

Universidad de las Ciencias Informáticas

Facultad 2



Título: Módulo para la evaluación de competencias antes de la realización de una práctica de laboratorio en un Sistema de Laboratorios a Distancia.

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

Autores: Annia Viera Sánchez
Jeffrey Díaz Mora

Tutor: Ing. Iralys Torres Pérez

Co-Tutor: Msc. Omar Mar Cornelio

Ing. Juan Carlos Fiorenzano González

Junio - 2015

Declaración de autoría

Declaramos que somos los autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los 23 días del mes de junio del año 2015.

Annia Viera Sánchez

Firma del Autor

Jeffrey Díaz Mora

Firma del Autor

Ing. Juan Carlos Fiorenzano González

Firma del Co-tutor

Msc. Omar Mar Cornelio

Firma del Co-tutor

Ing. Iralys Torres Pérez

Firma del Tutor

Datos de Contacto

Autores:

Annia Viera Sánchez

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo electrónico: aviera@estudiantes.uci.cu

Jeffrey Díaz Mora

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo electrónico: jdiaz@estudiantes.uci.cu

Tutor:

Ing. Iralys Torres Pérez

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo electrónico: itorres@uci.cu

Cotutor:

Ing. Juan Carlos Fiorenzano González

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo electrónico: jcfiorenzano@uci.cu

Cotutor:

Msc. Omar Mar Cornelio

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo electrónico: omarmar@uci.cu

Dedicatoria

Annia:

Dedico este trabajo a las personas más especiales e importantes en mi vida: a mis padres, a mis abuelos, que me han apoyado, comprendido, educado y formado. También se lo dedico a mis amigos y a mi hermano Anniel, quienes más que hermanos han sido las personas que me han enseñado a enfrentar la vida y no verla tan complicada como lo parece, a mis tíos, tías, primos, a toda mi familia a mi amor Juan Carlos, que me ha ayudado a levantarme tras cada caída, me ha comprendido, amado, consentido y guiado hasta el final de este difícil camino. A todas las personas que me han brindado su apoyo en esta dura tarea que es tratar de ser mejor cada día.

Jeffrey:

Dedico esta tesis a mi familia. Especialmente a mis abuelos Carmen y Ángel donde quiera que estén sé que están orgullosos de mí, también a mi abuelo René quien a pesar de la distancia siempre me ha dado su apoyo, a mi abuela María quien siempre se preocupó por mí. A mis padres quienes siempre me dieron su apoyo incondicional, a mi hermana querida quien siempre me dio la motivación y el estímulo para seguir adelante. Les agradezco a todos ellos porque han contribuido en mi formación para convertirme en la persona que soy hoy.

Agradecimientos

Quiero agradecer de forma muy especial a mis padres por siempre estar a mi lado apoyándome, brindándome todo su amor, su cariño, su comprensión para que pueda salir adelante en la vida, a ellos les debo todo lo que soy, creo que no me alcanzaría la vida entera para decirles cuanto los quiero, y para hacer por ellos lo que han hecho por mí.

Un agradecimiento muy especial para una persona súper especial en mi vida a mi abuela Franca por ser conmigo la mejor abuela del mundo, por su cariño, entrega, dedicación y por complacerme en todos mis antojos.

A mi amor Juan Carlos por brindarme su amor, por complacerme y acompañarme en todo momento, por estar ahí y luchar junto a mí para que este momento llegara por inspirarme a seguir luchando y no dejarme vencer.

A mi hermana Betty por estar presente cada vez que la necesito, por ser más que mi amiga por ser mi hermana.

A mis amigas Diani y Cristina las que siempre han estado ahí, que no han sido amigas de camino, sino de toda la vida.

A mis otros padres Marisel, Lázaro, Nuria y Juani muchas gracias por todo su apoyo y por acogerme como una más de la familia.

A mis tíos, a mis primos, a mis amigos, a mi familia en general que siempre han estado velando y cuidando de mí.

A mi compañero de tesis Jeffrey por su paciencia, amistad y comprensión en todo momento.

A todos los profesores que nos brindaron su ayuda incondicional en el desarrollo de la tesis, porque a veces quien menos uno espera, tiende su mano en el momento preciso. Gracias por brindarnos sus conocimientos y experiencias.

A mis compañeros de aula y a mis compañeras de residencia, especialmente a Balbina, Anabel, Mimi, Paima, Dayana, Liliu, Nahema, Anakaena y Misleidis que compartieron conmigo los buenos y malos momentos durante estos 5 años de la carrera y que han sido mi otra familia. Por convertir todos estos años los años más significativos de mi vida.

A todos muchísimas gracias por formar parte de la historia de esta etapa de mi vida.

Annia

A mis padres que sin ellos no hubiera sido posible la realización de este trabajo.

