



**Universidad de las Ciencias Informáticas
Facultad 5**

*Sistema informático de apoyo al aprendizaje en la
asignatura Física para el 11no grado.*

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

Autor:

Yaser Forcelledo Sarmiento.

Tutores:

MSc. Manuel Villanueva Betancourt.

Ing. Julio César Espronceda Pérez.

Co-Tutores:

Ing. Alberto Jesús Pérez Galban.

Ing. Carlos Manuel Castillo Chacón.

La Habana, julio 2016.

“Año 58 de la Revolución”

Declaración de autoría.

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente: a los ____ días del mes de _____ del año 2016.

Yaser Forcelledo Sarmiento

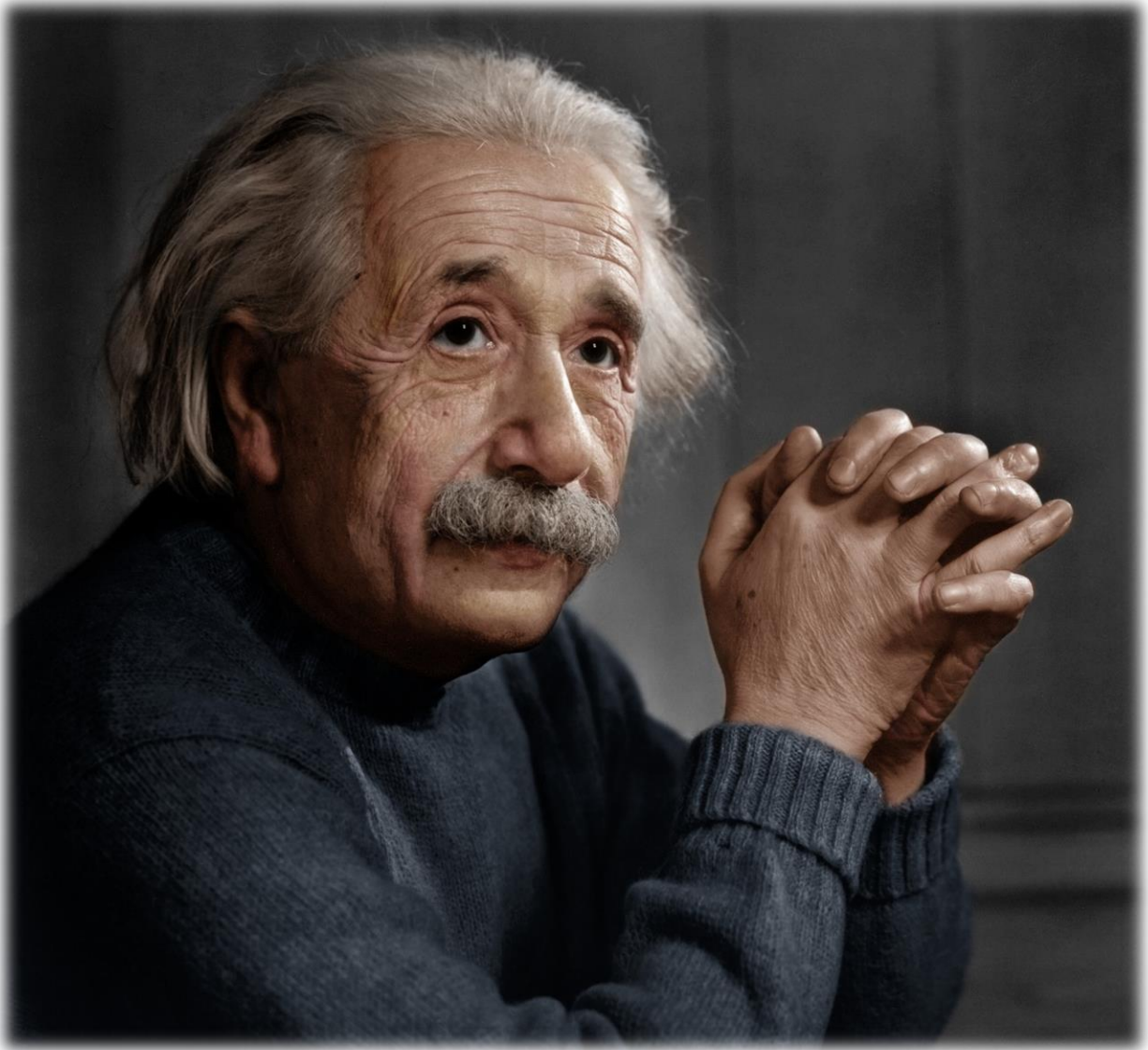
Autor

MSc. Manuel Villanueva Betancourt
Tutor

Ing. Julio César Espronceda Pérez.
Tutor

Ing. Alberto Jesús Pérez Galban
Co – Tutor

Ing. Carlos Manuel Castillo Chacón
Co – Tutor



"En lugar de ser un hombre exitoso, busca ser un hombre valioso, lo demás llegará naturalmente."

"Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad."

Albert Einstein.

Dedicatoria

Dedico este trabajo a mis padres. A mi madre por todo el amor, cariño y apoyo que siempre me ha dado, porque el amor de una madre es el amor más puro e incondicional que existe en el mundo. A mi padre por ser mi guía, súper héroe preferido y el que siempre me ha aconsejado durante todos estos años de mi vida. A los dos muchas gracias por todo, vivo y viviré orgulloso de ustedes toda la vida, los AMO.

A toda mi familia y amigos.

Agradecimientos

A mis padres por siempre confiar en mí y darme fuerzas para que este día se convirtiera en realidad.

A mi abuela Ada, la súper abuela, por ser tan especial y ser la gran madre de la familia.

A mi hermana Yeni, que, aunque cuando éramos pequeños siempre estábamos fajados, desde que crecimos me ha apoyado y siempre ha luchado por mantener a la familia bien unida; y ya nos regaló un Bryan que será el futuro profesional de la familia.

A mi cuñado Gerardo por ser un gran hombre, amigo, buen padre y muy atento con toda la familia; por eso y más es muy querido por todos.

A Milagro por siempre atenderme como si fuera su hijo, apoyarme y ser como una madre para mí.

A toda mi familia, de Camagüey, Holguín, Guardalavaca y los que están en Canadá que siempre me han apoyado.

A mis tutores por darme el tema de tesis y por la incondicionalidad en todo este tiempo de duro esfuerzo y sacrificio, por ser capitanes y marineros a la vez en esta tarea, gracias por apoyarme, y contribuir a la formación para hoy convertirme en ingeniero.

Al tribunal y oponente, por todas las sugerencias que me dieron en cada corte de tesis para que todo saliera lo mejor posible.

A Alberto Jesús por ser un gran amigo, al que convivimos varios años en el mismo cuarto y siempre nos llevamos muy bien. Por darme fuerzas, confianza y ayudarme cada día que pase en esta universidad.

A Carlos Manuel Castillo Chacón, por ser mi amigo, hermano y por todo el apoyo durante la realización de este trabajo, por todas las noches que te moleste. Solo tú y yo sabemos el trabajo que se pasó, gracias por todo.

A Liuver, Jorge Alfonso y Orlando Cruz, por la ayuda que me dieron en este trabajo.

A mis amigos y amigas de Holguín, los amigos de siempre y de toda la vida; que no pueden estar presentes, pero siempre me apoyan y están en contacto conmigo y sé que hoy en la distancia están junto a mí. Ellos son, Ayenay, Violeta, Yaisa, José, Lores, Isidro, Tito, Alexis, Frank, Moreno, Luis Ángel, Carlos del Toro y otros.

A todos los profesores que he tenido durante la carrera que han contribuido a que pueda convertirme en un profesional; en especial a las profesoras Zaida y Yirka por todo el apoyo siempre a sus estudiantes.

A Félix Páez, por ser siempre mi compañero de estudio y equipo en todos los trabajos durante estos años.

A todos los amigos que encontré aquí en la UCI, a mis compañeros de apartamento tanto del edificio 87 como del 91, Alberto (El Flaco), Carlos, Costa (Jiménez), Cuevas (El Tiburón), Emilio (El Bebé), Jorge Alfonso, José (Koko), Luis Enrique (El Cama), Edward, Roberto (El Pop), El Lachy, Yoan (El Nino) y Yoan (El Emo); por todos los buenos y malos momentos compartidos durante todos estos años en la universidad.

A mis compañeros de la brigada 5501 por todos los momentos compartidos durante todos estos años y experiencias vividas juntos. A Leduan por ser mi amigo y un gran líder dentro de la brigada.

A las chicas (las peluas) del 90 201, por soportarme y apoyarme en estos últimos tiempos.

A mis compañeros de mi antigua aula, por todo el apoyo y los momentos que pasamos juntos.

A todos los que, de una forma u otra, han estado presentes en mi vida y permitieron que este trabajo fuera posible y quienes, con mayor o menor esfuerzo, aportaron su granito de arena por ayudar.

A los que han estado presentes y a los que no también; a todos,

Muchas Gracias.

Resumen

Inmersos en todas las transformaciones que se llevaron a cabo con la Batalla de Ideas, los Institutos Pre – Universitarios fueron equipados con medios tecnológicos que en la mayoría de los casos no son explotados al máximo en el proceso de enseñanza aprendizaje, por no incluir *software* necesario que preste apoyo en la mayoría de las asignaturas. Apoyo tan necesario y en ocasiones imprescindible en muchas de las asignaturas donde estos centros carecen de instrumental necesario para llevar a cabo demostraciones didácticas o laboratorios equipados que permitan la experimentación. El presente trabajo se propone como objetivo fundamental, el desarrollo de un sistema informático que de apoyo a la asignatura de Física en 11no grado, con el fin de abrir nuevas vías de enseñanza, de intercambio y de trasmisión del conocimiento hacia el estudiantado, convirtiéndola en una poderosa herramienta que aporte novedosos caminos para el desarrollo educativo. Además de implantar hábitos de trabajo en grupo, desarrollar habilidades en el manejo de las tecnologías y la familiarización con equipos e instrumentos asociados al trabajo experimental en la asignatura. Para dar cumplimiento al objetivo se desea obtener el desarrollo de un sistema con la capacidad de vincular ejercicios, cuestionarios y demostraciones virtuales que apoyen el proceso de enseñanza aprendizaje de los estudiantes.

Palabras clave: demostraciones virtuales, física, proceso enseñanza aprendizaje, sistema informático.

Índice

| | |
|---|----|
| Introducción..... | 1 |
| Capítulo 1. Fundamentación teórica de la investigación..... | 5 |
| 1.1 El software educativo en el proceso de enseñanza – aprendizaje..... | 5 |
| 1.2 Herramienta y sistemas informáticos de apoyo a la docencia..... | 6 |
| 1.2.1 Plataforma ATutor..... | 6 |
| 1.2.2 Plataforma Moodle..... | 7 |
| 1.2.3 Sistemas informáticos nacionales..... | 9 |
| 1.2.4 Sistemas informáticos internacionales..... | 12 |
| 1.3 Sistemas basados en casos..... | 14 |
| 1.4 Metodología de desarrollo de software..... | 14 |
| 1.4.1 Proceso Racional Unificado (RUP)..... | 15 |
| 1.4.2 Programación Extrema (XP)..... | 16 |
| 1.4.3 Selección de la metodología de desarrollo..... | 16 |
| 1.5 Tecnologías y herramientas de desarrollo utilizadas..... | 17 |
| 1.5.1 Tecnologías del lado del cliente..... | 17 |
| 1.5.2 Tecnología del lado del servidor..... | 18 |
| 1.5.3 Marco de trabajo para el desarrollo..... | 19 |
| 1.5.4 Herramienta CASE..... | 22 |
| 1.5.5 Entorno de desarrollo integrado..... | 22 |
| 1.5.6 Sistema gestor de base de datos..... | 23 |
| 1.5.7 Servidor web..... | 24 |
| 1.6 Conclusiones parciales..... | 25 |
| Capítulo 2. Sistema Informático de Apoyo al Aprendizaje en la Asignatura Física (SIAF) | 26 |

| | | |
|-------------|--|----|
| 2.1 | Descripción de la solución propuesta..... | 26 |
| 2.1.1 | Sistema Informático SIAF | 26 |
| 2.1.2 | Sistema experto basado en casos para recomendar ejercicios a los estudiantes | 27 |
| 2.2 | Exploración..... | 30 |
| 2.2.1 | Historias de Usuario..... | 30 |
| 2.3 | Planificación | 33 |
| 2.3.1 | Lista de reservas del producto | 33 |
| 2.3.2 | Plan de Iteraciones | 36 |
| 2.3.3 | Plan de Entrega | 37 |
| 2.4 | Fase de diseño..... | 38 |
| 2.4.1 | Patrón de arquitectura Modelo Vista Controlador | 38 |
| 2.4.2 | Patrones de diseño | 39 |
| 2.4.3 | Tarjetas CRC | 41 |
| 2.5 | Modelo de datos..... | 43 |
| 2.6 | Conclusiones parciales..... | 44 |
| Capítulo 3. | Implementación y pruebas..... | 45 |
| 3.1 | Tareas de Ingeniería | 45 |
| 3.2 | Estándares de Codificación | 46 |
| 3.2.1 | Definiciones de clases | 47 |
| 3.2.2 | Definición de métodos | 47 |
| 3.2.3 | Asignaciones a variables | 47 |
| 3.2.4 | Estructuras de control | 48 |
| 3.3 | Diagrama de despliegue..... | 48 |
| 3.4 | Fase de Pruebas | 49 |

| | |
|---------------------------------------|----|
| 3.4.1 Pruebas de Aceptación | 50 |
| 3.4.2 Resultados de las pruebas | 54 |
| 3.5 Conclusiones parciales..... | 55 |
| Conclusiones Generales | 56 |
| Recomendaciones..... | 57 |
| Referencias bibliográficas..... | 58 |
| Anexos | 61 |
| Anexo 1: Historias de Usuario..... | 61 |
| Anexo 2: Tarjetas CRC | 65 |
| Anexo 3: Tareas de ingeniería | 68 |

Índice de Tablas

| | |
|---|----|
| Tabla 1: HU2 Crear usuario..... | 32 |
| Tabla 2: HU20 Evaluar ejercicios. | 32 |
| Tabla 3: HU23 Recomendar ejercicios. | 33 |
| Tabla 4: Requisitos Funcionales..... | 34 |
| Tabla 5: Requisitos no funcionales..... | 36 |
| Tabla 6: Plan de iteraciones. | 37 |
| Tabla 7: Plan de entregas..... | 38 |
| Tabla 8: Tarjeta CRC UsuarioController. | 42 |
| Tabla 9: Tarjeta CRC RecursoController..... | 42 |
| Tabla 10: Tarjeta CRC CuestionarioController. | 43 |
| Tabla 11: Tarea de ingeniería 1 HU2 Crear usuario..... | 45 |
| Tabla 12: Tarea de ingeniería 20 HU20 Evaluar ejercicio. | 46 |
| Tabla 13: Tarea de ingeniería 23 HU23 Recomendar ejercicio. | 46 |
| Tabla 14: Prueba de aceptación para la HU Crear usuario. | 52 |
| Tabla 15: Prueba de aceptación para la HU Evaluar ejercicio..... | 53 |
| Tabla 16: Prueba de aceptación para la HU Proponer ejercicio..... | 54 |
| Tabla 17: HU Autenticar usuario..... | 61 |
| Tabla 18: HU Modificar usuario. | 62 |
| Tabla 19: HU Eliminar usuario..... | 62 |
| Tabla 20: HU Crear recurso..... | 64 |
| Tabla 21: HU Proponer ejercicio..... | 64 |
| Tabla 22: Tarjeta CRC EstudianteController. | 65 |
| Tabla 23: Tarjeta CRC GrupoController. | 66 |
| Tabla 24: Tarjeta CRC TemaController. | 66 |
| Tabla 25: Tarjeta CRC SubtemaController..... | 67 |
| Tabla 26: Tarjeta CRC EjercicioController..... | 68 |
| Tabla 27: Tarea de ingeniería Autenticar usuario..... | 68 |
| Tabla 28: Tarea de ingeniería Modificar usuario. | 68 |

Índice de Figuras

| | |
|---|----|
| Figura 1: Modelo de Datos. | 44 |
| Figura 2: Definición de clases..... | 47 |
| Figura 3: Definición de métodos. | 47 |
| Figura 4: Asignación a variables..... | 47 |
| Figura 5: Estructuras de control..... | 48 |
| Figura 6: Diagrama de Despliegue. | 48 |
| Figura 7: Pruebas de aceptación satisfactorias por iteraciones..... | 54 |

Introducción

Las nuevas Tecnologías de la Información y las Comunicaciones (TIC) se han desarrollado desorbitantemente en los últimos años. En la actualidad las TIC están en todas partes y se han convertido en un instrumento imprescindible del desarrollo social, político y económico de los países, que ha introducido la necesidad de automatizar los procesos empresariales como punto de partida de la creación de una sociedad relacionada con su entorno.

