneas	<p>*Todas las líneas deben estar limitadas a un máximo de setenta y nueve (175) caracteres.</p> <p>* Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas.</p> <p>*En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas</p>
Líneas en blanco	<p>*Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco .</p> <p>* Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco</p> <p>* Se puede utilizar líneas en blanco escasamente para separar secciones lógicas</p>
Codificaciones	<p>* Utilizar la codificación UTF-8</p> <p>*Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U</p>
Importación	<p>*Las importaciones deben estar en líneas separadas</p> <p>*Siempre deben colocarse al comienzo del archivo</p> <p>* Deben quedar agrupadas de la siguiente forma:</p> <ol style="list-style-type: none">1. Importaciones de la librería estándar2. Importaciones terceras relacionadas3. Importaciones locales de la aplicación / librerías <p>* Cada grupo de importaciones debe estar separado por una línea en blanco</p>
Espacios en blanco en expresiones y sentencias	<p>*Evitar utilizar espacios en blanco en las siguientes situaciones:</p> <ul style="list-style-type: none">- Inmediatamente dentro de paréntesis, corchetes y llaves.- Inmediatamente antes de una coma, un punto y coma o dos puntos.- Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función.- Inmediatamente antes de un corchete que empieza una indexación

Capítulo 3. Implementación y pruebas

	<ul style="list-style-type: none"> - Más de un espacio alrededor de un operador de asignación (u otro) para alinearlo con otro. *Deben rodearse con exactamente un espacio los siguientes operadores binarios: - Asignación (=) - Asignación de aumentación (+=, -=, *=, /=) - Comparación (==, <, >, >=, <=, !=, <>, in, not in, is, is not) - Expresiones lógicas (and, or, not) * Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad. * No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto.
Comentarios	<ul style="list-style-type: none"> * Los comentarios deben ser oraciones completas * Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula * Si un comentario es corto el punto final puede omitirse * Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres <code>"//"</code> . * Los comentarios de varias línea para organización del código aparecerán dentro de los caracteres <code>"/** ... **/"</code> .

Identación, llaves de apertura y cierre y tamaño de líneas: usar una indentación sin tabulaciones, con un equivalente a cuatro espacios, para mantener integridad en las revisiones. El uso de las llaves "{}" será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres, para mantener la legibilidad del código.

Convención de nomenclatura

Variables: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Constantes: constantes siempre debe ser todo en mayúsculas, con caracteres de subrayado "_" para separar palabras en caso de nombres compuestos.

Capítulo 3. Implementación y pruebas

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado “_” y el resto en minúscula.

Funciones: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Ficheros: todo siempre en minúscula y en caso de nombres compuestos se usa el carácter subrayado “_”.

Para las vistas: intuitivo y relacionado con el formulario y/o vista que representa el sufijo `_view`.

Para los modelos: con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_mdl` o `_base` en caso de ser modelos base.

Para las librerías: con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_lib`.

Para las controladoras: con el mismo nombre de la clase que representa.

Estructuras de control: se incluye `if`, `for`, `foreach`, `while`, `switch`, entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Si las condiciones son muy largas que sobrepasan el tamaño de las líneas, estas se dividen en varias líneas. En el mejor de los casos cuando la condición es muy extensa, esta se puede dividir en variables y compararlas dentro de la estructura de control.

Documentación: todos los archivos deben de tener la documentación asociada al mismo.

Implementación y pruebas.

Buenas prácticas: los valores booleanos y nulos siempre se escriben con mayúscula. Para facilitar la legibilidad del código usar un salto de línea antes de las estructuras de control y definición de las funciones.

Tratamiento de errores:

La informática está muy lejos de ser perfecta y lo demuestran un buen número de mensajes de error que han perseguido y acompañado a los usuarios mientras trabajaban con distintos productos de hardware y software. Para solucionar estos problemas aparece el término

Capítulo 3. Implementación y pruebas

tratamiento de errores, elemento que permite identificar, localizar, analizar y eliminar los errores en la implementación de un programa de computadora.

Las aplicaciones web deben poseer un buen mecanismo de tratamiento de errores, enfocándose no solo en las entradas de datos suministradas por el usuario, sino también en cualquier error que pueda ser generado por el comportamiento incorrecto de componentes internos, por ejemplo, errores en las consultas a la base de datos.

En la Plataforma, los datos introducidos por el usuario son validados tanto del lado del cliente como del lado del servidor. Esto garantiza que cuando se produzca un error en el sistema, este sea capaz de alertar al usuario e impedir que se produzca el envío de datos incorrectos. De esta forma se logra la disponibilidad de la aplicación y la fluidez en la interacción usuario-aplicación.