A mi queridísima hermana por su apoyo y comprensión.

A mis tíos Jose y María de los Angeles por siempre preocuparse por mí y mi superación.

A mis queridos abuelos por su cariño y dedicación.

A mi consorte Luis Ángel por su motivación.

A mis amigos de la Universidad: Luisma, Pristi, Mario, Shanelish, Edward, Ricardo, Fusbiel, Joel, Manuel Alejandro, William, Ramón SA, Artiles, Reynier Arias, Sandra, Karla, Orbe con quienes viví momentos inolvidables.

A mi compañera de tesis por compartió conmigo momentos de alegría, tristeza, preocupación, serenidad. Por comprenderme cuando más lo necesité.

A mi cotutor y socio Juan Carlos que sin su ayuda no hubiese sido posible esta tesis.

A mi cotutor el Msc Omar Mar y a su familia por el tiempo, apoyo y dedicación brindados.

A mi tutora Tralys.

A mis compañeros del grupo 2510.

A mis antiguos compañeros del grupo 7106.

A mis compañeros de cuarto: Alberto, Jairo, Danielito el Fuli, Claudia.

A todos los profesores que con su ejemplo y sabiduría me inculcaron valores imprescindibles para la vida. Especialmente al profesor Menéndez.

A los que ya no se encuentran en la universidad pero que compartieron momentos de su vida conmigo.

Quisiera agradecer a todas las personas que de una manera u otra influyeron en mi vida universitaria y que me ayudaron a hacer posible este sueño.

Jeffrey

Resumen

La evolución de las tecnologías de la Información y las Comunicaciones han introducido mejoras en la educación, posibilitando superar barreras que antes se imponían, surgiendo un nuevo paradigma para el Proceso de Enseñanza Aprendizaje conocido como la educación a distancia. En el Departamento de Automática y Sistemas Computacionales, de la Universidad Central “Marta Abreu” de las Villas se ha implementado el Sistema de Laboratorios a Distancia que permite la realización de prácticas tanto simuladas como reales permitiendo disminuir los problemas existentes con los laboratorios tradicionales como son la disponibilidad horaria para el acceso a los equipos y la limitación en los puestos de trabajo. Debido a la carencia de mecanismos para la evaluación de competencias y conocimientos sobre la capacitación de un estudiante para la realizar una práctica de laboratorio se decide realizar la implementación de un módulo integrado al Sistema de Laboratorio a Distancia que permita la evaluación de las competencias de los mismos. La aplicación propone un conjunto de cuestionarios creados previamente por los profesores como base inferencial para determinar las competencias que debe cumplir una persona para realizar una práctica de laboratorio con equipos remotos

Palabras clave: competencia, evaluación, educación a distancia, Sistema de Laboratorios a Distancia

Índice

INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS-METODOLÓGICOS RELACIONADOS CON LA INVESTIGACIÓN.....	15
1.1. Educación a distancia.....	15
1.1.1. <i>E-learning</i>	16
1.1.2. <i>Tipos de Laboratorios de prácticas</i>	17
1.1.3. <i>Sistema de Laboratorios a Distancia (SLD)</i>	18
1.1.4. <i>Sistema de Laboratorios a Distancia de la Universidad Central de las Villas Marta Abreu</i>	20
1.2. Competencias en Ingeniería Automática	21
1.2.1. <i>Tipos de Competencia</i>	22
1.2.2. <i>La evaluación</i>	22
1.2.3. <i>Evaluación de competencias</i>	23
1.2.4. <i>Modelo de evaluación de competencias</i>	24
1.3. Propuesta de adecuación al Modelo de Mar	26
1.3.1. <i>Descripción de las etapas del Modelo</i>	27
1.3.2. <i>Estructura del Modelo propuesto</i>	28
1.3.3. <i>Enfoque multi-criterio</i>	28
1.4. Integración con el SLD	29
1.5. Soluciones Existentes	29
1.4.1. <i>Comparación de soluciones</i>	32
1.6. Tecnologías, Herramientas y Metodologías.....	33
CONCLUSIONES PARCIALES DEL CAPÍTULO	38