Las TIC constituyen una oportunidad para el desarrollo del trabajo colaborativo y de aprendizaje. La sociedad en su conjunto vive hoy un proceso de transformación que afecta la manera de cómo se relacionan, se organizan, trabajan, aprenden, investigan, crean y se comunican sus miembros. Caracterizada por una constante circulación de información, donde el conocimiento es un recurso flexible, fluido, en continua expansión y movimiento (1).

Esta sociedad exige ciudadanos con habilidades y competencias que les permitan desempeñarse de forma adecuada, así como, «manipular y actualizar el conocimiento, seleccionar lo apropiado en un contexto particular, aprender de manera permanente y comprender lo que se aprende, de tal forma que pueda adaptarlo a situaciones nuevas y de rápido cambio» (2).

Por lo cual hoy en día la educación a todos los niveles demanda de actualización constante y del aprovechamiento de estas nuevas tecnologías en el Proceso de Enseñanza Aprendizaje (PEA) de nuestros estudiantes.

Para el desarrollo de este proceso, se hace necesario utilizar diferentes procedimientos o instrumentos comúnmente llamados medios de enseñanza, los cuales requieren o no de equipos óptico-mecánicos o eléctricos y resultan de apoyo a la exposición oral. Los medios de enseñanza se agrupan de manera general, en medios de percepción directa, imágenes fijas y en movimiento, sonido, situación real y simulación, así como los apoyados en el uso de las TIC (3).

A nivel mundial, la enseñanza virtual va ganando cada vez más adeptos.

El aprendizaje práctico, experimental y virtual se encontraría obstaculizado en muchos de nuestros centros educacionales ya que dada la situación económica que aún atraviesa el país es difícil proveer a dichos centros con material e instrumental suficiente para desarrollar demostraciones en clases y prácticas de laboratorio en las ciencias naturales, en particular en la Física. Además, en esta asignatura tradicionalmente los estudiantes obtienen bajos resultados académicos, el nivel de motivación por la misma es bajo, por su propia naturaleza requiere de ilustraciones, experimentos, entre otros, que faciliten su comprensión.

En especial el Instituto Pre – Universitario del capitalino municipio de San Miguel del Padrón presenta la siguiente **situación problemática**:

El no poder llevar a cabo demostraciones didácticas por la falta de material o instrumento experimental ha obstaculizado el PEA de los estudiantes de 11no grado en la asignatura de Física.

No se pueden utilizar los instrumentos existentes en el centro por la falta de personal capacitado.

No se pueden explotar los beneficios de las TIC existentes en el centro en la asignatura Física de 11no grado, por no existir un *software* que sirva para apoyar el PEA de los estudiantes.

Por la situación problemática planteada se define como **problema de la investigación**: ¿Cómo apoyar el proceso de enseñanza – aprendizaje de la asignatura Física para estudiantes de 11no grado?

Objeto de Estudio: Sistemas informáticos de apoyo al proceso enseñanza – aprendizaje.

Por todo lo anterior y en respuesta a la pregunta esencial del pasado Congreso Internacional Universidad 2014 de: “Cómo puede la Educación Superior favorecer transformaciones sociales que promuevan el aprendizaje, la justicia, la inclusión, la integración, la lucha contra la pobreza y el desarrollo sostenible”, el **objetivo general** de este trabajo es: Desarrollar un sistema informático para apoyar el proceso de enseñanza

– aprendizaje mediante el componente didáctico medio de enseñanza, en la asignatura Física del 11no grado.

Campo de acción: Sistemas informáticos de apoyo al proceso enseñanza – aprendizaje en la asignatura de Física de 11no grado.

Como **Idea a defender:** Se plantea que con el desarrollo de un sistema informático de Física para estudiantes de 11no grado se puede dotar al Pre-universitario con un medio de enseñanza que posibilite vincular los conocimientos teóricos con la actividad práctica, base de toda formación científica.

Tareas de la Investigación:

- Elaboración del Marco Teórico de la investigación a partir del estado del arte existente actualmente sobre el tema.
- Elaboración del análisis y diseño de la aplicación.
- Implementación de la herramienta informática para apoyar el proceso de enseñanza – aprendizaje de la asignatura Física para estudiantes de la Enseñanza Media Superior.
- Pruebas del sistema informático.

En la investigación se destaca la utilización de los siguientes **métodos teóricos:**

Analítico – sintético: Se emplea en el momento de buscar la esencia del tema en cuestión, también permite realizar análisis de teorías, de documentos, entre otros; realizando una síntesis de la misma posibilitando así encontrar los elementos más importantes que se relacionan con el proceso de diseño de sistemas informáticos.

Análisis histórico – lógico: Utilizado para conocer con mayor profundidad, cómo han evolucionado los sistemas informáticos, con el objetivo de detectar las deficiencias y aspectos positivos a tener en cuenta en la solución que se propone.

Modelación: Utilizado para representar cómo ocurren los procesos que se desean automatizar, además para elaborar los diferentes modelos definidos en la metodología escogida y que sirven de guía durante el diseño de la solución.

Los **métodos empíricos** utilizados para obtener información sobre el objeto de estudio fueron:

Consulta bibliográfica: Permite la elaboración del marco teórico de la investigación fundamentada por la información consultada.

Consulta de especialistas: Permite orientar el trabajo teórico – metodológico en la realización de la investigación.

Observación científica: Utilizado con el objetivo de conocer cómo se comportaba el proceso enseñanza – aprendizaje de la asignatura Física de 11no grado, en los diferentes dominios de aplicación.

Estudio de casos: Utilizado con el objetivo de valorar el resultado científico de las investigaciones, validar los beneficios y la aplicabilidad de la propuesta.

Capítulo 1. Fundamentación teórica de la investigación

Introducción:

En el presente capítulo se precisan un conjunto de elementos que conforman la fundamentación teórica de la investigación. De manera general contiene los principios y conceptos imprescindibles para el desarrollo de un sistema informático para el aprendizaje. Además de describir las principales características de las herramientas existentes respecto a la temática analizada, así como metodología de desarrollo, tecnologías y herramientas utilizadas en el desarrollo de la propuesta de solución.

1.1 El software educativo en el proceso de enseñanza – aprendizaje

Según Rodríguez Lamas el *software* educativo “es una aplicación informática, que, soportada sobre una bien definida estrategia pedagógica, apoya directamente el proceso de enseñanza - aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo” (4).

Labañino César lo define como “una aplicación informática concebida especialmente como medio, integrado al proceso de enseñanza aprendizaje” (4).

De consenso con las citas anteriores el autor de esta tesis considera que, el *software* educativo no es más que un producto o recurso informático diseñado con el propósito de fortalecer el proceso de enseñanza-aprendizaje.

El desarrollo del *software* educativo tiene como base la confección de herramientas que soporten efectivamente el proceso de enseñanza-aprendizaje. Con el uso de las nuevas tecnologías abre nuevas posibilidades de innovación y realización de diferentes modelos pedagógicos que junto con la intrepidez, curiosidad y motivación del maestro para con los estudiantes, se tiende a mejorar y cambiar de una forma positiva el proceso educativo. Varios autores al referirse a las características del *software* educativo coinciden en que (5):

- Son materiales elaborados con una **finalidad didáctica**.

- **Utilizan el ordenador** como soporte en el que los alumnos realizan las actividades que ellos proponen.
- **Son interactivos**, pues contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- **Individualizan el trabajo** de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.
- **Son fáciles de usar.** Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son mínimos, aunque cada programa tiene sus reglas de funcionamiento que son necesarias conocer.

1.2 Herramienta y sistemas informáticos de apoyo a la docencia

En la actualidad son muchas las herramientas y *software* educativo que existen en el mundo, estos brindan una gran ayuda en el proceso enseñanza aprendizaje de los estudiantes.

Durante la investigación se analizaron algunas herramientas dirigidas al proceso de enseñanza - aprendizaje y algunos sistemas informáticos de apoyo a la docencia tanto nacionales como internacionales. A continuación, se mencionan algunas características de estos.

1.2.1 Plataforma ATutor

Es una plataforma de código abierto, basado en la aplicación de gestión de contenidos de aprendizaje que se utiliza para desarrollar y entregar cursos en línea. Donde los usuarios involucrados, los administradores pueden instalar o actualizar ATutor en minutos y desarrollar temas personalizados; los profesores pueden rápidamente ensamblar, empaquetar y redistribuir contenido educativo, y llevar a cabo sus clases en línea; y los estudiantes pueden aprender en un entorno de aprendizaje adaptativo, dinámico y visualmente atractivo (6).

Esta plataforma tiene como ventajas y desventajas las siguientes (7):

Ventajas

- Posee un sistema de correo electrónico propio e interno.
- Es fácil para personas con pocos conocimientos de estas plataformas.
- Cuida la estética de los fondos y fuentes.
- Cursos orientados al autoaprendizaje.
- Administración sencilla.

Desventajas

- Los foros, actividades y recursos están separados.
- La interfaz en la que crea el profesor es diferente a la del alumno.
- No se pueden poner tareas *offline/online*.
- No cuenta con la posibilidad de crear itinerarios de aprendizaje.

1.2.2 Plataforma Moodle

Moodle es una plataforma de aprendizaje diseñada para proporcionarles a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados (8).

Es *software* libre, con Licencia pública GNU. Básicamente, esto significa que los usuarios de Moodle tienen algunas libertades: pueden copiar, usar y modificar Moodle siempre que acepten proporcionar el código fuente a otros, no modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado de él (9).

Características generales de la plataforma Moodle (9)

- Permite la autogestión del tiempo, lo que posibilita que las personas puedan lograr mayor independencia y autonomía.
- Posibilita disponer de recursos didácticos constantemente actualizados, en gran variedad de formatos y a un menor costo.
- Facilita la comunicación bidireccional de los alumnos, tanto con sus profesores como con sus pares, sobre todo fuera del horario de clases.

- Se actualiza muy fácilmente desde una versión anterior a la siguiente, puesto que conserva la misma estructura en la base de datos. Tiene un sistema interno para actualizar y reparar su base de datos cada cierto tiempo.
- Usa solamente una base de datos (si lo necesita puede compartirla con otras aplicaciones).
- Los recursos que el docente entrega a sus estudiantes pueden ser de cualquier fuente y con cualquier formato, puesto que su programación está orientada a objetos. Es decir, soporta objetos como una característica fundamental del mismo, y es necesario tener la fuente del mismo para poder ejecutarlo.
- Ofrece una serie de actividades para los cursos: foros, diarios, diálogos, cuestionarios, consultas, encuestas, tareas, chat, talleres, lecciones, etc.
- Lleva registro y seguimiento completo de los accesos del alumno. Se dispone de informes de actividad de cada estudiante, con gráficos y detalles sobre su paso por cada módulo (último acceso, número de veces que lo ha leído).

Desventajas de la plataforma Moodle (9)

- Algunas actividades pueden ser un poco mecánicas, dependiendo mucho del diseño instruccional.
- Por estar basado en tecnología PHP, la configuración de un servidor con muchos usuarios debe ser cuidadosa para obtener el mejor desempeño.
- Falta mejorar su interfaz de una manera más sencilla.
- Hay desventajas asociadas a la seguridad, dependiendo en dónde se esté alojando la instalación de Moodle y cuáles sean las políticas de seguridad y la infraestructura tecnológica con la cual se cuente durante la instalación.
- Existen también desventajas relacionadas con el soporte técnico. Al ser una plataforma de tecnología abierta y por lo tanto gratuita, no se incluyen servicios gratuitos de soporte por lo que los costos de consultoría y soporte técnico están sujetos a firmas y entidades externas.
- Muestra los mismos contenidos a todos los alumnos. Es decir, no tiene manera de ir acompañando el proceso de aprendizaje de cada alumno, mostrándole sólo las lecciones y actividades, que más se adecuen a su perfil y desempeño. Moodle

debería contar con un módulo que le permita presentar los contenidos de un curso (lecciones, actividades y cuestionarios) según el perfil que tenga cada alumno (por ejemplo: alto, medio o bajo). Ese perfil se determina según criterios de evaluación definidos por el docente y el historial de desempeño del alumno. Esto es necesario porque se considera que los procesos de aprendizaje varían en función de las capacidades de cada alumno. Por ejemplo: un alumno con un buen desempeño seguramente requerirá de un proceso de aprendizaje diferente de aquel que tiene un perfil bajo. Por lo tanto, los contenidos a utilizar por uno deberían ser diferentes de los utilizados por el otro.

A partir de lo anterior, se decide no usar una plataforma de aprendizaje debido a las pocas prestaciones de la infraestructura tecnológica existente en nuestros Institutos Pre Universitarios, la cual dará soporte a la herramienta propuesta. Al igual se pretende incluir en la herramienta propuesta un sistema de recomendaciones basado en casos, el cual recomiende a los estudiantes los ejercicios idóneos para su superación, utilizando las experiencias de los estudiantes para construir la base de casos.

1.2.3 Sistemas informáticos nacionales

Los *softwares* educativos como apoyo a las actividades docentes en la escuela cubana, se ha implementado teniendo en cuenta los avances tecnológicos en este sentido, es decir, a medida que avanza la informatización de la sociedad y, por tanto, la implementación de los recursos informáticos, en las escuelas se introducen beneficiando la calidad de las clases en las aulas.