En los módulos, el usuario introduce los datos mediante las vistas, siendo estos validados en JavaScript. Una vez que se cometa algún error, se muestra un mensaje en rojo encima del campo indicando los valores válidos que deben introducirse. Además, se muestran mensajes de confirmación y de información que le indican al usuario la conclusión exitosa de una acción.

Proceso de pruebas

Las pruebas de software son actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos específicos, son un elemento crítico para la garantía de calidad del software y representan una revisión final de las especificaciones del diseño. Verifican, además, los resultados de la implementación del sistema y son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un software.

Con el objetivo de medir el grado en que los módulos cumplen con los requerimientos, se adoptará como nivel de prueba: prueba de sistema, utilizando como tipo de pruebas las pruebas funcionales. Para este caso se utilizará el método de prueba de Caja negra, que se refiere a las pruebas que se llevan a cabo sobre la interfaz del software y que permiten obtener un conjunto de condiciones de entrada, que ejercitan completamente todos los requisitos funcionales del programa.

Una de las técnicas de pruebas que se usan en este método es la partición de equivalencia. Esta divide el campo de entrada de un programa en clases de equivalencia de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran tipos de errores. Los diseños de casos de pruebas permiten guiar el proceso de prueba que se le debe realizar al sistema durante la validación del mismo.

Comentado [w3]: No plantea la estrategia de pruebas y solo habla del método de caja negra

Capítulo 3. Implementación y pruebas

3.3. Pruebas de caja negra

Las pruebas de caja negra se centran en los requisitos funcionales de software. El método de caja negra examina el programa para que cuente con todas las funcionalidades analizando los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos. Con este método se intenta encontrar los errores: de inicialización y terminación, de interfaz y en las estructuras (Sánchez Peño, 2015).

Estas pruebas hacen referencia a pruebas que se llevan a cabo sobre la interfaz del software sin tener en cuenta el código de la aplicación. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Según (Pressman, 2009) estas pruebas tienen como propósito detectar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Las pruebas de caja negra comprenden un conjunto de diferentes técnicas como, por ejemplo: Partición de Equivalencia, Análisis de Valores Límites y Grafos de Causa-Efecto. De estas técnicas quedará seleccionada para ser utilizada la **Partición de Equivalencia**, la cual divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Un caso de prueba ideal descubre de forma inmediata una clase de errores que de otro modo, se requiere la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Para realizar el proceso de las pruebas de software se tienen que generar los casos de pruebas referentes a cada uno de los requisitos de software teniendo en cuenta cada uno de los escenarios a probar en cada caso según el requisito correspondiente.

Capítulo 3. Implementación y pruebas

Para probar la solución se deben realizar tantas iteraciones de los casos de pruebas como sean necesarias con el fin de resolver cada una de las no conformidades, las cuales pueden ser catalogadas en correspondencia, interfaz y funcionalidad.

Al realizar todas las iteraciones se debe lograr obtener una solución funcional, resolviéndose todas las no conformidades detectadas.

3. 4. Validación por el método Criterio de Expertos

El proceso de validación de resultados científicos en el área de la Ingeniería de Software y el desarrollo de tecnologías web generalmente se subyugan a las pruebas o testeos de software. Sin embargo, recientes investigaciones han incorporado otros enfoques para la validación, basado en el uso combinado del método Criterio de Expertos, que tiene su aplicación en sistemas y entornos de Ingeniería Web (Torrecilla-Salinas, 2018).

En cuanto a los pasos o número de etapas que debe seguir el método, existen diferentes propuestas, se concuerda con (Astigarraga, 2005), quien propone cuatro fases para asegurar la calidad de los resultados:

- Formulación del problema.
- Elección de expertos.
- Elaboración y lanzamiento de los cuestionarios.
- Desarrollo práctico.
- Explotación de resultados.

Cada una de estas fases tiene cierta complejidad y particularidades que hay que cuidar. De la formulación del problema dependerá mucho el éxito de la aplicación del método. Este debe estar libre de ambigüedades y contar con objetivos y metas bien definidos que los expertos deben asumir. La elección de éstos es una cuestión medular en el procedimiento, sobre todo porque el concepto de experto tiene una acepción muy subjetiva.

Para la presente investigación, su definición está acorde a la que proponen (Oñate, y otros, 1990) al considerar experto “tanto al individuo en sí, como a un grupo de personas u organizaciones capaces de ofrecer valorizaciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia”. La European Aid Cooperation Office, en su apartado Evaluation, señala que los criterios para la selección de expertos evaluadores son:

Capítulo 3. Implementación y pruebas

- a) La experiencia profesional (reconocida y respetada por sus pares),
- b) La independencia del evaluador (respecto de lo que se vaya a evaluar)
- c) La capacidad de trabajo en equipo (mentalidad abierta y saber escuchar), y que hay dos maneras para conformar el grupo: por medio de una selección rápida o una selección por etapas.