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.....	39
2.1. Descripción de la solución.....	39
2.1.1. <i>Análisis de la estructura del modelo propuesto</i>	39
2.1.2. <i>Descripción de los Objetos del Modelo del Dominio</i>	44
2.2. Exploración.....	45
2.2.1. <i>Historias de Usuario</i>	45
2.3. Planificación	47
2.3.1. <i>Estimación de esfuerzo por historias de usuarios</i>	47
2.3.2. <i>Plan de iteraciones</i>	48
2.3.3. <i>Plan de entregas.</i>	50
2.4. Requerimientos de software	50
2.4.1 <i>Características funcionales del sistema</i>	51
2.4.2 <i>Características no funcionales del sistema</i>	51
2.5. Roles del sistema	52
2.6. Diseño	52
2.6.1. <i>Diagrama de Clases</i>	52
2.6.2. <i>Arquitectura base de la aplicación</i>	53
2.6.3. <i>Arquitectura Cliente-Servidor</i>	54
2.6.4. <i>Patrones de diseño</i>	55
2.6.5. <i>Patrones GRASP</i>	55
2.6.6. <i>Tarjetas Clase, Responsabilidad y Colaboración.</i>	57
CONCLUSIONES PARCIALES DEL CAPÍTULO	58
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS.	59

3.1. Implementación de la aplicación	59
3.1.1. <i>Tareas de la Ingeniería</i>	59
3.1.2. <i>Estándares de codificación</i>	60
3.1.3. <i>Interfaces principales de la aplicación.</i>	63
3.2. Proceso de pruebas de la metodología XP	65
3.2.1. <i>Tipos de pruebas.</i>	66
3.3. Técnicas de prueba	67
3.3.1. <i>Pruebas de caja blanca</i>	67
3.3.2. <i>Pruebas de caja negra</i>	67
3.3.3. <i>Método de caja negra</i>	67
3.3. Descripción del proceso de pruebas	68
3.5. Análisis de los resultados del proceso de pruebas	72
CONCLUSIONES PARCIALES DEL CAPÍTULO	73
CONCLUSIONES.....	74
RECOMENDACIONES	75
REFERENCIAS BIBLIOGRÁFICAS	76

Introducción

La utilización de las Tecnologías de la Información y las Comunicaciones (TIC) suscita nuevas líneas de trabajo e investigación, abriendo la posibilidad al desarrollo de nuevas estrategias docentes en el dictado de asignaturas de carácter científico. En la actualidad Internet juega uno de los principales papeles en la formación de los individuos, ofreciendo herramientas y eliminando todo tipo de fronteras.

La experimentación interactiva con dispositivos del mundo real mejora la motivación de los estudiantes y desarrolla aptitudes para resolver problemas reales. Así, el trabajo con sistemas reales resulta más enriquecedor que el que se realiza sólo con modelos matemáticos.

Durante los últimos años, el uso de las nuevas tecnologías en la educación ha experimentado un extraordinario crecimiento. En la actualidad es fácil encontrar direcciones que ofrecen cursos de aprendizaje por Internet. Herramientas, tales como tutoriales, imágenes, videos o simulaciones, ofrecen excelentes oportunidades a los profesores y a los estudiantes. En todo caso, estos sistemas no permiten sustituir la realización de prácticas presenciales cuando la manipulación de sistemas físicos por parte del alumno, constituye una parte importante del proceso de aprendizaje. Este hecho es especialmente importante en las asignaturas impartidas en el área de Ingeniería de Sistemas y Automática. (Aktan, y otros, 1996) (Poindexter, y otros, 1999)

La utilización de sistemas físicos como servomotores, péndulos, depósitos acoplados o sistemas de control de temperatura constituye una herramienta básica en los cursos de teoría de control. Por otra parte, el uso de autómatas programables que controlan líneas de producción, sistemas de clasificación o manipuladores es típico en cursos de automatización.

La enseñanza del Control Automático, o de alguna otra disciplina con gran contenido experimental, requiere de un elemento que permita al estudiante poner en práctica todos los conocimientos que vaya adquiriendo a lo largo del estudio de la materia. Este papel en las enseñanzas tradicionales lo desempeña el laboratorio de prácticas (Kheir, 1996), el cual, inexorablemente, requiere de la presencia física del estudiante para poder manipular los sistemas de control y las plantas existentes en un entorno controlado bajo la supervisión del profesor. (Leiner, 2002) (Messom, y otros, 2002) En estos casos en que el desarrollo de las prácticas experimentales tienen lugar en forma presencial, se presenta la desventaja de que el alumno debe trasladarse al lugar donde se encuentra el laboratorio, también los recursos utilizados en estos laboratorios son costosos, así como el número de estudiantes suele ser elevado en la mayoría de los casos. Adicionalmente, los laboratorios generalmente no cuentan con los recursos físicos y humanos suficientes para atender a la totalidad de los alumnos de un curso, lo cual limita la cantidad y calidad de la experimentación.

Por estos motivos surgen los Sistemas de Laboratorios a Distancia (SLD) permitiendo sacar más provecho de un laboratorio físico sin tener que implementar otros. El SLD es un laboratorio a distancia

virtual y real, cuyo principal objetivo es permitir a los usuarios aprender a ajustar controladores predefinidos y a diseñar sus propios controladores para probarlos en dispositivos reales que son accesibles a través de Internet.