Estos evidencian un cambio favorable en el Sistema Educativo, pues es una alternativa válida para ofrecer a los alumnos y estudiantes un ambiente propicio para la construcción del conocimiento. En nuestro país se encuentran en práctica varios *softwares* educativos como: Colección Multisaber, Colección El Navegante, Colección Futuro, entre otros.

Colección Multisaber

La colección Multisaber es un grupo de multimedias educativas destinadas a la enseñanza primaria, está compuesta por 32 *software* y tiene 6 módulos: Temas, Ejercicios, Biblioteca, Profesor, Resultados y Juegos. El nombre Multisaber para la colección expresa la idea del aprendizaje de múltiples contenidos para los niños desde temprana edad. Los *softwares* de esta colección poseen una interfaz estandarizada, que proporcionan un ambiente de trabajo amigable e intuitivo con alto nivel de interactividad para hacer más amena la enseñanza y más efectivo el aprendizaje en los estudiantes.

Colección El Navegante

La colección El Navegante instalada en las escuelas secundarias de Cuba cuenta con 10 productos, en cada uno están presentes 6 módulos: Temas, Ejercicios, Juegos, Biblioteca, Registro y Maestro. El módulo Ejercicios permite realizar ejercicios de las tipologías: selección simple, selección múltiple, completamiento por desplazamiento y escritura, verdadero y falso, seleccionar textos, seleccionar palabras, pero no brinda la posibilidad de modificar los existentes ni agregar nuevos. Esta colección fue desarrollada con herramientas propietarias y solo puede ser usada en el sistema operativo *Windows*.

Colección Futuro

Es la colección de *software* educativo que está dedicada a la enseñanza Media Superior, esta colección cuenta con 14 *softwares* educativos: Sunrise, Nuestro Planeta, Mirarte, El arte de las letras, Física más allá, Sophia, El Planeta vivo, Eureka, Un mundo mejor es posible, Pedagogía a tu alcance, ADN, Campo y Sustancia, Redox y Convicciones (10).

En esta colección antes mencionada, se destacan en la presente investigación los *softwares* “Física más allá” o “FisMat” (como también es conocido) y “Campo y Sustancia”, ya que van dirigidos a la asignatura de Física de la enseñanza media superior.

Física más allá es un simulador Físico – Matemático cuya función fundamental es la de elaborar animaciones interactivas de objetos que se desplazan a partir de ecuaciones Matemáticas. Contiene una amplia gama de objetos de carácter multimedia entre los que se encuentran Imágenes, Sonidos y Videos. Permite un amplio trabajo gráfico para el estudio del comportamiento de las variables para cualquier ecuación Matemática, acorde

con los niveles antes mencionados, mediante el empleo de gráficas configurables imprescindibles para la Física y la Matemática (11). Mientras que el *software* Campo y Sustancia cuenta con 10 temas, divididos en subtemas con todo el contenido de Preuniversitario. Este producto permite hacer simulaciones y laboratorios virtuales abordados en los temas de dicha asignatura.

Cubaeduca (<http://www.cubaeduca.cu>) es un portal accesible desde todas las redes nacionales realizado por el Ministerio de Educación con el objetivo de garantizar un espacio para la comunicación y el intercambio de la información mediante el empleo racional y óptimo de las tecnologías para el desarrollo científico y pedagógico.

Objetivos fundamentales:

- Garantizar un espacio para la comunicación y el intercambio de la información mediante el empleo racional y óptimo de las tecnologías de la información y las comunicaciones en el desarrollo científico y pedagógico.
- Brindar a los docentes, profesores en formación, estudiantes y familia, recursos y herramientas para el aprovechamiento pedagógico de las TIC.
- Orientar metodológicamente a los docentes.
- Profundizar en los contenidos de los programas curriculares.

Funcionamiento

El portal hospeda varias páginas del ministerio que facilitan la consulta de contenidos de aprendizaje necesarios en cada uno de los niveles de enseñanza a los usuarios, dígase alumnos, profesores y toda la familia cubana. Además, el sitio contiene un centenar de *softwares* educativos, acceso a una biblioteca digital y un buscador académico.

Este sitio web no solo permite adquirir información, sino que prepara y orienta metodológicamente a los docentes a partir de resultados de investigaciones y de la superación científica. A la vez, los estudiantes pueden acceder con acceso internet.

Después de analizadas las características de cada uno de los *softwares* antes mencionados, se observa que, las Colecciones Multisaber y El Navegante no van dirigidos al nivel de enseñanza en cuestión (nivel medio superior), sino a las enseñanzas primaria y secundaria respectivamente.

Mientras que la Colección Futuro es la que cuenta con *softwares* educativos dedicados a la enseñanza media superior en Cuba. En sentido general tratan aspectos de interés de las ciencias naturales y otras áreas; estos no siguen un orden cronológico y ascendente para la adquisición del conocimiento y los temas que tratan no siguen la secuencia del programa de la asignatura del nivel de enseñanza correspondiente. Esta colección, de los 14 productos que tiene, se pudo ver que cuenta con 2 que están dedicados a la asignatura de Física y pueden apoyar el proceso de enseñanza – aprendizaje de los estudiantes. La dificultad de estos *softwares* es que no son sistemas completos, ya que se enfocan nada más en la parte de simulación virtual de la Física. Donde en el caso de contener ejercicios, cuestionarios, bibliografías y demostraciones virtuales, pudieran brindar un mejor apoyo al estudio de los estudiantes.

Por otra parte, cubaeduca es un portal donde se pueden encontrar todas las asignaturas de todos los niveles de enseñanza de nuestro país, pero este cuenta con varias deficiencias. La primera es que para poder acceder hay que contar con acceso a internet, la información no es actualizada sistemáticamente, es muy estático y poco gestionable.

1.2.4 Sistemas informáticos internacionales

En el ámbito internacional, son muchos los *softwares* educativos existentes que sirven para darle apoyo a la docencia de los estudiantes. Durante la investigación de este trabajo se analizaron algunos de estos sistemas. A continuación, se presentan y se da una breve explicación de estos *softwares*.

Quars - Laboratios de Física

Es un *software* educativo de aprendizaje, repaso y práctica para el área de Física, destinado a la educación media. El mismo cuenta con clases teóricas organizadas por temas y gráficas prácticas, facilitando el aprendizaje de los estudiantes. Cada uno de los temas tratados dispone de introducción, objetivos, conclusiones y en algunos incorpora videos explícitos muy útiles para la Física (12).

ELG Magic

ELG Magic permite realizar cálculos y gráficas de muchos de los problemas relacionados con el tema de cinemática, en total se pueden estudiar hasta cinco movimientos (13):

- Movimiento Rectilíneo Uniforme (M.R.U.)
- Movimiento Rectilíneo Uniformemente Acelerado (M.R.U.A.)
- Tiro Vertical
- Tiro Oblicuo
- Tiro Horizontal

En cada uno de ellos se puede resolver incógnitas, realizar gráficas detalladas y asignar valores a todos los datos implicados en los cálculos (espacio, tiempo, velocidad y altura). El *Software* es gratuito.

GeorK - Física Bachillerato

Es un libro de texto digital que incluye los principales temas de la mecánica clásica tratados en la enseñanza media. Está dividido por temarios en las siguientes unidades o capítulos (14):

- Cinemática del punto en una dimensión.
- Cinemática del punto en dos dimensiones.
- Dinámica de la partícula.
- Trabajo y Energía.
- Conservación de la Energía.

Los contenidos pretenden ser una solución alternativa o complementaria del libro en papel, que permita a los profesores aumentar la calidad de sus clases y elevar el nivel de sus alumnos. Por tanto, la solución contempla unas prestaciones que faciliten el acceso a los contenidos a cualquier alumno y en cualquier escuela, dispongan o no de ordenadores individuales en clase o de conexión a Internet en casa o en la escuela, ya que puede ser usado en cualquier dispositivo capaz de mostrar páginas HTML (computadoras, tabletas o móviles). Este *software* no es libre, ya que para poder hacer uso del mismo se debe pagar una licencia (14).

FreeCAD

FreeCAD es un modelador 3D CAD paramétrico. El desarrollo es completamente de código abierto (LGPL *License*). Está dirigido directamente a la ingeniería mecánica y diseño de productos, pero también cabe en una amplia gama de usos en la ingeniería, como la arquitectura u otras especialidades. Es multiplataforma (15).

Estos *softwares* de forma general están especializados en determinados temas y asignaturas de las Ciencias Naturales. El principal problema radica en que no se rigen por un programa estructurado para la asignatura y no responden a las necesidades de una institución en específico. Están desarrollados como materiales de apoyo en determinado momento del área que tratan, pero no están dirigidos para formar parte del plan de clases de la ciencia que apoyan. Solo Quars - Laboratorios de Física y GeorK - Física Bachillerato contienen clases de determinados temas de la Física, pero no cuentan con ejercicios o cuestionarios para que los estudiantes los resuelvan y puedan comprobar sus conocimientos; además, este último *software* mencionado es privado.

1.3 Sistemas basados en casos

Los Sistemas con Razonamiento Basado en Casos (SRBC) son una de las tecnologías actuales para construir Sistemas Basados en el Conocimiento. En ellos los nuevos problemas se resuelven considerando la solución dada a problemas similares resueltos en el pasado. La arquitectura básica de un SRBC consiste de una base de casos, un procedimiento para buscar casos similares y un procedimiento de adaptación para ajustar las soluciones de los problemas similares a los requerimientos del nuevo problema (16).

Teniendo en cuenta que el sistema a desarrollar debe recomendarle ejercicios al estudiante, luego de resolver los ejercicios que son propuestos por el profesor. Se decide hacer uso de un sistema experto, para que recomiende más ejercicios según las experiencias de los estudiantes en el sistema.

1.4 Metodología de desarrollo de software

En el desarrollo de un *software* se hace necesario el uso de una metodología que represente una guía para el proceso de desarrollo y permita que todas las personas del proyecto trabajen bajo un marco común y una planificación de actividades en base a un conjunto de tareas definidas.

Las metodologías de desarrollo de *software* se clasifican en 2 grupos:

- Metodologías pesadas/tradicional: Orientadas al control de los procesos, establecen rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán.
- Metodologías ligeras/ágiles: Orientadas a la interacción con el cliente y el desarrollo incremental del *software*, mostrando versiones parcialmente funcionales del *software* al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando.

No existe una metodología absoluta para el desarrollo de *software*, algunas se ajustan mejor que otras a las características y necesidades específicas de los proyectos de desarrollo, pero su ausencia en el desarrollo de un proyecto de *software* garantiza con seguridad también la ausencia de calidad. A continuación, se realiza un estudio de algunas metodologías para el desarrollo de *software*.

1.4.1 Proceso Racional Unificado (RUP)

El proceso racional unificado conocido mundialmente como RUP por sus siglas en inglés, es una metodología tradicional, favorece el desarrollo de *software* a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de estándares de calidad. Esta metodología es considerada como un proceso bien definido, estructurado y adaptable a las características y necesidades de cada proyecto en específico. RUP persigue una meta y es la de asegurar la producción de *software* de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

Este proceso se divide en cuatro fases para el desarrollo de *software* (17):

- **Inicio:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

RUP propone un gran cúmulo de documentación de acuerdo al tamaño del proyecto y por consecuencia del equipo de desarrollo, es recomendable utilizarla en el desarrollo de proyectos medianos y grandes.

1.4.2 Programación Extrema (XP)

La programación extrema (del inglés *Extreme Programming* (XP)), es una metodología de desarrollo ágil, una de las más exitosas en tiempo reciente. Su autor principal es Kent Beck, quien eligió algunas características de otras metodologías y las relacionó de forma que cada una complementara a la otra. XP se puede definir como un conjunto de pasos de diversas metodologías, acopladas de manera que sean pasos flexibles a seguir utilizadas con el uso común, para realizar un desarrollo más agradable y sencillo. Esta metodología tiene como base la simplicidad y como objetivo principal la satisfacción del cliente; para lograrlo se deben tomar en cuenta cuatro valores fundamentales (18):

- **Comunicación:** Es muy importante que haya una comunicación constante con el cliente y dentro de todo el equipo de trabajo, de esto dependerá que el desarrollo se lleve a cabo de una manera sencilla, entendible y que se entregue al cliente lo que necesita.
- **Simplicidad:** En la XP se refiere que ante todo y sin importar qué funcionalidad requiera el usuario en su sistema, éste debe ser fácil. El diseño debe ser sencillo y amigable al usuario, el código debe ser simple y entendible, programando sólo lo necesario y lo que se utilizará.
- **Retroalimentación:** Es la comunicación constante entre el desarrollador y el usuario.
- **Coraje:** Se refiere a la valentía que se debe tener al modificar o eliminar el código que se realizó con tanto esfuerzo. También se refiere a tener la persistencia para resolver los errores en la programación.

1.4.3 Selección de la metodología de desarrollo

Luego del análisis realizado entre las dos metodologías antes mencionadas, se selecciona XP para la realización del trabajo, debido a que se adapta en gran medida tanto al tipo de proyecto a desarrollar en este caso pequeño ya que consta de una persona. Además, la Programación Extrema es una metodología ágil centrada en

potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del *software*. Este tipo de método se basa en una realimentación continua entre el cliente y el equipo de desarrollo, con una comunicación fluida entre todos los participantes. También busca la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios.

1.5 Tecnologías y herramientas de desarrollo utilizadas

Existe gran variedad de tecnologías y herramientas usadas en el desarrollo de aplicaciones web. Seguidamente se abarcan las que constituyen la base del desarrollo de la propuesta de esta investigación, donde se pueden encontrar: HTML, CSS, JavaScript, PHP, Laravel, jQuery, Bootstrap y PhpStorm; estableciéndose como imprescindibles para este trabajo.

1.5.1 Tecnologías del lado del cliente

Las tecnologías del lado del cliente son aquellas que pueden ser directamente interpretadas por el navegador y no necesitan un pre tratamiento. El navegador web es también llamado cliente web debido a que realiza tareas de solicitud y consumo de servicios. Se conecta con un servidor al que solicita páginas, el servidor web se las sirve para consumo del cliente, que las muestra a la persona que las ha solicitado. A continuación, se mencionarán las tecnologías del lado del cliente a emplear.

HTML5

Es el lenguaje de marcado de hipertexto predominante para la elaboración de páginas web. Es un lenguaje muy simple que se utiliza para describir la estructura y el contenido de las páginas en forma de texto incluyendo imágenes, animaciones y otros medios soportados. El formato de los documentos se marca mediante etiquetas (tags) que indican el comienzo y el final de los elementos que componen el mismo, teniendo cada uno de ellos un significado estructural diferente (19).

JavaScript

JavaScript es un lenguaje de script multiplataforma, no es necesario compilar los programas para ejecutarlos, los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Es pequeño y ligero, no es útil como un lenguaje independiente, más bien está diseñado

para una fácil incrustación en otros productos y aplicaciones. Es seguro y fiable, ya que los *scripts* tienen capacidades limitadas por razones de seguridad, es ejecutado en el cliente, por lo que el servidor no es utilizado más de lo debido. Es dinámico, responde a eventos en tiempo real (20).