La estrategia de validación que se presentará se sustenta en el diseño de un método de juicio experto, para evaluar la relevancia e implicaciones prácticas en el desarrollo ingenieril de la plataforma tecnológica, así como la utilidad para el proceso de aprendizaje de las habilidades investigativas para la escritura académica, la comunicación científica y la ingeniería y gestión de software.

Selección y caracterización del panel de expertos

El panel de expertos debe estar compuesto por expertos de las áreas de conocimientos Metodología de la Investigación Científica, Tecnología Educativa y Desarrollo de Software. Esta composición responde a criterios en el orden teórico y metodológico para valorar el componente didáctico y el impacto que tendrán los recursos digitales que componen cada unidad temática de la plataforma. También el aspecto técnico de desarrollo de la plataforma es un criterio de evaluación sometido a criterio de los expertos de desarrollo de software.

En este sentido se identificaron nueve expertos para participar en esta evaluación, de ellos cuatro de Metodología de la Investigación Científica pertenecientes a la Facultad 1 y el Centro de Innovación y Calidad de la Educación, tres expertos del área Tecnología Educativa pertenecientes al Centro Nacional de Educación a Distancia y dos expertos del área de Desarrollo de Software, específicamente de desarrollo web, pertenecientes al Centro de Innovación y Desarrollo de Internet, todos pertenecientes a la UCI. Se puede afirmar que el panel de expertos está conformado por un grupo equilibrado de personas que tienen un conocimiento adecuado de las tres áreas de conocimientos.

Para la elaboración del cuestionario (Anexo 1) se identificaron los aspectos más relevantes y que tienen un impacto directo en los estudiantes, en función de la interacción que tendrán con la plataforma virtual para la formación de habilidades investigativas para la escritura académica, la comunicación científica, así como la ingeniería y gestión de software, que deben desarrollar durante el proceso de elaboración de la tesis.

Capítulo 3. Implementación y pruebas

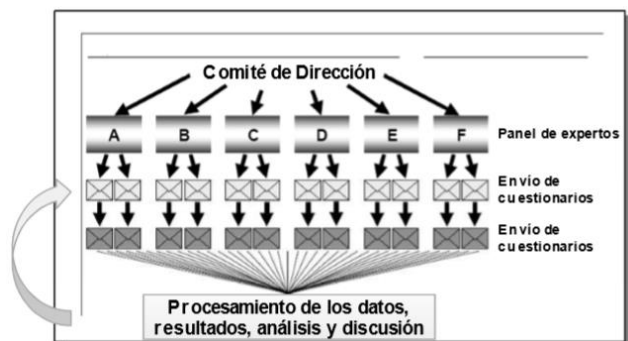


Figura 7 Representación del panel de expertos

Es importante precisar que la ejecución del método Criterio de Experto se aplicará una vez se haya desarrollado y desplegado la plataforma virtual para la formación de habilidades investigativas. Como método de consenso el criterio de experto se considera útil para alcanzar una opinión compartida. Además, se espera que la retroalimentación de los expertos también ayude a mejorar el diseño didáctico de los contenidos, así como en los aspectos técnicos de desarrollo de la plataforma.

Finalmente, la aplicación del método criterio de experto corroboraría la relevancia e implicaciones prácticas, así como la utilidad metodológica de la plataforma de formación de habilidades investigativas, lo cual constituye una etapa fundamental de la presente investigación para demostrar, en la práctica, la pertinencia y veracidad del presupuesto hipotético asumido.

3. 5. Conclusiones del capítulo

A partir de lo planteado en este capítulo se ha logrado referenciar el correspondiente entorno de proceso de verificación y validación de la propuesta de solución, en donde se propone validar la hipótesis a partir de diferentes procesos de validación que en este capítulo se exponen. Se ha hecho referencia de forma genérica a aspectos tales como técnicas de programación, estándar de codificación, convención de nomenclatura, estructuras de control y buenas prácticas. Se definieron tipos de pruebas a implementar, como lo son pruebas de sistema, funcionales, vinculandolas con las pruebas de caja negra, específicamente partición de equivalencia. De forma global cada uno de estos procesos de pruebas aportarán un valor real a la solución para el desarrollo de habilidades investigativas, debido a que las mismas, una vez solucionadas las

Capítulo 3. Implementación y pruebas

deficiencias y no conformidades detectadas, servirán de sustento para la realización de mejoras que servirán de base para futuras versiones.

La estrategia presentada para la validación por el método criterio de experto, se pondría en práctica cuando se logre el despliegue de la plataforma de formación de habilidades investigativas para la escritura académica, la comunicación científica y la ingeniería y gestión de software.

Conclusiones

CONCLUSIONES

Se concluye de la presente investigación:

- El análisis de los referentes teórico-metodológicos, así como los sistemas existentes y los procedimientos ingenieriles de configuración y personalización de plataformas virtuales para la formación universitaria, evidenció la necesidad de desarrollar una nueva plataforma para la formación de habilidades investigativas en el estudiantado.
- La selección de herramientas, lenguajes y tecnologías facilitó la definición de una estrategia para la implementación de la plataforma para la formación de habilidades de escritura académica, habilidades para la comunicación y habilidades de ingeniería y gestión de software, en los estudiantes que transitan por el proceso de elaboración del trabajo de diploma.
- Se obtiene para cada proceso del negocio el diseño y modelación correspondientes para el escenario en cuestión, lo que permitió representar de forma visual cada uno de estos procesos, aportando una visión más global del resultado a obtener.

Recomendaciones

RECOMENDACIONES

Se recomienda:

- Realizar la implementación de la plataforma para la formación de habilidades investigativas en los estudiantes que transitan por el proceso de elaboración del trabajo de diploma.
- Ejecutar las pruebas de software para detectar y corregir deficiencias presentes en las funcionalidades implementadas, validando así la propuesta de solución por parte del cliente a partir de las pruebas de aceptación.
- Aplicar la estrategia de validación de la presente investigación por el método criterio de experto para demostrar, en la práctica, la pertinencia y veracidad del presupuesto hipotético asumido.

Referencias bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

Álvarez-Villar, V. M., Pérez-Díaz, A., Durand-Rill, R., Álvarez, V. M., Pérez, A., & Duran, R. 2016. *Metodología para la formación de competencia investigativa en los estudiantes de la Universidad de Guantánamo.* EduSol, 16(55), 38–53.

Araujo, F., & Larreal, B. A. J. 2014. *Comunidades de prácticas virtuales para el desarrollo de competencias investigativas.* Opción, 75(75), 75–91.

Agut, Raúl Monferrer. 2015. *Especificación de Requisitos Software según el estándar de IEEE 830.* [PDF] 2015.

ANUIES. 2006. *Consolidación y avance de la educación superior en México.* [ed.] la Edición ed. 2006.

Astigarraga. 2005. *Criterios de calidad para la evaluación de cursos virtuales.* 2005.

Astudillo, Marcello, Visconti and Hernán. 2011. *Fundamentos de Ingeniería de Software.* s.l. : Universidad Técnica Federico Santa María, 2011.

Balboa, López. 2001. *El desarrollo de las habilidades de investigación en la formación inicial del profesorado de química.* . [Tesis inédita de doctorado] Universidad de Cienfuegos Carlos Rafael Rodríguez, Cienfuegos, Cuba : s.n., 2001.

Bolívar. 2005. Enfoque estratégico en la tutoría de la tesis de grado: un modelo alternativo para aprender a investigar en el postgrado. *Revista Universitaria de Investigación.* 2005, pp. 61-84.

Briol, Patrice. 2008. *BPMN, the Business Process Modeling Notation Pocket Handbook.* 2008.

Briones, E., & Vera, J. 2015. *Sistemas de evaluación del Trabajo de Fin de Grado y satisfacción de los estudiantes de distintas áreas de conocimiento.* In XIII FORO INTERNACIONAL SOBRE LA EVALUACIÓN DE LA CALIDAD DE LA INVESTIGACIÓN Y DE LA EDUCACIÓN SUPERIOR (FECIES). Granada.

Calzada, Aymée Hernández. 2013. *UN SISTEMA DE ACTIVIDADES EVALUATIVAS CON LA INTEGRACIÓN DE LAS TIC PARA PROPICIAR LA EVALUACIÓN FORMATIVA EN EL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA FÍSICA.* s.l., Islas Baleares : Departamento de Pedagogía Aplicada y Psicología de la Educación Universidad de las Islas Baleares, Noviembre 2013.

Camisón, et al. 2009. *¿Hacia dónde se dirige la función de calidad?: la visión de expertos en un estudio Delphi.* s.l. : Revista Europea de Dirección y Economía de la Empresa, 2009.

Carlos, M. 2001. Aprender a enseñar para la sociedad del conocimiento. *Revista complutense de educación.* 2001, pp. 531-593.

Cataldi and Zulma. 2000. Una metodología para el diseño, desarrollo y evaluación de software educativo. 2000. p. 130.

Cobo, Angel. 2005. *PHP y MySQL Server.* 2005.

Referencias bibliográficas

Colás, B. P., & de Pablos, P. J. 2012. *Aplicaciones de las tecnologías de la información y la comunicación en la investigación cualitativa*. Revista Española de Pedagogía2, LXX(251), 77–92.

Concha Martínez, Eva. 2020. Infosalus. [Online] Europa Press, 2020. <https://www.infosalus.com>.

Cristiá, Maximiliano. 2009. *Ingeniería de Software*. Facultad de Ciencias Exactas, Ingeniería y Agrimensura Universidad Nacional de Rosario : s.n., noviembre 2009. Introducción al Testing de Software.