CSS3

Hoja de estilo en cascada (CSS por sus siglas en inglés), es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (21).

El empleo de hojas de estilo CSS es una práctica que aporta una serie de beneficios como:

- Aumento de la legibilidad y reducción del peso de las páginas Web.
- Mejora del mantenimiento y actualización de los sitios Web.
- Mejora de la accesibilidad: se pueden definir hojas de estilo locales en función de las necesidades o preferencias del usuario.
- Versatilidad: Se ofrecen diferentes hojas de estilo para los diferentes tipos de medio existentes (sintetizadores de voz, dispositivos braille, pantallas de computador a color, televisión).

1.5.2 Tecnología del lado del servidor

Las tecnologías del lado servidor son aquellas reconocidas, ejecutadas e interpretadas por el propio servidor y se envían al cliente en un formato comprensible por él. El uso de estas tecnologías ofrece un entorno rápido de creación de scripts y soporte para los estándares más importantes. Un ejemplo de esto es el lenguaje PHP.

PHP

PHP se basa en una sintaxis parecida a los lenguajes C, al Java y Perl, con algunos otros cambios. Hace posible el desarrollo de páginas web que se generen dinámicamente y de forma rápida, las cuales pueden ejecutarse en distintos tipos de servidores web. Fue uno de los primeros lenguajes de programación del lado del servidor, el cual se podía incorporar directamente en el documento HTML en lugar de llamar a un archivo externo

que procese los datos. Debido a que es de código abierto, se tiene un fácil acceso a él, ventaja que tiene sobre el *software* privativo. Es considerado fácil de asimilar y a partir de las últimas versiones pasó a ser un orientado a objetos y compatible con la mayoría de los gestores de bases de datos que se utilizan hoy en día.

PHP posee un número importante de ventajas:

- Es multiplataforma.
- Gratuito y con documentación muy amplia en internet, donde se puede encontrar tutoriales y guías gratuitas acerca de cómo utilizarlo.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es un lenguaje de una sintaxis muy simple, por lo cual se deduce que es fácil de aprender, además, posee una gran variedad de funciones que pueden ser utilizadas para mejorar el rendimiento de las aplicaciones.
- El código fuente escrito en PHP no se puede visualizar en el navegador web, ya que es en el servidor donde se ejecuta el código y envía su resultado HTML al navegador.
- Capacidad de expandir su potencial utilizando módulos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables, aunque las mismas se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Posee una amplia comunidad de desarrolladores que usan este lenguaje y colaboran con el mejoramiento del motor de PHP, por lo cual es cada vez más seguro y estable a medida que pasa el tiempo y aumentan sus versiones.

1.5.3 Marco de trabajo para el desarrollo

Un marco de trabajo (del inglés *frameworks*) en lo que refiere a desarrollo web, se trata de un conjunto de procesos, técnicas y archivos previamente confeccionados, que facilitan y aceleran la producción de sitios y aplicaciones web (22).

Bootstrap

Bootstrap es un *framework* que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. La mayor ventaja es que se pueden crear interfaces que se adapten a los distintos navegadores apoyándonos en un *framework* potente con numerosos componentes web que ahorrarán mucho esfuerzo y tiempo (23).

Características principales de Bootstrap

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como *tablets* y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo, JQuery.
- Ofrece un diseño sólido usando estándares como CSS3/HTML5.
- Es un *framework* ligero que se integra de forma limpia en los proyectos.

jQuery

JQuery es una biblioteca o *framework* de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM (Document Object Model), manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX (acrónimo de Asynchronous JavaScript + XML, que se puede traducir al español como "JavaScript asíncrono + XML") a páginas web. JQuery es *software* libre y de código abierto, posee un doble licenciamiento bajo la licencia MIT y de la GNU General Public License, Versión 2.1. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (24).

Yii

Yii es un *framework* para PHP de alto rendimiento basado en componentes web para desarrollar aplicaciones de gran escala. Permite una máxima reusabilidad en la programación web y puede acelerar significativamente el proceso de desarrollo.

Algunas de sus características se mencionan a continuación:

- Fácil de instalar.
- Patrón de diseño Modelo Vista Controlador (MVC).
- Es multiplataforma.
- Liviano de correr y equipado con soluciones de cacheo sofisticadas.
- Permite máxima reutilización.

Laravel

Laravel es un *framework* de código abierto para desarrollar aplicaciones y servicios web con PHP. Su filosofía es desarrollar código PHP de forma elegante y simple. Fue creado en 2011 y tiene una gran influencia de *frameworks* como *Ruby on Rails*, *Sinatra* y *ASP.NET*, con las características (25):

- Está diseñado para desarrollar bajo el patrón Modelo Vista Controlador.
- Posee un sistema de mapeo de datos relacional llamado *Eloquent ORM*.
- Utiliza un sistema de procesamiento de plantillas llamado *Blade*, el cual hace uso de la caché para darle mayor velocidad.
- Integración con jQuery y Bootstrap.
- Permite máxima reutilización.
- Sistema de ruteo.
- Basado en Composer.
- Usa componentes de Symfony.
- Es multiplataforma.

Selección del *framework* para la manipulación de PHP

Se elige como marco de trabajo a Laravel debido a que destaca entre los *frameworks* PHP, por su eficiencia y su rica librería de funcionalidades, así como también su clara documentación. Es multiplataforma y aunque es un *framework* que tiene muy poco tiempo de creado, ya tiene mucho auge en el mundo. La curva de aprendizaje es baja, lo cual permite un mejor trabajo en el desarrollo de las aplicaciones web. Es sencillo de instalar y permite una máxima reutilización. Como la mayoría de los *framework* para PHP, este sigue el patrón de diseño MVC.

1.5.4 Herramienta CASE

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *Software*. Como es conocido, los estados en el ciclo de vida de desarrollo de un *Software* son: Investigación Preliminar, Análisis, Diseño, Implementación, Prueba e Instalación básicamente.

Ejemplos de herramientas CASE se puede mencionar *Rational Rose* y *Visual Paradigm*, este último es utilizado para el desarrollo del presente trabajo por ser multiplataforma y por todas las características que tiene.

Visual Paradigm v8.0:

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (26).

1.5.5 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (en inglés *Integrated Development Environment* o IDE) es una aplicación de *software* que proporciona servicios integrales a los programadores de *software*. Un IDE normalmente está compuesto por:

- Un editor de código fuente.
- Un compilador y / o un intérprete.
- Automatización de generación de herramientas.

- Un depurador.

Pueden ser aplicaciones que funcionan por sí solas o pueden ser parte de aplicaciones existentes y poseen entre sus características más importantes, el poder compilar más de un lenguaje de programación. Para el desarrollo de la propuesta de solución se hará uso del IDE PhpStorm.

PhpStorm

Este editor de código provee excelente soporte en diferentes tipos de lenguaje de programación, los cuales son PHP, HTML, JAVASCRIPT, CSS, SASS, LESS y muchos más. Brinda grandes herramientas las cuales permiten cambiar y saltar desde el proyecto a cualquier archivo, clase o símbolo. PhpStorm entiende donde deseas ir y te lleva al lugar donde quieras de manera instantánea. La mayoría de los PHP *Frameworks* son soportados por PhpStorm como lo son Laravel, Symfony, Drupal, Magento, CakePHP, Zend Framework, Bootstrap y otros. Posee muchas características innovadoras y funciones que facilitaran realizar proyectos de manera más ágil y eficiente.

Algunas de las principales características que cuenta PhpStorm son:

- Editor de código inteligente PHP.
- Calidad de análisis en el código.
- Entorno de desarrollo.
- Editor HTML/CSS/JavaScript.
- Depuración y pruebas.
- Experiencia multiplataforma

1.5.6 Sistema gestor de base de datos

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: *DataBase Management System*) es un sistema de *software* que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos (27).

PostgreSQL v9.4

Como sistema gestor de base de datos se utilizará PostgreSQL, este es un gestor de base de datos relacional, libre y orientado a objetos. Funciona en todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Además, brinda una amplia documentación lo que hace más fácil trabajar con el mismo (28).

PostgreSQL es compatible con el almacenamiento de objetos binarios, incluyendo imágenes, sonidos y videos. Tiene interfaces de programación con diferentes lenguajes como PHP, C, C++, Java, .NET, Perl, Python, entre otros. Es muy eficiente cuando se desea almacenar un gran número de datos.

1.5.7 Servidor web

Un servidor web es un programa que implementa el protocolo HTTP (del inglés *Hypertext Transfer Protocol*). Este protocolo está diseñado para transferir los llamados hipertextos, páginas web o paginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproducciones de sonidos (29).

XAMPP

XAMPP, es un servidor de plataforma libre, que integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje PHP y Perl, un servidor de base de datos MySQL y un servidor de FTP FileZilla. Es multiplataforma (Linux, Windows, MAC o Solaris) (30).

1.6 Conclusiones parciales

El análisis del *software* educativo permitió determinar que ninguno soluciona totalmente la problemática planteada, se analizaron las potencialidades que pudiesen ser incorporadas a la solución y se detectaron las tendencias actuales en el desarrollo de este tipo de *software*. Se realizó un estudio del marco teórico en el que se fundamenta el desarrollo de la plataforma informática propuesta, además de las herramientas y tecnologías que se utilizarán, quedando escogido *Visual Paradigm* para modelar el sistema, el uso de la metodología de desarrollo de *software* XP, PostgreSQL como gestor de bases de datos, el lenguaje de programación PHP del lado del servidor y del lado del cliente *JavaScript*, *HTML* y *CSS*. También se utilizó *PhpStorm* para la gestión de código fuente.

Capítulo 2. Sistema Informático de Apoyo al Aprendizaje en la Asignatura Física (SIAF)

Introducción

El presente capítulo estará dirigido a la elaboración de la propuesta de diseño del sistema a desarrollar. Se profundizará en cada una de las etapas de Exploración, Planificación y Diseño, propias de la metodología XP que es utilizada. En la primera etapa se definirán las historias de usuario las cuales describirán las funcionalidades que brindará el sistema. En la fase que le sigue se describen los requisitos funcionales y no funcionales que deberá cumplir la solución, además se especifica el plan de iteraciones y el plan de entrega. En la etapa de diseño se utilizan técnicas como las tarjetas CRC (clase, responsabilidad y colaboración) para diseñar las aplicaciones.

2.1 Descripción de la solución propuesta

En este epígrafe de la presente investigación, se dará una breve descripción de lo que tendrá la solución propuesta y la explicación del sistema basado en casos (SBC) para recomendar ejercicios a los estudiantes.

2.1.1 Sistema Informático SIAF

SIAF tiene como principal objetivo apoyar el proceso docente educativo en la asignatura de Física en 11no grado, buscando elevar la comprensión por parte de los educandos y ayudando al profesor en cuanto al criterio de evaluación.

Dicho sistema contará con varias funcionalidades para que los estudiantes puedan apoyarse a la hora de estudiar algún contenido que se encuentre en el sistema, dígame resolver cuestionarios, ejercicios propuestos, consultar bibliografías y realizar demostraciones virtuales.

Las funcionalidades con las que contará el sistema serán gestionadas por los profesores, estos serán los que insertaran todos los ejercicios, que deben de ser de tipo selección y texto. Donde los de selección son para que los estudiantes puedan escoger una respuesta según la descripción del ejercicio y el sistema dará una nota. Los de tipo texto son para que los estudiantes puedan escribir la respuesta y enviarla mediante una

notificación que le llegará al profesor para que pueda revisarlo y emitir una nota. En el caso de los cuestionarios, los profesores podrán crearlos por subtemas con todos los ejercicios de tipo selección con que cuente, estos cuestionarios servirán para la consolidación del estudio en cada subtema de la asignatura. También se debe gestionar bibliografías que serán organizadas por subtemas para que a la hora de estudiar puedan ser consultadas fácilmente.

SIAF incluye además un SBC, el cual se encarga de recomendar a los estudiantes qué ejercicios podrían resolver según su desempeño en los problemas antes resueltos, propuestos por el profesor. Constituyendo esto un aporte significativo y novedoso en los sistemas de apoyo al proceso de enseñanza-aprendizaje, pues, según los resultados del estudio realizado, no existe un precedente que logre vincular todos los elementos antes descritos.

2.1.2 Sistema experto basado en casos para recomendar ejercicios a los estudiantes

Dicho sistema va a ser capaz de recomendar a los estudiantes cuales son los ejercicios que debe realizar para lograr la ejercitación de los contenidos según sus diferentes ritmos de aprendizaje.

El mismo va a partir de un grupo de ejercicios que el profesor propone realizar en determinada temática. Los cuales formarán parte de la base de casos que va a estar compuesta por, una relación entre los ejercicios, el estudiante y la propuesta del profesor.

- **Módulo recuperado de casos**

Este módulo se va a encargar de extraer de la base de casos, los casos correspondientes al estudiante actual. Cada tupla de dicha tabla o base de casos va a estar compuesta por:

| | | | |
|------------------------|--------------------------------------|----------------------------------|------------------------|
| id_caso: <i>int</i> | estudiante_id_usuario: <i>string</i> | ejercicio_id_recurso: <i>int</i> | propuesta: <i>bool</i> |
|------------------------|--------------------------------------|----------------------------------|------------------------|

Donde propuesta será el *rasgo objetivo*.

- **Módulo adaptación**

La adaptación es el proceso de ajustar los casos recuperados a una solución, conociendo las restricciones impuestas por el problema a resolver. En este caso no es necesario adaptar los casos recuperados ya que se utiliza un $k \times n \times n$, con $k=1$, seleccionando los más semejantes, para entre ellos recomendar el más semejante.

- **Módulo de evaluación de soluciones**

Este módulo se va a encargar de comparar los casos, con los ejercicios candidatos haciendo uso de la función de semejanza, las funciones de comparación y los rasgos predictores siguientes:

El conjunto de rasgos predictores va a estar formado por:

1. **Temática:** Es una característica primordial en los ejercicios, en el momento de comparar la similitud entre dos ejercicios, es de vital importancia que pertenezcan a una misma temática.

Factor de peso: $\pi_i = 0.3$

Función de comparación:

$$\partial_i(O, O_t) = \begin{cases} 1 & \text{si } x_i(O) = x_i(O_t) \\ 0 & \text{e. o. c.} \end{cases}$$

2. **Complejidad:** Otra característica esencial para comparar, siendo este el más complejo a la hora de determinar su similitud. La complejidad va a estar condicionada por el resultado obtenido en el caso específico en cual se está comparando. Si el ejercicio propuesto por el profesor, el estudiante lo realizó de forma satisfactoria las complejidades de los ejercicios a recomendar debe de ser igual o mayor, en caso contrario debe tener una complejidad menor o igual.

Factor de peso: $\pi_i = 0.3$

Función de comparación:

$$\partial_i(O, O_t) = \begin{cases} 1 & \text{si } (r = 1 \ \&\& \ x_i(O) - x_i(O_t) = 1) \ || \ (r = 0 \ \&\& \ x_i(O_t) - x_i(O) = 1) \\ 0.9 & \text{si } (r = 0 \ \&\& \ x_i(O) = x_i(O_t)) \\ 0.66 & \text{si } (r = 1 \ \&\& \ x_i(O_t) = x_i(O)) \\ 0.33 & \text{si } (r = 1 \ \&\& \ x_i(O_t) - x_i(O) = 1) \\ 0 & \text{e. o. c.} \end{cases}$$

r: tomará valor 1 si el estudiante aprobó el ejercicio, 0 en caso contrario.

- 3. Efectividad:** Esta característica permite a la hora de recomendar el ejercicio, saber el grado de asimilación que está teniendo el mismo entre los compañeros del estudiante actual. Un ejercicio que tiene una efectividad muy baja entre los estudiantes no sería ideal su recomendación del mismo modo que un ejercicio, que los estudiantes resuelvan de manera sencilla. Por lo cual se recomienda establecer un intervalo que permita determinar cuáles son los ejercicios que han tenido una efectividad razonable para ser recomendados.

Factor de peso: $p_i = 0.2$

Función de comparación:

$$\delta_i(O, O_t) = \begin{cases} 1 & \text{si } |x_i(O) - x_i(O_t)| < \varepsilon \\ 0 & \text{e. o. c.} \end{cases}$$

ε : determina el margen de diferencia entre efectividades.

- 4. Profesor:** Mediante este rasgo determinamos si existe semejanza respecto al profesor que introdujo el ejercicio en el sistema.

Factor de peso: $p_i = 0.1$

Función de comparación:

$$\delta_i(O, O_t) = \begin{cases} 1 & \text{si } x_i(O) = x_i(O_t) \\ 0 & \text{e. o. c.} \end{cases}$$

- 5. Tipo de respuesta:** Rasgo que determina la forma en que el estudiante va a dar solución al ejercicio.

Factor de peso: $p_i = 0.1$

Función de comparación:

$$\delta_i(O, O_t) = \begin{cases} 1 & \text{si } x_i(O) = x_i(O_t) \\ 0 & \text{e. o. c.} \end{cases}$$

Por último, la función de semejanza va a estar dada por:

$$\beta(O, O_t) = \sum_{i=1}^n p_i \times \delta_i(O, O_t)$$

Donde:

- **n**: Es el número de rasgos predictores.
- **p_i**: Peso del rasgo i, donde $i = 1, \dots, n$, $\sum_{i=1}^n p_i = 1$ cumpliendo con la condición de completitud de los pesos.
- $\partial_i(O, O_t)$: **Función de comparación** para el rasgo i.

- **Módulo almacenamiento**

Este módulo lleva a cabo la actualización de la base de casos y a su vez el encargado de garantizar el “aprendizaje” del Sistema Experto, añadiendo nuevos casos o modificando los existentes. Esto se logra cada vez que el profesor proponga nuevos ejercicios para determinada temática y cuando el estudiante resuelva los ejercicios que fueron recomendados por el sistema.

El mismo tomará los casos más semejantes a los propuestos por el profesor y en concordancia con las complejidades según lo demostrado por el estudiante, propondrá la ejercitación en los problemas idóneos para su superación.

2.2 Exploración

En esta fase, es donde se plantean las historias de usuario, las mismas son uno de los artefactos más importantes que se generan en la metodología, donde los clientes definen cuales son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (31).

2.2.1 Historias de Usuario

Las historias de usuario (HU) son descripciones realizadas por el cliente, cortas y con un nivel de detalle mínimo de los requerimientos de un sistema como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Permiten responder rápidamente a los requerimientos cambiantes y aunque

se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas.

Las HU se representan mediante tablas las cuales contienen las siguientes secciones:

- **Nombre:** Nombre que identifica la HU.
- **Prioridad:** Esta característica es dada por el cliente con los valores: alta, media o baja en dependencia de la importancia en que desean ser implementadas.
- **Riesgo en desarrollo:** El riesgo que puede tener durante la fase de desarrollo: alto, medio o bajo según el criterio de los desarrolladores.
- **Iteración asignada:** Número de la iteración en la cual se desarrollará la HU.
- **Puntos estimados:** Tiempo estimado en semanas que se le asignará.
- **Descripción:** Breve descripción del proceso que define la HU.
- **Observaciones:** Alguna acotación importante de señalar acerca de la HU.

En el presente trabajo se obtuvieron un total de 27 HU, las cuales se realizarán en 3 iteraciones. A continuación, se describen 3 de estas y en el Anexo 1 se pueden ver otras 5.

| Historia de Usuario | |
|--|---|
| Número: 2 | Nombre de Historia de Usuario: Crear usuario |
| Usuario: profesor | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 0.6 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 0.6 |
| <p>Descripción: El sistema debe permitir al administrador y profesores crear usuarios en el sistema, dándole los privilegios que le corresponda al usuario creado. El administrador podrá crear profesores y estudiantes mediante un formulario que contenga los siguientes campos: Nombre y Apellidos, Usuario, Rol, Contraseña y Repetir Contraseña. El profesor solo podrá crear los estudiantes que vaya a atender en el sistema, mediante un formulario similar al del administrador, pero solo debe salir el Rol de estudiante y los grupos que el profesor atiende. Luego de llenados todos los campos debe dar clic en el</p> | |

botón “**Crear**” y automáticamente el sistema crea el usuario y lo guarda en la base de datos. En caso de dejar algún campo vacío en el formulario o introducir un usuario existente, el sistema debe mostrar un mensaje de error informando en que campo se cometió el mismo y se mantiene en el formulario.

Observaciones:

- El administrador o profesor debe de estar autenticado para poder crear usuarios.

Tabla 1: HU2 Crear usuario.

| Historia de Usuario | |
|---|--|
| Número: 20 | Nombre de Historia de Usuario: Evaluar ejercicios |
| Usuario: estudiante y profesor | Iteración Asignada: 2 |
| Prioridad en Negocio: Media | Puntos Estimados: 1.2 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1.2 |
| <p>Descripción: Los ejercicios de tipo selección los estudiantes podrán resolverlos seleccionando una respuesta y el sistema debe darle una nota luego de hacer clic en el botón “Evaluar”. Mientras que los ejercicios de tipo texto, le saldrá la descripción y un campo para pueda escribir la respuesta y un botón “Enviar”, para que le llegue la respuesta al profesor en una notificación donde pueda revisarlo y dar una nota. En caso de que el estudiante no marque una respuesta en el de selección o no escriba en el de texto, el sistema le mostrará un mensaje informándole que no puede resolver el ejercicio sin dar una respuesta.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> • El estudiante debe estar autenticado en el sistema para poder resolver ejercicios. | |

Tabla 2: HU20 Evaluar ejercicios.

| Historia de Usuario | |
|---|---|
| Número: 23 | Nombre de Historia de Usuario: Recomendar ejercicios |
| Usuario: estudiante | Iteración Asignada: 3 |
| Prioridad en Negocio: Baja | Puntos Estimados: 2.8 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 2.8 |
| <p>Descripción: El sistema debe ser capaz de recomendar ejercicios a los estudiantes según los resultados obtenidos en los ejercicios que se encuentran en una base de casos. Luego que el estudiante resuelva los ejercicios propuestos por el profesor estos estarán en una base de casos y el sistema debe de recomendarle ejercicios al estudiante según el tema que. En caso de no tener ejercicios resueltos en el tema seleccionado, el sistema debe mostrar un mensaje informándole que no hay ejercicios para recomendarle.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> • El estudiante debe de estar autenticado para que el sistema pueda recomendarle ejercicios. • Si el estudiante no tiene ejercicios resueltos en la base de casos, el sistema no podrá recomendarle ejercicios. | |

Tabla 3: HU23 Recomendar ejercicios.

2.3 Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente (31).

En la lista de reservas del producto se define como fueron estimadas cada historia de usuario y la planificación de los requisitos no funcionales del producto.

2.3.1 Lista de reservas del producto

La lista de reserva del producto muestra las tablas con los requisitos funcionales y no funcionales de la aplicación (RNF).

Los requisitos funcionales establecen los comportamientos del sistema, cada uno puesto en un orden de prioridad con su estimación en semanas y el rol que realiza esta estimación. Donde cada requisito lo máximo que puede estar estimado es 3 semanas, 1 día estimado equivale a 0.2 y cada semana valorada en 5 días de trabajo. Mientras que los requisitos no funcionales, describen las cualidades o propiedades que el sistema debe tener. A continuación, se muestran ambas tablas.

| Número | Descripción | Estimación | Estimado por |
|--------------------------|--|------------|--------------|
| Prioridad: Alta. | | | |
| 1 | Autenticar usuario. | 0.6 | Analista. |
| 2, 3, 4 | Crear/ Modificar/ Eliminar usuario. | 1.8 | Analista. |
| 5, 6, 7 | Crear/ Modificar/ Eliminar grupo. | 1.8 | Analista. |
| 8, 9, 10 | Crear/ Modificar/ Eliminar tema. | 1.8 | Analista. |
| 11, 12, 13 | Crear/ Modificar/ Eliminar subtema. | 1.8 | Analista. |
| Prioridad: Media. | | | |
| 14, 15, 16 | Crear/ Modificar/ Eliminar recurso. | 2 | Analista. |
| 17,18,19 | Crear/ Modificar/ Eliminar cuestionario. | 1.8 | Analista. |
| 20 | Evaluar ejercicios. | 1 | Analista. |
| 21 | Evaluar cuestionarios. | 1 | Analista. |
| Prioridad: Baja. | | | |
| 22 | Proponer ejercicios. | 1 | Analista. |
| 23 | Recomendar ejercicios. | 2.8 | Analista. |
| 24, 25 | Crear/ Eliminar notificación. | 1.2 | Analista. |
| 26 | Mostrar notas de ejercicio. | 0.6 | Analista. |
| 27 | Mostrar notas de cuestionario | 0.6 | Analista. |

Tabla 4: Requisitos Funcionales.

Requisitos No Funcionales:

| # | Descripción | Tipo |
|---|---|--------------------------------------|
| 1 | <ul style="list-style-type: none"> Se necesitará un nivel bajo o medio en computación, ya que el manejo de la aplicación será sencillo, permitiendo al usuario una fácil comprensión y trabajo con el sistema. El diseño debe permitir al usuario navegar con facilidad por la aplicación. | Usabilidad |
| 2 | <ul style="list-style-type: none"> Las interfaces de la aplicación contarán con los componentes visuales necesarios para las operaciones correspondientes, evitando la sobrecarga de imágenes. El sistema debe mostrar una interfaz profesional, sin información repetida, ni en exceso. | Apariencia o interfaz externa |
| 3 | <ul style="list-style-type: none"> Lenguaje de programación a utilizar PHP, <i>framework</i> de desarrollo Laravel y como IDE PhpStorm. | Restricciones de diseño |
| 4 | <ul style="list-style-type: none"> El sistema debe soportar un tiempo de respuesta menor o igual a 5 segundos. | Eficiencia |
| 5 | <ul style="list-style-type: none"> La seguridad de la base de datos estará a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información. Políticas de seguridad por usuarios y roles: el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado. | Seguridad |
| 6 | <p>Para el cliente:</p> <ul style="list-style-type: none"> Navegador web: Mozilla Firefox (30 o superior) o Google Chrome (25 o superior). Sistema operativo Windows o GNU/Linux. | Restricciones de Software |

| | | |
|---|---|----------------------------------|
| | <p>Para el servidor:</p> <ul style="list-style-type: none"> • Servidor de base de datos PostgreSQL. • Servidor con módulo PHP disponible (XAMPP, WAMP u otros). | |
| 7 | <p>Para el cliente:</p> <ul style="list-style-type: none"> • Memoria RAM: Mínimo 512 MB. • Procesador: Intel Core 2 Quad con 1.6 GHz de velocidad de procesamiento o superior. <p>Para el servidor:</p> <ul style="list-style-type: none"> • Memoria RAM: Mínimo recomendada 2 GB. • Procesador: Recomendado Intel i3 con 2.1 GHz de velocidad de procesamiento o superior. • Tarjeta de Red: Ethernet a 100 Mbps o superior. • Capacidad de almacenamiento en disco duro: Mínimo 100 GB. | Restricciones de Hardware |

Tabla 5: Requisitos no funcionales.

2.3.2 Plan de Iteraciones

Luego de identificar y redactar cada una de las HU y de la estimación del esfuerzo necesario para realizarlas, se debe conformar el plan de iteraciones. Este plan muestra en que iteración serán implementadas las historias de usuarios definidas para el desarrollo del sistema. Para la solución que se propone se cuentan con 3 iteraciones correspondientes a la implementación de las historias de usuario de prioridad Alta, Media y Baja.

A continuación, se muestra el plan con la descripción de cada iteración, la cantidad de HU y la duración en semanas.

| Iteración | Descripción | Cantidad de HU por iteración | Duración total |
|-----------|---|------------------------------|----------------|
| 1 | Las HU que tienen prioridad de negocio <i>Alta</i> se implementarán en la primera iteración, ya que estas funcionalidades son fundamentales para cubrir las necesidades del cliente. Además son imprescindibles para lograr una buena base, funcionamiento y desarrollo de la aplicación. | 1 – 13 | 7.8 |
| 2 | Las HU de prioridad de negocio <i>Media</i> se implementarán en la segunda iteración, ya que complementan a las HU de la iteración anterior aunque son de gran importancia en el sistema a desarrollar. | 14 – 21 | 5.8 |
| 3 | Las HU de prioridad de negocio <i>Baja</i> se desarrollarán en la tercera iteración, las cuales son tan importantes como las citadas anteriormente, pero debido a la cantidad de HU y que dependen de las anteriores, se decidió realizarlas en la última iteración. | 22 – 27 | 6.2 |

Tabla 6: Plan de iteraciones.

2.3.3 Plan de Entrega

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes u otros) (32).

A partir del plan de iteraciones definido, se procede a realizar el plan de entrega, el cual tiene como objetivo dar las fechas aproximadas de cada iteración con la cantidad de HU que cuenta. A continuación, se muestra el plan de entrega.

| Iteración | 1 | 2 | 3 |
|------------------|------------|------------|-----------|
| Cantidad de HU | 13 | 8 | 6 |
| Fecha de Entrega | 20-02-2016 | 29-04-2016 | 6-06-2016 |

Tabla 7: Plan de entregas.

2.4 Fase de diseño

La metodología XP hace especial énfasis en los diseños simples y claros, procurando hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar.

2.4.1 Patrón de arquitectura Modelo Vista Controlador

El patrón de arquitectura modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos (33):

- El modelo (la información de datos) contiene únicamente los datos puros de aplicación; no contiene la lógica que describe cómo pueden presentarse los datos a un usuario.
- La vista (la información de presentación) presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos.
- El controlador (la información de control) está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista.

El uso del patrón MVC ofrece para el desarrollo de las aplicaciones múltiples ventajas, ya que define una separación entre datos y presentación, presenta gran sencillez para crear distintas representaciones de los mismos datos, permite a los desarrolladores la

reutilización de los componentes, la tarea de mantenimiento de los sistemas se simplifica grandemente, brinda a los desarrolladores la facilidad de desarrollar prototipos rápidos y aplicaciones escalables.