Dago, et al. 2007. *Indicadores de riesgo de morbilidad prevenible causada por medicamentos*. s.l. : Gaceta Sanitaria, 2007.

De Pablos, J. M., Colás Bravo, P., & González, M. T. 2011. *Bienestar Docente e Innovación con Tecnologías de la Información y la Comunicación*. Revista de Investigación Educativa, 29, 59–81.

Durette, B., Fournier, M., & Lafon, M. 2016. *The core competencies of PhDs*. Studies in Higher Education, 41(8), 1355–1370. <https://doi.org/10.1080/03075079.2014.968540>

EBSCO. 2020. *EBSCO Host*. [Bases de datos de EBSCOhost] 2020.

Egbert de Smet and Sangeeta Dhamdhare. 2016. *E-Discovery Tools and Applications in Modern Libraries*. [book auth.] Egbert de Smet. *E-Discovery Tools and Applications in Modern Libraries*. Hershey PA : Information Science Reference, 2016.

Eguilíz Pérez, Javier. 2008. *Introducción a CSS*. 2008.

ELSEVIER. 2020. *ELSEVIER América Latina*. s.l., América Latina : ScienceDirect, julio 17, 2020.

Estrada, M. O., Alfonso, P. M., Hidalgo, I. L., Blanco, H. S. M., & Ciudad, R. F. Á. 2014. *Procedimiento para determinar las Tendencias Estadísticas del Desarrollo de la Competencia Investigativa del Ingeniero en Ciencias Informáticas*. GECONTEC: Revista Internacional de Gestión Del Conocimiento Y La Tecnología, 2(2), 2255–5684.

Estrada, M. O., Blanco, H. S. M., & Ciudad, R. F. Á. 2015. *Exigencias didácticas en diseño didáctico de tareas para el desarrollo de las habilidades investigativas*. Enseñanza & Teaching, 33(2), 191–212. <https://doi.org/10.14201/et2015332191211>

Esmerlis; Edgar. 2015. *Competencias Investigativas: Desarrollo de habilidades para la construcción del conocimiento en la formación profesional*. 2015. pp. 1418-1424.

Expósito, María Lucía. 2019. *Aclara tus dudas escolares desde casa con el Repasador en Línea*. La Habana, Cuba : Juventud Técnica , septiembre 24, 2019.

Fabien Potencier and François Zaninotto . 2015. *Symfony, la guía definitiva*. 2015.

Ferré Grau, Xavier and Sánchez Segura, María Isabel. 2018. *Desarrollo Orientado a Objetos con UML*. 2018.

Referencias bibliográficas

Gamma, E., Helm, R. Johnson and R. Viissides, J. 2009. Patrones de diseño. *Patrones de diseño*. [Online] 2009. [Cited: febrero 28, 2017.] <http://www.vico.org/pages/PatronsDissey.html...>

García Martínez. 2009. Consideraciones sobre el Uso de Espacios Virtuales en la Formación de Investigadores. . *Revista de Informática Educativa y Medios Audiovisuales*. 2009.

Gisela. 2019. *Servidor web*. Santiago, Chile : s.n., 2019.

Guevara, Yurisander. 2017. *Educación cubana, en línea y más allá*. s.l., La Havana, Cuba : Juventud Rebelde, septiembre 21, 2017. ISSN 1563-8340.

Hernández, Roxana Brito. 2020. *Sistema informático para la gestión de información de la consulta de Oncología*. [Documento] La Habana : s.n., 2020.

Hibernate. 2018. Hibernate. *Hibernate*. [Online] Noviembre 7, 2018. <http://hibernate.org/orm/>.

Horruitiner, S. P. 2009. *LA UNIVERSIDAD CUBANA MODELO DE FORMACIÓN*. La Habana: Editorial Universitaria.

IBM Knowledge Center. 2018. IBM Knowledge Center. *IBM Knowledge Center*. [Online] Octubre 8, 2018.

Laguna, Miguel A. 2018. *Ingeniería del Software I*. [ed.] I.T.I.Gestión. 2018. Vol. 3.

Lagunes, Agustín Domínguez, et al. 2015. *Plataformas educativas virtuales como apoyo a la tutoría de investigación para estudiantes universitarios*. diciembre 24, 2015.

Landazábal; Claro; Cruz. 2007. *Visibilidad y formación en investigación. Estrategias para el desarrollo de competencias investigativas*. s.l. : Revista studiosita, 2007. pp. 43-56.

Laravel. 2019. *Laravel es un popular framework de PHP. Permite el desarrollo de aplicaciones web totalmente personalizadas de elevada calidad*. 2019.

Linda; Ligia; Jency; Yarit; Marianella. 2012. Competencias investigativas a en el personal académico de la escuela de ciencias de la educación de la UNED. *Journal of the Costa Rican Distance Education University*. 2012, pp. 273-282.

Machado and Montes de Oca. 2019. s.l. : Humanidades Médicas, mayo 2, 2019. Las habilidades investigativas y la nueva Universidad: Terminus a quo a la polémica y la discusión.

Manizales. 2014. Tutores inteligentes mejoran aprendizaje en matemáticas. 2014.

Marquès and Pere. 1995. Biblioteca virtual de tecnología educativa. *El software educativo*. 1995.

Marta & Nora. 2013. *Desarrollo de Competencias Investigadoras en la Sociedad Red*. Revista de Educación a Distancia. 2013, pp. 1-10.

Martín, Antonio, León, Carlos and García, Antonio. 2014. *INNOVACION DOCENTE PARA LA INTEGRACIÓN DE AUTOFORMACIÓN Y AUTOEVALUACIÓN EN LA PLATAFORMA WEBCT*. Sevilla, España : Píxel-Bit.Revista de Medios y Educación, enero 2014. Vol. 44. ISSN: 1133-8482.

Referencias bibliográficas

Martínez , Medina and Salazar. 2018. *Desarrollo de competencias investigativas en los estudiantes.* s.l. : Revista electrónica Opuntia Brava, 2018.

Mayor, Alicia Cañellas. 2016. *CMS, LMS y LCMS Definición y diferencias.* 2016.

MES. 2013. *PLAN DE ESTUDIOS “D” INGENIERÍA EN CIENCIAS INFORMÁTICAS.* La Habana.

Mendeley. 2020. *Métodos “I + D” de la Informática.* Presentación. 2020.

Elisa Barchini, Graciela. 2005. *Revista de Informática Educativa y Medios Audiovisuales,* pp. 16-24.

Miranda , Guillermo Luis Herrera. 2014. *Concepción pedagógica del proceso de formación de habilidades investigativas.* Pinar del Río : Revista de Ciencias Médicas de Pinar del Río , 2014. Vol. 18.

Molina; María. 2013. *ORIENTACIONES METODOLÓGICAS PARA EL DESARROLLO DE LA COMPETENCIA INVESTIGATIVA DESDE EL PROCESO DE DESARROLLO DE SOFTWARE.* Universidad de las Ciencias Informáticas : s.n., 2013. pp. 105-121.

Morelos. 2020. *¿Qué es Apache? 2020.*

Moreno. 2005. *Potenciar la educación.* s.l. : Revista Electrónica Iberoamericana sobre Calidad, Eficacia y Cambio en Educación, 2005. Un currículum transversal de formación para la investigación, Vol. 1.

Muñoz , Yudaimy Estevez and Fal , Ariel Gómez . 2013. *Solución para la gestión de la estructura investigativa de la Universidad de las Ciencias Informáticas.* junio 2013. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas .

Muñoz , Yudaimy Estevez and Fal, Ariel Gómez. 2013 . *Solución para la gestión de la estructura investigativa de la Universidad de las Ciencias Informáticas.* Ciudad de La Habana : s.n., junio 2013 .

MySQLOficial. 2020. 2020.

Pagani, L., Argentin, G., Gui, M., & Stanca, L. 2016. The impact of digital skills on educational outcomes: evidence from performance tests. *Educational Studies*, 42(2), 137–162. <https://doi.org/10.1080/03055698.2016.1148588>

Olabuénaga, Ruiz and Ispizua. 1989. *La técnica Delphi. La descodificación de la vida cotidiana. Métodos de investigación cualitativa.* Bilbao : s.n., 1989.

Oñate and Martínez. 1990. *Utilización del método Delphi en la pronosticación: Una experiencia inicial* Instituto de Investigaciones Económicas. Cuba : Cuba: JUCEPLAN, 1990.

Ortega. 2008. *El método Delphi, prospectiva en Ciencias Sociales a través del análisis de un caso práctico.* s.l. : Revista Escuela de Administración de Negocios, septiembre-diciembre 2008.

Parisca. 1995. *El método Delphi. Gestión tecnológica y competitividad. En Parisca, S.* [Document] La Habana: Academia : s.n., 1995.