Para el desarrollo del sistema propuesto se utiliza el patrón MVC ya que el *framework* Laravel basa su funcionamiento en este patrón de arquitectura. A continuación, se detalla cómo se emplea este patrón en la solución propuesta:

Modelo

Laravel incluye un sistema de mapeo de datos relacional llamado *Eloquent* ORM que facilita la creación de modelos y su funcionamiento es muy sencillo. Todos los modelos al crearlos usando *Eloquent* ORM deben extender de *Model*, que es la carpeta que contiene todos los modelos del sistema.

Vista

Laravel incluye 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, permite una sintaxis mucho más reducida en su escritura.

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 sólo 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.

Controlador

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 *Controller*.

2.4.2 Patrones de diseño

Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son basadas en la experiencia y que se ha demostrado que

funcionan. Es evidente que a lo largo de multitud de diseños de aplicaciones hay problemas que se repiten o que son análogos, es decir, que responden a un cierto patrón. Una vez entendido el funcionamiento de los patrones de diseño, estos serán mucho más flexibles, modulares y reutilizables (34).

Patrones GRASP

Los Patrones de *Software* para la Asignación General de Responsabilidad (GRASP por sus siglas en inglés), el nombre fue elegido para sugerir la importancia de comprender los principios de diseñar con éxito los *softwares* orientados a objetos (34). Estos ayudan a capturar conocimiento y a crear un vocabulario técnico, hacen el diseño orientado a objetos más flexible, elegante y en algunos casos reutilizables. Los patrones que se utilizaron en el sistema son los siguientes:

- **Experto:** asignación de una responsabilidad a la clase que cuenta con la información necesaria para cumplirla (experto de información). Este patrón se puede evidenciar ya que garantiza que cada clase del sistema asume las responsabilidades que le corresponden, según las funcionalidades que se quieren implementar y a partir de la información que posee; por lo que cada clase contendrá la información necesaria para cumplir su responsabilidad.
- **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos. Su propósito fundamental es encontrar un creador que debe ser conectado con el objeto producido en cualquier evento.
- **Bajo acoplamiento:** El bajo acoplamiento soporta un diseño de clases más independientes, que reduzca el impacto de los cambios en cada una de ellas y la reutilización de las mismas, aumentando la productividad. Para el desarrollo de la solución propuesta, para alcanzar un bajo acoplamiento las clases controladoras heredan únicamente de la clase *Controller*.
- **Alta cohesión:** una clase con alta cohesión posee un número relativamente pequeño de responsabilidades, relacionadas entre sí por sus funcionalidades. Dicha clase colabora con otras clases para compartir el esfuerzo si la tarea es grande.

- **Controlador:** ofrece una guía para tomar decisiones apropiadas en la elección de los controladores de eventos. Su utilización propicia que las operaciones del sistema se manejen en la capa de dominio de los objetos, y no en la de presentación. Este patrón se evidencia en las clases *Controller*.

Estos patrones se aplicaron de una forma u otra en la codificación de la aplicación a desarrollar a partir de los requerimientos de la arquitectura definida, logrando un código legible y fácil de modificar en caso de ser necesario sin la necesidad de hacer grandes cambios.

2.4.3 Tarjetas CRC

Las tarjetas de clase - responsabilidad - colaboración (CRC) se utilizan para el desarrollo y mantenimiento de la configuración de productos de sistemas. Las tarjetas CRC se introdujeron como técnica informal y de fácil uso para la enseñanza del modelado orientado a objetos. Estas tarjetas se aplican a menudo en las primeras fases de un proyecto de desarrollo de *software* para llegar a alternativas de diseño. Las tarjetas CRC representan objetos y se describen a partir de los siguientes elementos.

- **Clase:** Nombre de la clase a la cual pertenece la tarjeta.
- **Responsabilidad:** Describe cuales son las funcionalidades que deben ser implementadas por la clase.
- **Colaboración:** Enumera las diferentes clases con las cuales tiene relación la clase a la cual pertenece la tarjeta CRC.

A continuación, se muestran 3 tarjetas, en el anexo 2 se muestran otras 5.

| Tarjeta CRC | |
|---|------------------------------|
| Clase: UsuarioController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • protected function createUser (): Método encargado de que se puedan crear usuarios en el sistema. | Estudiante. Profesor. |

| | |
|---|--|
| <ul style="list-style-type: none"> • public function updateUser(): Método encargado de poder actualizar usuarios en el sistema. • public function destroyUser(): Método encargado de poder eliminar usuarios en el sistema. | |
|---|--|

Tabla 8: Tarjeta CRC UsuarioController.

| Tarjeta CRC | |
|---|---|
| Clase: RecursoController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • public function createRecurso(): Método encargado de crear recursos en el sistema, del tipo: bibliografía, demostración, ejercicio y gráfico. • public function updateRecurso(): Método encargado de actualizar los recursos que fueron gestionados por el profesor. • public function destroyRecurso(): Método encargado de eliminar recursos por el profesor. | <p>Profesor.</p> <p>Tema.</p> <p>Subtema.</p> |

Tabla 9: Tarjeta CRC RecursoController.

| Tarjeta CRC | |
|---|---|
| Clase: CuestionarioController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • public function createCuestionario(): Método encargado de poder crear cuestionarios por el profesor. • public function updateCuestionario(): Método encargado de poder actualizar cuestionarios por el profesor. • public function destroyCuestionario(): Método encargado de poder eliminar cuestionarios por el profesor. • public function revisarCuestionario(): Método encargado de revisar el cuestionario resuelto por el estudiante y emitir una nota. • public function notaCuestionario(): Método encargado de mostrar las notas de los cuestionarios resuelto por el estudiante. | Profesor. Estudiante. Ejercicio. Tema. Subtema. |

Tabla 10: Tarjeta CRC CuestionarioController.

2.5 Modelo de datos

El modelo de datos describe el comportamiento lógico y físico de los elementos persistentes de utilidad para dar soporte de información al sistema. Es una representación abstracta de los datos de una organización y las relaciones entre ellos. El propósito de un modelo de datos es, por una parte, representar los datos y, por otra, ser comprensible. Estos modelos surgen de la necesidad de mecanismos que capten con mayor facilidad la semántica del mundo real, mejorando la calidad de diseño de sistemas.

Con el fin de garantizar la persistencia de los datos se confeccionó el modelo entidad relación, el cual se muestra a continuación.

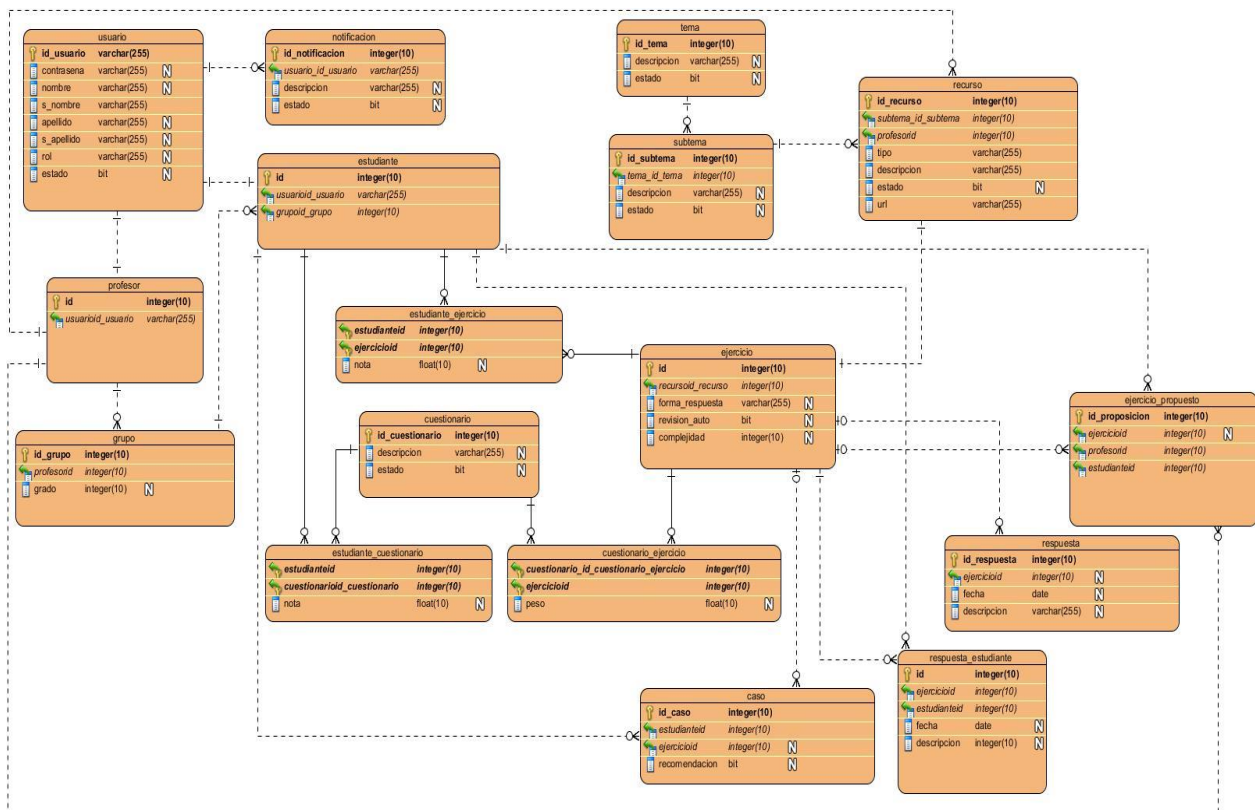


Figura 1: Modelo de Datos.

2.6 Conclusiones parciales

- Se definieron las historias de usuario que permitieron tener la base para la implementación de la aplicación.
- Se definieron un total de 27 requisitos funcionales y 7 requisitos no funcionales referentes a términos de usabilidad, eficiencia, seguridad entre otros.
- Se definió el plan de iteraciones en correspondencia con las Historias de Usuarios del sistema, llegando a la conclusión de que la duración de la implementación del proyecto será aproximadamente de 5 meses.
- Los artefactos generados según la metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

Capítulo 3. Implementación y pruebas

Introducción

En el presente capítulo se describen cuáles son los elementos necesarios para la implementación del sistema propuesto mediante las tareas de ingeniería llevadas a cabo por el equipo de desarrollo, partiendo de los artefactos generados en el capítulo anterior del presente trabajo de diploma. Se obtiene el diagrama de despliegue en el cual se presenta a través de nodos la organización de los componentes del sistema, sus dependencias y la relación existente entre ellos. Además, se describen los casos de pruebas, que luego de implementada la solución, serán usados para probar que la misma cumpla con los requerimientos establecidos.

3.1 Tareas de Ingeniería

Durante el transcurso de las iteraciones, se realiza la implementación de las historias de usuarios definidas por el cliente y descritas por el equipo de desarrollo en la etapa de Exploración. Como parte del plan, se descomponen las historias de usuarios en tareas de la ingeniería, las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente. Una historia de usuario puede tener asociada varias tareas de ingeniería para poder darle respuesta a las necesidades de su cliente. A continuación, se describen 3 de estas tareas, en el Anexo 3 se muestran otras 2.

| Tarea de Ingeniería | |
|--|---|
| Número de tarea: 1 | Número de Historia de Usuario: 2 |
| Nombre de la tarea: Crear usuario. | |
| Tipo de tarea: Desarrollo | Puntos estimado: 0.6 |
| Fecha inicio: 25-11-2015 | Fecha fin: 27-11-2015 |
| Descripción: Se implementan los métodos encargados para crear un usuario, con nombre y apellidos, usuario, rol, contraseña y confirmar contraseña, se verifica que el usuario que se quiere registrar no existe en la base de datos para poder crearlo. | |

Tabla 11: Tarea de ingeniería 1 HU2 Crear usuario.

| Tarea de Ingeniería | |
|---|--|
| Número de tarea: 20 | Número de Historia de Usuario: 20 |
| Nombre de la tarea: Evaluar ejercicio. | |
| Tipo de tarea: Desarrollo | Puntos estimado: 1.2 |
| Fecha inicio: 22-2-2016 | Fecha fin: 1-3-2016 |
| Descripción: Se implementan los métodos encargados de verificar las respuestas de los ejercicios que resuelven los estudiantes, y según la respuesta, el sistema emite una evaluación. | |

Tabla 12: Tarea de ingeniería 20 HU20 Evaluar ejercicio.

| Tarea de Ingeniería | |
|--|--|
| Número de tarea: 23 | Número de Historia de Usuario: 23 |
| Nombre de la tarea: Recomendar ejercicio. | |
| Tipo de tarea: Desarrollo | Puntos estimado: 2.8 |
| Fecha inicio: 11-5-2016 | Fecha fin: 27-5-2016 |
| Descripción: Se implementan los métodos según un SBC encargado de verificar que ejercicios recomendarle al estudiante que este autenticado, luego de resolver varios ejercicios propuestos por el profesor en el sistema. | |

Tabla 13: Tarea de ingeniería 23 HU23 Recomendar ejercicio.

3.2 Estándares de Codificación

Los estándares de codificación, también conocidos como estilos de programación o convenciones de código, son convenios para escribir código fuente en ciertos lenguajes de programación. Permiten que el código en consecuencia sea entendible y que todos los participantes lo puedan entender en un menor tiempo. Los estilos de codificación explicados en este epígrafe son los empleados por el lenguaje de programación PHP en el IDE PhpStorm.

3.2.1 Definiciones de clases

En las declaraciones de clases los nombres de estas comienzan con letra mayúscula y tienen su llave de apertura una línea más abajo de la declaración. En la siguiente figura se muestra un ejemplo.

```
class Estudiante extends Model
{
    protected $table = 'estudiantes';
    protected $fillable = [
        'user_id',
        'grupo_id'
    ];
}
```

Figura 2: Definición de clases.

3.2.2 Definición de métodos

Los nombres de los métodos son frases que incluyen verbos, dado que los mismos generalmente son el producto de concatenar varias palabras, se debe emplear la primera palabra en minúscula, mayúscula para denotar la letra de inicio de cada una de las palabras restantes por las que esté formado y minúscula para las letras intermedias. Los atributos pasados por parámetro se separarán por una coma y un espacio después de ésta en caso de existir más de uno. A continuación, se muestra un ejemplo en la siguiente figura:

```
public static function getEjercicios(Request $request, $subtema_id)
```

Figura 3: Definición de métodos.

3.2.3 Asignaciones a variables

La asignación a variables se realiza siempre dejando un espacio antes y después del signo de igualdad (=) como se puede observar en la siguiente figura:

```
$ejercicio = $recurso->ejercicio;
$ejercicio->recurso_id = $recurso->id;
$ejercicio->tipo_respuesta = $request->tipo_respuesta;
$ejercicio->rev_auto = $request->rev_auto;
$ejercicio->complejidad = $request->complejidad;
```

Figura 4: Asignación a variables.