Referencias bibliográficas

- Peñalvo, Francisco José García. 2018.** *MODELO DE DOMINIO*. Salamanca : s.n., 2018. p. 35.
- Pérez Porto, Julián and Merino, María. 2018.** Definicion.de. *Definicion.de*. [Online] 2018. <https://definicion.de/organizacion/>.
- Piattini, Mario. 1996.** *Análisis y diseño detallado de aplicaciones informáticas de gestión*. 1 , Madrid : RA-MA Editorial, 1996.
- PMBOK. 2013.** *Guía de los fundamentos para la dirección de proyectos* . 5ta. Newtown Square : Project Management Institute, Inc, 2013. p. 490.
- PostgreSQL. 2018.** PostgreSQL. *PostgreSQL*. [Online] Noviembre 7, 2018. <https://www.postgresql.org/docs/9.4/static/intro-what-is.html>.
- Pressman. 2009.** *Ingeniería De Software, Un Enfoque Práctico*. . 2009.
- PRESSMAN. 2006.** *Ingeniería del Software. Un enfoque Práctico*. Madrid : s.n., 2006.
- Quintero , Linda Castañeda and Vicent, Patricia López . 2007.** *Herramientas Telemáticas Para La Enseñanza Universitaria En El Marco Del Espacio Europeo De Educación Superior*. . [CD – ROM] Murcia : Grupo de Investigación de Tecnología Educativa, 2007. ENTORNOS VIRTUALES DE ENSEÑANZA APRENDIZAJE LIBRES: MOODLE. ISBN: 978-84-611-7947-3 .
- Radio Cubana. 2019.** Educación cubana lanza plataforma para impulsar el estudio virtual. [ed.] ICRT. Octubre 24, 2019.
- RAE. 2015.** *Diccionario de la lengua española*. (Vigésima segunda edición ed.), Madrid, España : s.n., 2015.
- Ramírez, Machado, Recio, Montes de Oca and Campos, Mena . 2008.** El desarrollo de habilidades investigativas como objetivo educativo en las condiciones de la universalización de la educación superior. *Pedagogía Universitaria*. 2008, Vol. XIII, pp. 156-180.
- Ramos, Chirino. 2002.** *Perfeccionamiento de la formación inicial investigativa de los profesionales de la educación*. [Tesis inédita de doctorado] La Habana, Cuba : Instituto Superior Pedagógico Enrique José Varona, 2002.
- Reiban, B. R. E., De la Rosa, R. H., & Zeballos, C. J. M. 2017.** *Competencias investigativas en la Educación Superior*. Publicando, 4 No 10.(May).
- Rekalde, R. I. 2011.** *¿Cómo afrontar el trabajo fin de grado? Un problema o una oportunidad para culminar con el desarrollo de las competencias*. *Revista Complutense de Educación*, 22(2), 179–193.
- Roberto Barchino, José M Gutiérrez and Salvador Otón . 2006.** *Panorámica de las Herramientas de Apoyo a la Teleformación*. Universidad de Alcalá, Alcalá de Henares, España : Departamento de Ciencias de la Computación,, 2006.
- Rodríguez , Dianelkys Martínez and Delgado , Dora Lilia Márquez. 2014.** *LAS HABILIDADES INVESTIGATIVAS COMO EJE TRANSVERSAL DE LA FORMACIÓN PARA LA INVESTIGACIÓN*. Pinar del Río : TENDENCIAS PEDAGÓGICAS, 2014.

Referencias bibliográficas

Rodríguez, Dianelkys Martínez and Delgado, Dora Lilia Márquez. 2014. *Tendencias de la formación y desarrollo de habilidades investigativas en el pregrado.* [ed.] Juan Carlos Martínez Coll. s.l. : Tlatemoani. Revista académica de investigación, 2014.

Rodríguez, Yuniort Portilla, Rizo, Cesar Labañino and Carranza, Marianela Granado. 2015. *OPALE una alternativa para el desarrollo de objetos de aprendizajes.* Holguín : Centro de Información y Gestión Tecnológica de Santiago de Cuba, 2015. pp. 1-18.

Romero, Yenisleidy Fernández and González, Yanette Díaz . 2012. Patrón Modelo-Vista-Controlador. 2012, Vol. 11, pp. 47-57.

Rouse, Margaret and Cameron, McKenzie. 2017. TheServerSide. *TheServerSide.* [Online] Marzo 2017. <https://www.theserverside.com/definition/J2EE-Java-2-Platform-Enterprise-Edition>.

SeamFramework.org. 2018. SeamFramework. *SeamFramework.* [Online] Noviembre 7, 2018. <http://seamframework.org/Home.html>.

Softel. 2017. Softel Soluciones Informáticas. *Softel Soluciones Informáticas.* [Online] 2017. <http://www.softel.cu/portal/index>.

Sommerville. 2011. *Ingeniería de Software.* Mexico : Pearson Educación de México, 2011. ISBN: 978-607-32-0603-7.

Thomson Reuters. 2020. s.l. : EndNote, 2020.

Toro; Pedro. 2010. El proceso de formación investigativa del profesional ingeniero y las competencias investigativas. 2010.

Torrecilla-Salinas, C.J., De Troyer, O., Escalona, M.J. et al. A. 2019. Delphi-based expert judgment method applied to the validation of a mature Agile framework for Web development projects. *Inf Technol Manag* 20, 9–40. <https://doi.org/10.1007/s10799-018-0290-7>

Torres, Rodríguez, et al. 2018. *Herramientas metodológicas para el desarrollo de habilidades investigativas en los estudiantes: una praxis necesaria.* s.l. : Revista Científica Olimpia, 2018.

Universidad de las Ciencias Informáticas. 2019. *Metodología de desarrollo para la Actividad productiva de la UCI.* [Documento] La Habana : PROGRAMA DE MEJORA, 2019.

Universidad Nacional Abierta y a Distancia. 2016. *Diagramas de Casos de Uso.* s.l. : Creative Commons , 2016.

Vallejo, S. C. 2011. *Evaluación de competencias investigativas.* XIII Conferencia Interamericana de Educación Matemática. Recife.

Vélez. 2006. El docente ante los retos educativos del siglo XXI. *Pampedia.* 2006, pp. 55-59 .

—. 2003. *El método Delphi.* s.l. : Universidad Tecnológica de Bolívar Department of Finance and International Business - Instituto de Estudios para el Desarrollo (IDE). , 2003.

Visual Paradigm. 2018. Visual Paradigm. *Visual Paradigm.* [Online] Noviembre 7, 2018. https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html.

Anexos

ANEXOS

ANEXO 1. Cuestionario para expertos

Estimado experto (a), la presente encuesta forma parte de una investigación en el área de las Ciencias Informáticas que está dirigida al desarrollo de una plataforma tecnológica para la formación de habilidades investigativas en los estudiantes universitarios. Por cuanto estamos convencidos de que sus valoraciones acerca de los asuntos que sometemos a su consideración nos servirán de considerable ayuda, le solicitamos la más responsable atención a esta consulta.

I- Datos generales del encuestado: Institución y departamento donde labora: _____

Título universitario: _____

Categoría científica: _____

Categoría docente: _____

Años de experiencia en la docencia: _____ Años de experiencia en la producción de software:

_____. Según resultados obtenidos en diferentes investigaciones realizadas con anterioridad, hemos extraído los siguientes indicadores que someteremos a su consideración en esta consulta para juzgar su efectividad en la medición de las variables de la tesis.

Es por ello que el objetivo de la presente encuesta consiste en que usted evalúe cada uno de los indicadores que se le presentarán. Para expresar su evaluación, por favor, luego de analizar cuidadosamente el material auxiliar que se adjunta, evalúe a cada uno de los indicadores que se le presentan en la tabla de la subsiguiente sección II, marcando con una cruz en la casilla correspondiente y teniendo en cuenta para ello el siguiente código de categorías de clasificación: 5: Manifestación MUY ADECUADO; 4: Manifestación BASTANTE ADECUADO; 3: Manifestación ADECUADO; 2: Manifestación POCO ADECUADO; 1: Manifestación INADECUADO.

II. Listado de indicadores a valorar:

No	Indicador	5	4	3	2	1
1	Cómo evalúa usted el desarrollo de habilidades investigativas en la carrera Ingeniería en Ciencias Informáticas (ICI)					
2	Cómo evalúa usted las capacidades investigativas demostradas por los estudiantes de 5to año de ICI					
3	Qué valoración pudiera otorgar a las habilidades de desarrollo intelectual de los estudiantes que se enfrentan a la realización del trabajo de diploma.					

Anexos

4	Qué valoración pudiera otorgar a las habilidades comunicativas de los estudiantes que se enfrentan a la realización del trabajo de diploma.								
5	Qué valoración pudiera otorgar a las habilidades de Ingeniería y Gestión de Software en los estudiantes que se enfrentan a la realización del trabajo de diploma.								
6	Cómo valora el diseño didáctico de los contenidos en la Plataforma de formación de habilidades investigativas.								
7	Qué opina sobre las herramientas de diagnóstico y evaluación presentes en la plataforma de formación de habilidades investigativas.								
8	Cómo evalúa las herramientas de comunicación presentes en la plataforma, donde participan tanto estudiantes como profesores.								
9	Cómo valora los reportes gestionados desde la plataforma.								

III. Si desea exponer cualquier otra opinión, por favor, exprese en el espacio disponible a continuación.

VI. Como parte del método de procesamiento de los datos obtenidos por medio de la presente encuesta, necesitamos caracterizar estadísticamente la competencia del conjunto de expertos del cual usted forma parte, por lo que finalmente le rogamos nos ayude respondiendo lo más fielmente posible al siguiente TEST DE AUTOVALORACIÓN DEL CONSULTADO: a) Evalúe su nivel de dominio acerca de la esfera sobre la cual se le consultó marcando con una cruz sobre la siguiente escala (1: dominio mínimo; 10: dominio máximo)

1	2	3	4	5	6	7	8	9	10

MUCHAS GRACIAS POR SU COLABORACIÓN.