3.2.4 Estructuras de control

Las estructuras de control `if()` y `foreach()`, se utilizaron colocando la llave de apertura en la línea de código que le sigue a la estructura de control y la de cierre quedando alineada con la de apertura, en la siguiente figura se muestra un ejemplo:

```
if ($request->ajax())
{
    $ejercicios = collect(null);
    $recursos = Recurso::where('id_subtema', $subtema_id)->where('tipo', 'Ejercicio')->get();
    foreach ($recursos as $recurso)
    {
        $ejercicios->push($recurso->ejercicio);
    }
    return response()->json($ejercicios);
}
```

Figura 5: Estructuras de control.

3.3 Diagrama de despliegue

El modelo de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse. Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos (35).



Figura 6: Diagrama de Despliegue.

- **PC – cliente:** Ordenador cliente que se conecta a través de un navegador web al servidor central donde reside la aplicación. No se necesita tener instalado el producto localmente.

- **Servidor Web y de Base de Datos:** El servidor de aplicación es utilizado para la publicación de la aplicación. Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor y es el encargado de ejecutar el código de las páginas servidor. Además, se utilizará PostgreSQL como Base de Datos (BD) que permitirá almacenar toda la información de la aplicación.
- **Protocolos de comunicación:** Un protocolo de comunicación es un conjunto de reglas establecidas entre dos dispositivos para permitir la comunicación entre ambos:
 - **Conexión HTTPS:** Protocolo de Transferencia de Hipertexto Seguro conocido (por sus siglas en inglés HTTPS), es el protocolo utilizado entre los navegadores de los clientes y el servidor web. Este elemento en la arquitectura representa un tipo de comunicación segura y confiable entre clientes y servidor, ya que garantiza que cualquier dato introducido será cifrado.

3.4 Fase de Pruebas

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código, las cuales están diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida y están diseñadas por el cliente final.

- **Pruebas de aceptación:** Consideradas como pruebas de caja negra (del inglés “*Black box system tests*”). En esta prueba los clientes son responsables de verificar que los resultados sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Además, que las pruebas de aceptación poseen un peso superior a las unitarias, muestran a los desarrolladores la satisfacción del cliente (32).
- **Pruebas unitarias:** El propósito de las pruebas es aislar cada parte del programa y mostrar que las partes individuales son correctas. Las pruebas unitarias son otra forma de comunicación ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo utilizar su funcionalidad.

Se decide trabajar para esta investigación con las pruebas de aceptación porque muestran a los desarrolladores la satisfacción del cliente, además, poseen un peso superior a las unitarias.

3.4.1 Pruebas de Aceptación

Las pruebas de aceptación se realizan para asegurar que las funcionalidades cumplen su objetivo y si satisfacen las necesidades del cliente. En esta prueba los clientes son responsables de verificar que los resultados sean correctos. Se realizan sobre la interfaz del *software* y son completamente indiferentes al comportamiento interno y a la estructura del programa.

Las pruebas de aceptación permiten que:

- Las funciones del *software* son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Las pruebas de aceptación intentan encontrar errores de las siguientes categorías:

- Funciones incorrecta o ausente.
- Errores de interfaz.
- Errores de rendimiento.
- Errores de inicialización y de terminación.
- Una vez concluidas las pruebas realizadas a cada HU se considera terminada la aplicación informática.

Para representar las pruebas de aceptación se definieron los siguientes elementos:

- **Código:** Representa al caso de prueba, incluye el número de la HU de la prueba y si posee diferentes escenarios.
- **HU:** Número de la HU a la cual pertenece.
- **Nombre:** Junto al código, conforma el identificador del caso de prueba.
- **Descripción:** Se describe el caso de prueba y se especifican los aspectos fundamentales de los datos de entrada.

- **Condición de ejecución:** Se verifica que cada parámetro cumpla las condiciones de ejecución.
- **Entrada:** Se muestran los parámetros de entrada del procedimiento.
- **Resultados Esperados:** Se especifica el resultado que debe arrojar el procedimiento.
- **Resultado Obtenido:** Respuesta visual del sistema después de realizar el caso de prueba.
- **Evaluación de la prueba:** Clasificación de la prueba en satisfactoria o insatisfactoria.

Pruebas de aceptación de la primera iteración

En esta primera iteración se realizaron las pruebas a las 13 HU que están propuestas en el plan de entrega. A continuación se muestra un caso de prueba de esta iteración.

| Caso de Prueba de Aceptación | |
|--|-------------------------------|
| Código: HU2_P1 | Historia de Usuario: 2 |
| Nombre: Crear usuario. | |
| Descripción: Esta prueba se encarga de validar la HU “Crear usuario”, adiciona en la BD la información de un nuevo usuario que el administrador o profesor introduzca. | |
| Condiciones de Ejecución: El administrador o profesor debe estar autenticado en el sistema. | |
| Entradas/Pasos de Ejecución: | |
| <ol style="list-style-type: none"> 1. Autenticarse con el usuario “yfs” y la contraseña “Yaser123”. 2. Ir a Crear usuario. 3. Introducir en el formulario los datos del usuario, nombre “Luis Pérez Pérez”, usuario “lperez”, rol “estudiante”, grupo “2”, contraseña “Luis 123” y repetir contraseña “Luis 123”. 4. Hacer clic en el botón Crear. | |

| |
|--|
| Resultado Esperado: El sistema debe almacenar en la BD los datos del nuevo usuario "Iperez". |
| Resultado Obtenido: Los datos del nuevo usuario "Iperez" fueron almacenados correctamente en la BD. |
| Evaluación de la Prueba: Satisfactoria. |

Tabla 14: Prueba de aceptación para la HU Crear usuario.

En esta primera iteración de pruebas se realizaron 2 etapas con un total de 13 casos, en la primera etapa se detectaron 2 no conformidades representando un 15%, luego de corregir los errores, se realizó un segundo momento de pruebas donde todas fueron satisfactorias dando un 100% de resultados, por lo que se decidió pasar a la segunda iteración.

Pruebas de aceptación de la segunda iteración

En esta segunda iteración se probaron las 8 HU planificadas en el plan de entrega. A continuación, se muestra un caso de prueba de esta iteración.

| Caso de Prueba de Aceptación | |
|---|--------------------------------|
| Código: HU20_P1 | Historia de Usuario: 20 |
| Nombre: Evaluar ejercicio. | |
| Descripción: Esta prueba se encarga de validar la HU "Evaluar ejercicio", dando una nota según la respuesta que envíe el estudiante. | |
| Condiciones de Ejecución: El estudiante debe de estar autenticado en el sistema. | |
| Entradas/Pasos de Ejecución: | |
| <ol style="list-style-type: none"> 1. Autenticarse con el usuario de un estudiante "lmederos" y la contraseña "Luis123". 2. Ir a algún Tema y escoger un ejercicio para resolver. 3. Seleccionar una respuesta de las 3 posibles que sale. 4. Hacer clic en el botón Evaluar. | |

| |
|--|
| Resultado Esperado: El sistema debe mostrar un mensaje con la nota obtenida en ese ejercicio. |
| Resultado Obtenido: El sistema muestra un mensaje “Su nota es 10”. |
| Evaluación de la Prueba: Satisfactoria. |

Tabla 15: Prueba de aceptación para la HU Evaluar ejercicio.

En esta segunda iteración se realizaron 2 etapas de pruebas al igual que en la iteración anterior, ya que se encontraron 3 no conformidades representando un 37 % de resultados insatisfactorios, una vez corregidos los errores, se desarrolló la segunda etapa donde no se encontraron no conformidades representando el 100% de resultados satisfactorios.

Pruebas de aceptación de la tercera iteración

En esta tercera iteración se realizaron las pruebas de las 6 HU acordadas en el plan de entrega. A continuación, se muestra un caso de prueba de esta iteración.

| Caso de Prueba de Aceptación | |
|--|--------------------------------|
| Código: HU22_P1 | Historia de Usuario: 22 |
| Nombre: Proponer ejercicio. | |
| Descripción: Esta prueba se encarga de validar la HU “Proponer ejercicio”, dándole la opción al profesor de proponerle ejercicios a sus estudiantes. | |
| Condiciones de Ejecución: El profesor debe de estar autenticado en el sistema. | |
| Entradas/Pasos de Ejecución: | |
| <ol style="list-style-type: none"> 1. Autenticarse con el usuario de un profesor “yfs” y la contraseña “Yaser123”. 2. Ir a Proponer Ejercicios. 3. Seleccionar Tema, Subtema, Ejercicio y estudiante(s) que le va a proponer el ejercicio. 4. Hacer clic en el botón Proponer. | |
| Resultado Esperado: El sistema debe mostrar un mensaje informando que el ejercicio se propuso correctamente. | |

Resultado Obtenido: El sistema muestra un mensaje “El ejercicio se propuso correctamente”.

Evaluación de la Prueba: Satisfactoria.

Tabla 16: Prueba de aceptación para la HU Proponer ejercicio.

En esta última iteración se realizaron 2 etapas de pruebas al igual que en las iteraciones anteriores, donde se encontraron 2 no conformidades representando un 33 % de resultados insatisfactorios, una vez corregidos los errores, se desarrolló la segunda etapa donde no se encontraron no conformidades representando el 100% de resultados satisfactorios.

3.4.2 Resultados de las pruebas

Luego de terminar las etapas de pruebas en cada iteración, se logró mitigar todas las no conformidades. En la siguiente gráfica se muestran las pruebas satisfactorias de cada etapa en sus respectivas iteraciones.

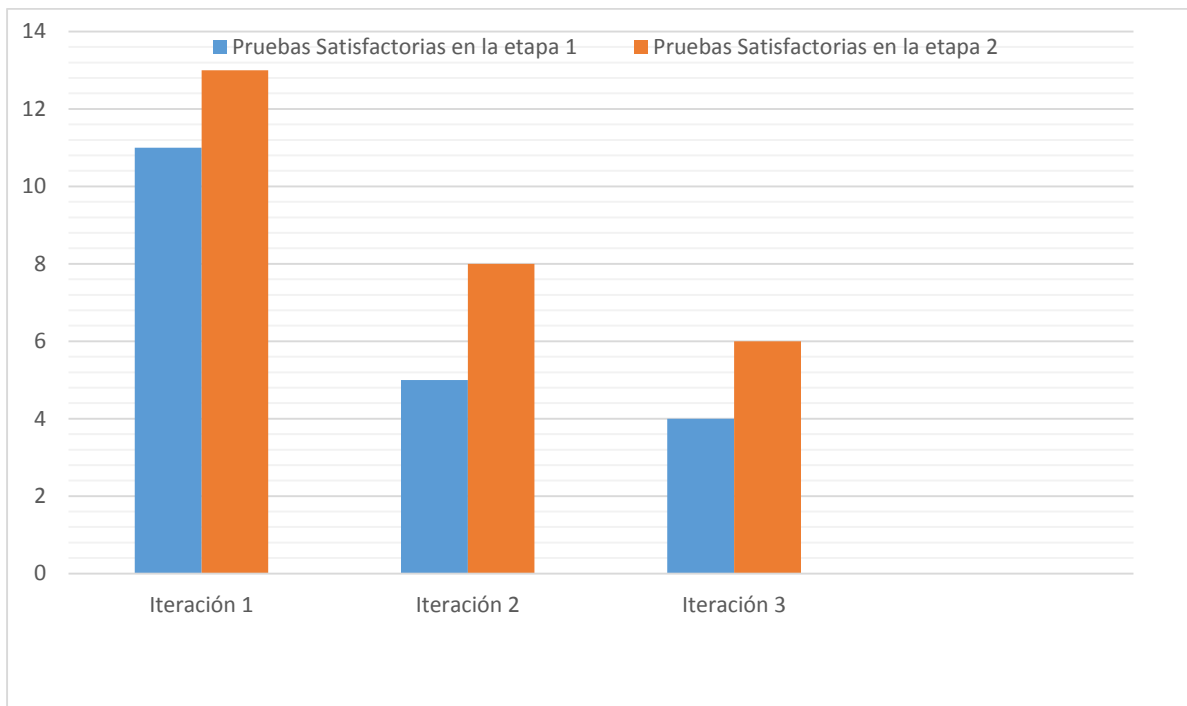


Figura 7: Pruebas de aceptación satisfactorias por iteraciones.

3.5 Conclusiones parciales

- En la fase de desarrollo se definió la codificación utilizada para las clases, métodos y estructuras de control definidas en el sistema, además se desarrollaron las tareas de ingeniería.
- La realización de las pruebas de caja negra permitió la detección de No Conformidades que fueron creadas durante la etapa de codificación y a su vez corregidas.

Conclusiones Generales

Con la finalización de la presente investigación se alcanzan las siguientes conclusiones:

- Se desarrolló una aplicación web del tipo apoyo al aprendizaje en la asignatura de Física, dotando a los estudiantes y profesores de 11no grado con un medio de apoyo para diseñar, orientar, evaluar, estudiar y consolidar los contenidos dados en clases.
- Se incluyó en la propuesta SIAF un Sistema Experto Basado en Casos posibilitando la inclusión de elementos de Inteligencia Artificial que logran perfeccionar el trabajo que pueda desarrollar una plataforma educativa común.
- Se diseñaron y aplicaron pruebas a la solución propuesta lo que permitió validar el grado de terminación y la calidad del *software*.

Recomendaciones

Teniendo en cuenta los resultados obtenidos en esta investigación, el autor recomienda:

- Implementar las demostraciones virtuales para los restantes temas de la asignatura Física de 11no grado.
- Continuar perfeccionando el SBC para obtener las recomendaciones de los ejercicios lo más óptimas posible.
- El despliegue de la aplicación en los Pre – Universitarios de la capital para que pueda ser utilizada por los estudiantes y profesores.

Referencias bibliográficas

1. **Vialart Vidal, Niurka.** Revista Cubana de Enfermería. [En línea] Enfermería Informática ¿una contradicción o una oportunidad para el trabajo en red?, 2011. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-03192011000200010&lng=es&nrm=iso&tlng=es.
2. **Hargreaves, Andy.** *Enseñar en la sociedad del conocimiento (La educación en la era de la inventiva)*. Barcelona : s.n., 2003.
3. **de la Torre Navarro, Lilia María.** Revista Cubana de Informática Médica. [En línea] Las TIC en el proceso de enseñanza aprendizaje a través de los objetos de aprendizaje, 2012. http://scielo.sld.cu/scielo.php?pid=S1684-18592012000100008&script=sci_arttext.
4. **Morejón Labrada, Sonia .** Cuadernos de Educación y Desarrollo. [En línea] EL SOFTWARE EDUCATIVO UN MEDIO DE ENSEÑANZA EFICIENTE. <http://www.eumed.net/rev/ced/29/sml.htm>.
5. **Marquès, Pere .** [En línea] http://www.lmi.ub.es/te/any96/marques_software/.
6. **Sitio oficial de ATutor.** [En línea] <http://www.atutor.ca>.
7. **Analizamos 19 Plataformas de e-learning.** Ariel Clarenc, Claudio, Mariel Castro, Silvina y otros. s.l. : Congreso Virtual Mundial de e-Learning, 2013.
8. **Moodle.** [En línea] https://docs.moodle.org/all/es/Acerca_de_Moodle.
9. **Ecured.** [En línea] http://www.ecured.cu/Plataformas_Educativas.
10. **CubaEduca.** [En línea] http://www.cubaeduca.cu/index.php?option=com_content&view=article&id=3838&Itemid=153.
11. **CubaEduca.** [En línea] http://www.cubaeduca.cu/index.php?option=com_content&view=article&id=4705:en-construccion&catid=26.
12. **Softonic.** *Quars - Laboratios de Física.* [En línea] <http://quars-laboratios-de-fisica.softonic.com/>.

13. **Educando El Portal de la Educación Dominicana.** [En línea] [Citado el: 10 de 11 de 2015.] <http://www.educando.edu.do/centro-de-recursos/multimedia/software/elg-magic/>.
14. **GeorK.** *Libros de texto digitales.* [En línea] <http://www.geork.com/es/fisg11es>.
15. **FreeCAD.** [En línea] [Citado el: 10 de Noviembre de 2015.] <http://www.freecadweb.org/>.
16. **UN SISTEMA BASADO EN CASOS PARA LA TOMA DE DECISIONES EN CONDICIONES DE INCERTIDUMBRE.** Gutiérrez Martínez, Iliana y Bello Pére, Rafael E. No.2, Departamento de Cincia de la Computación, Universidad de Las Villas,Cuba : Revista investigación operacional, 2002, Vol. 23.
17. **GOMEZ GALLEGO, JUAN PABLO.** *Fundamentos de la Mtodología RUP.* UNIVERSIDAD TECNOLÓGICA DE PEREIRA : s.n., 2007.
18. **Universo el Periódico de los Universitarios.** [En línea] Programación extrema: “Metodología para desarrollo ágil de aplicaciones”, 2012. [Citado el: 12 de Noviembre de 2015.] http://www.uv.mx/universo/486/infgral/infgral_15.html.
19. **Pemberton, Steven .** W3C. [En línea] [Citado el: 11 de Noviembre de 2015.] <https://www.w3.org/MarkUp/>.
20. **LibrosWeb.** [En línea] [Citado el: 11 de Noviembre de 2015.] http://librosweb.es/libro/javascript/capitulo_1.html.
21. **LibrosWeb.** [En línea] [Citado el: 11 de 11 de 2015.] http://librosweb.es/libro/css/capitulo_1.html.
22. **PlusGlobal.** [En línea] [Citado el: 11 de 11 de 2015.] <http://plusglobal.com/blog/agiliza-tu-trabajo-con-un-framework/>.
23. **Rodríguez, Txema .** GenBeta. [En línea] [Citado el: 13 de 11 de 2015.] <http://www.genbetadev.com/frameworks/bootstrap>.
24. **jQuery.** [En línea] [Citado el: 14 de 11 de 2015.] <http://jquery.com/>.
25. **Laravel.** [En línea] [Citado el: 20 de 11 de 2015.] <https://laravel.com>.

26. **Visual Paradigm.** [En línea] [Citado el: 17 de 11 de 2015.] <https://www.visual-paradigm.com>.
27. **Ecured.** [En línea] Sistema Gestor de Base de Datos. [Citado el: 20 de 11 de 2015.] http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos.
28. **PostgreSQL.** [En línea] [Citado el: 22 de 11 de 2015.] <https://www.postgresql.org/about/>.
29. **Brending.** [En línea] [Citado el: 20 de Enero de 2016.] <http://www.brendingdesigners.com.ar/articulos/servidorweb.html>.
30. [En línea] 2013. [Citado el: 20 de Enero de 2016.] <http://pnfiservidores.blogspot.com/2013/01/ventajas-y-desventajas-de-xampp.html>.
31. **Letel, Patricio y Penadés, M^a Carmen.** Ciencia y Técnica Administrativa. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. [En línea] [Citado el: 11 de 1 de 2016.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
32. **Joskowicz , José .** *Reglas y Prácticas en eXtreme Programming* . Universidad de Vigo, España : s.n., 2008.
33. **Guías de programación y aprendizaje.** IBM. s.l. : IBM WebSphere Commerce, 2013, Vol. Versión 5.5.
34. **Larman, Craig.** *Applying UML and Patterns: An Introduction to Object - Oriented Analysis and Desing*. 2004.
35. **Universidad Sale Siana, Bolivia.** *Diagrama de despliegue*. [En línea] [Citado el: 10 de 5 de 2016.] <http://virtual.usalesiana.edu.bo/web/conte/practica/22012/2132.pdf>.

Anexos

Anexo 1: Historias de Usuario

| Historia de Usuario | |
|---|--|
| Número: 1 | Nombre de Historia de Usuario: Autenticar usuario |
| Usuario: usuario_sistema | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 0.6 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 0.6 |
| Descripción: El sistema debe permitir la autenticación de estudiantes, profesores y administrador para acceder al sistema mediante usuario y contraseña según el rol y los permisos que tengan. Si es estudiante solo tendrá acceso a las vistas de los estudiantes, en caso de ser profesor a las vistas de los profesores y administrador a la vista que le corresponde. | |
| Observaciones: <ul style="list-style-type: none">• Si el usuario no está registrado en el sistema o introduce mal el usuario o contraseña, no podrá autenticarse en el sistema. | |

Tabla 17: HU Autenticar usuario.

| Historia de Usuario | |
|---|---|
| Número: 3 | Nombre de Historia de Usuario: Modificar usuario |
| Usuario: administrador y profesor | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 0.6 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 0.6 |
| Descripción: El sistema debe permitirle al administrador modificar todos los usuarios creados en el sistema y el profesor podrá solo modificar los estudiantes que atiende. Esta funcionalidad debe ser con el mismo formulario que se usa en el crear usuario. Luego de llenados todos los campos debe dar clic en el botón “ Modificar ” y automáticamente el sistema actualiza los datos | |

del usuario en la base de datos. En caso de dejar algún campo vacío en el formulario, el sistema debe mostrar un mensaje de error informando en que campo se cometió el mismo y se mantiene en el formulario.

Observaciones:

- El administrador o profesor debe de estar autenticado para poder modificar usuarios.

Tabla 18: HU Modificar usuario.

| Historia de Usuario | |
|--|--|
| Número: 4 | Nombre de Historia de Usuario: Eliminar usuario |
| Usuario: administrador y profesor | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 0.6 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 0.6 |
| Descripción: El sistema debe permitirle al administrador y profesor eliminar usuarios que causen baja en el sistema. El administrador podrá eliminar todo tipo de usuarios, mientras que los profesores solo podrán eliminar estudiantes de su grupo de clases. | |
| Observaciones: | |
| <ul style="list-style-type: none"> • El administrador o profesor debe de estar autenticado para poder eliminar usuarios. | |

Tabla 19: HU Eliminar usuario.

| Historia de Usuario | |
|------------------------------------|---|
| Número: 14 | Nombre de Historia de Usuario: Crear recurso |
| Usuario: profesor | Iteración Asignada: 2 |
| Prioridad en Negocio: Media | Puntos Estimados: 0.7 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 0.7 |

Descripción: El sistema debe permitir al profesor crear algún recurso de los siguientes tipos: bibliografías, ejercicios, demostraciones y gráficos. Para ello el profesor debe de llenar un formulario, en el caso de que sea un ejercicio, aparecerán los siguientes campos: **“Tema”**, **“Subtema”**, **“Título”**, **“Descripción”**, **“Complejidad”** y **“Tipo de Ejercicio”**. Donde el campo tema será para que el profesor seleccione un tema de los existentes en el sistema, el campo subtema será para que seleccione un subtema del tema seleccionado anteriormente, el campo título es para que pueda escribir un título al ejercicio que está creándose, la descripción es para que escriba la orientación del ejercicio, el campo complejidad es para que pueda seleccionar el nivel de dificultad del ejercicio: **“Baja”**, **“Media”** o **“Alta”**, el tipo de ejercicio es para que pueda seleccionar cómo será el ejercicio que se está creando: **“Selección”** o **“Texto”**, si escoge de tipo selección le saldrá otro formulario para que introduzca las posibles respuestas, una respuesta correcta y otras dos incorrectas. Luego de llenados dichos campos debe de hacer clic en el botón **“Crear”** y automáticamente el sistema guardará el ejercicio creado satisfactoriamente. En caso de dejar algún campo vacío, se mostrará un mensaje de error en el campo donde se cometió el mismo y se mantiene en el formulario. Si selecciona bibliografía, demostración o gráfico, debe de llenar un formulario que contenga los siguientes campos: **“Tema”**, **“Subtema”**, **“Descripción”** y **“Cargar Archivo”**. Donde el campo tema será para que el profesor seleccione un tema de los existentes en el sistema, el campo subtema será para que seleccione un subtema del tema seleccionado anteriormente, la descripción es para que escriba de que es el recurso y el campo cargar archivo es para que el profesor pueda buscar donde se encuentra el archivo que va a subir al sistema, los archivos que vaya a subir el profesor solo podrán ser del formato pdf en el caso de las bibliografías, jpg los gráficos y html las demostraciones. Luego de llenados dichos campos debe de hacer clic en el botón **“Crear”** y automáticamente el sistema guardará el recurso creado satisfactoriamente. En caso de dejar algún campo vacío, se mostrará un

mensaje de error en el campo donde se cometió el mismo y se mantiene en el formulario.

Observaciones:

- El profesor debe de estar autenticado para poder crear algún recurso.

Tabla 20: HU Crear recurso.

| Historia de Usuario | |
|--|---|
| Número: 22 | Nombre de Historia de Usuario: Proponer ejercicios |
| Usuario: profesor | Iteración Asignada: 3 |
| Prioridad en Negocio: Baja | Puntos Estimados: 1.2 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| <p>Descripción: El sistema debe permitir al profesor proponerles ejercicios, estos van para una base de casos y se le enviara una notificación a los estudiantes de los ejercicios que le propuso el profesor para que los resuelvan. Para proponer los ejercicios se mostrará un formulario que contenga los siguientes campos: “Tema”, “Subtema”, “Ejercicios” y “Estudiantes”. El campo tema es para que el profesor pueda seleccionar de que tema del ejercicio, el subtema será del tema que escoja el profesor, luego se mostraran todos los ejercicios del subtema seleccionado y todos los estudiantes que atiende el profesor autenticado, luego selecciona el ejercicio y los estudiantes que quiere proponerle el ejercicio. Una vez llenado todos los campos debe hacer clic en el botón “Proponer” y automáticamente el sistema muestra un mensaje diciendo si la operación fue satisfactoria o no. En caso de algún error, se muestra un mensaje informando en que campo fue el mismo.</p> | |
| <p>Observaciones:</p> <ul style="list-style-type: none">• El profesor debe de estar autenticado para poder proponerle ejercicios a sus estudiantes. | |

Tabla 21: HU Proponer ejercicio.

Anexo 2: Tarjetas CRC

| Tarjeta CRC | |
|--|--|
| Clase: EstudianteController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • public function crateEstudiante(): Método encargado de crear estudiantes por el profesor. • public function updateEstudiante(): Método encargado de actualizar estudiantes por el profesor. • public function destroyEstudiante(): Método encargado de poder eliminar estudiantes por el profesor. • public function recomendarEjercicio(): Método encargado de recomendarle ejercicios al estudiante según sus resultados en los ejercicios que tenga en la base de casos. | <p>Profesor.</p> <p>Tema.</p> <p>Subtema.</p> <p>Ejercicio.</p> <p>Caso.</p> |

Tabla 22: Tarjeta CRC EstudianteController.

| Tarjeta CRC | |
|---|-------------------------------------|
| Clase: GrupoController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • public function createGrupo(): Método encargado de crear grupos por el administrador. • public function updateGrupo(): Método encargado de poder | <p>Estudiante.</p> <p>Profesor.</p> |

| | |
|--|--|
| <p>actualizar grupos por el administrador.</p> <ul style="list-style-type: none"> • public function destroyGrupo(): Método encargado de poder eliminar grupos por el administrador. | |
|--|--|

Tabla 23: Tarjeta CRC GrupoController.

| Tarjeta CRC | |
|--|------------------|
| Clase: TemaController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • public function createTema(): Método encargado de poder crear temas de la asignatura por el profesor. • public function updateTema(): Método encargado de poder actualizar temas por el profesor. • public function destroyTema(): Método encargado de poder eliminar temas por el profesor. | <p>Profesor.</p> |

Tabla 24: Tarjeta CRC TemaController.

| Tarjeta CRC | |
|---|----------------------------|
| Clase: SubTemaController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • public function createSubTema(): Método encargado de poder crear subtemas por el profesor. • public function updateSubTema(): Método encargado de poder actualizar subtemas por el profesor. | <p>Profesor. Tema.</p> |

| | |
|--|--|
| <ul style="list-style-type: none"> • public function destroySubTema(): Método encargado de poder eliminar subtemas por el profesor. | |
|--|--|

Tabla 25: Tarjeta CRC SubtemaController.

| Tarjeta CRC | |
|---|---|
| Clase: EjercicioController | |
| Responsabilidades | Colaboraciones |
| <ul style="list-style-type: none"> • public function createEjercicio(): Método encargado de crear ejercicios por el profesor en el tema y subtema seleccionado. • public function updateEjercicio(): Método encargado de poder actualizar ejercicios por el profesor. • public function destroyEjercicio(): Método encargado de poder eliminar ejercicios por el profesor. • public function evaluarEjercicio(): Método encargado de evaluar los ejercicios de tipo selección resueltos por el estudiante. • public function revisarEjercicio(): Método encargado de que el profesor pueda revisar los ejercicios de tipo texto que los estudiantes resuelvan. • public function notaEjercicio(): Método encargado de mostrar las notas de los ejercicios resueltos por los estudiante. | <p>Recurso.</p> <p>Profesor.</p> <p>Estudiante.</p> <p>Cuestionario.</p> <p>Caso.</p> |

| | |
|---|--|
| <ul style="list-style-type: none"> • public function proponerEjercicio(): Método encargado de que el profesor pueda proponerle ejercicios a sus estudiantes. | |
|---|--|

Tabla 26: Tarjeta CRC EjercicioController.

Anexo 3: Tareas de ingeniería

| Tarea de Ingeniería | |
|---|---|
| Número de tarea: 1 | Número de Historia de Usuario: 1 |
| Nombre de la tarea: Autenticar usuario. | |
| Tipo de tarea: Desarrollo | Puntos estimado: 0.6 |
| Fecha inicio: 23-11-2015 | Fecha fin: 25-11-2015 |
| Descripción: Se implementan los métodos encargados de verificar que el usuario que quiere autenticarse tiene permisos y mandarlo para la vista que le corresponde según el rol que tenga. Son 3 roles, administrador, profesor y estudiante. | |

Tabla 27: Tarea de ingeniería Autenticar usuario.

| Tarea de Ingeniería | |
|---|---|
| Número de tarea: 3 | Número de Historia de Usuario: 3 |
| Nombre de la tarea: Modificar usuario. | |
| Tipo de tarea: Desarrollo | Puntos estimado: 0.6 |
| Fecha inicio: 30-11-2015 | Fecha fin: 2-12-2015 |
| Descripción: Se implementan los métodos encargados de modificar los datos de un usuario en el sistema. | |

Tabla 28: Tarea de ingeniería Modificar usuario.