La docencia tradicional presencial en estos temas viene afectada por tres factores que degradan los resultados educativos: (Paya, 2003)

1) La limitada disponibilidad horaria de los laboratorios: los alumnos pueden realizar experimentos en los laboratorios en unos horarios poco flexibles y en la mayoría de los casos demasiado ajustados. Por tanto el estudiante no puede experimentar libremente y analizar los aspectos que considere más necesarios con un mayor énfasis.

2) Limitación en el número de puestos de laboratorio: el alto coste de las maquetas y sistemas de control hace que el número de puestos disponibles sea bajo para el número de alumnos por grupo práctico. Esto provoca que deba compartir los equipos con tres o más compañeros reduciéndose en muchos casos el aprovechamiento de los experimentos realizados. Asimismo, limita la variedad de elementos físicos disponibles en un laboratorio.

3) El sistema de evaluación: los métodos tradicionales de presentación de informes con los resultados, en muchos casos no supone un mecanismo eficaz para el aprendizaje real. La autoevaluación por parte del alumno incentiva el autoaprendizaje y precisa de una actitud más activa durante las prácticas de laboratorio.

El uso de Laboratorios Remotos a través de Internet permite mejorar estos aspectos consiguiendo: (Paya, 2003)

- Mejorar la disponibilidad de los equipos de laboratorio.
- Incrementar el número de prácticas.
- Horarios de acceso más amplios y flexibles.
- Incrementar la dedicación del alumno a la realización de las prácticas.
- Permitir la repetición del experimento.
- Mejorar el proceso de aprendizaje.
- Alarga la vida útil de los equipos.
- Enfocar al estudiante en los aspectos relevantes del experimento.

Sin embargo al no existir un profesor supervisando las estrategias de control diseñadas por los estudiantes en la práctica de laboratorio, es posible que personas no competentes accedan a la plataforma y puedan generar un mal funcionamiento en sus estaciones de trabajo. Según (Laboratorio distribuido con acceso remoto para la enseñanza de la robótica., 2009) las consecuencias de la

manipulación por parte de una persona no competente traen consigo los problemas que a continuación se mencionan:

- Se podría “colgar” el sistema tratando de computar datos erróneos.
- Si las entradas hacen que el sistema presente oscilaciones sostenidas de alta frecuencia, entonces se puede desajustar mecánicamente el equipo o dañarlo.
- Una persona que no requiera de dicha práctica puede tener el equipo ocupado en el momento que otra persona lo necesite.
- El uso intensivo de los equipos requiere una gran inversión económica en mantenimiento preventivo y mayormente correctivo, por lo que la utilización desmesurada no está concebida para estos equipos.

La evaluación por competencias es un proceso de recogida de evidencias (a través de actividades de aprendizaje) y de formulación de valoraciones sobre la medida y la naturaleza del progreso del individuo, según unos resultados de aprendizaje esperados.

La norma UNE 66173 (Aenor, 2003) sobre gestión de competencia define la valoración de competencias como la *"operación que consiste en determinar el estado de las competencias y habilidades de un individuo"*, también destaca el concepto de evaluación de competencias como valoración para subrayar que *"es ante todo un procedimiento para generar valor a lo que las personas aprenden"*.

Esta identificación de la evaluación como la valoración del logro de la competencia adquirido por el estudiante tiene como objetivo principal valorar en qué grado el estudiante posee y domina una determinada competencia. La tarea del profesor como evaluador se ve revolucionada ya que no es suficiente con emitir una calificación final que refleje el nivel de conocimientos adquirido por el estudiante. (Aenor, 2003) Es necesario que se plantee la evaluación como un proceso que requiere conocer en qué grado el estudiante posee la competencia antes de iniciar una determinada tarea. Por lo antes planteado se puede arribar que la valoración del logro de la competencia por el estudiante se define así como el eje cardinal que vertebra todo el proceso de evaluación.

La evaluación de competencias es el resultado de diseñar instrumentos para que el alumno demuestre con evidencias que puede realizar las tareas de la competencia. Los instrumentos de evaluación se utilizan para conocer la marcha y resultado del proceso de enseñanza-aprendizaje. Debido a las peculiaridades que propone el sistema de educación a distancia en cuanto a la ausencia física del estudiante y del profesor y la necesidad de obtener una respuesta inmediata de la evaluación, se decide hacer uso del cuestionario porque cumple con las expectativas por las posibilidades que brinda de poder ser conformado por un experto en el tema a evaluar, ofrece respuestas predefinidas y de igual manera evalúa eficazmente el conocimiento.

Del análisis realizado anteriormente se puede identificar que la evaluación de competencias constituye actualmente un elemento clave para la realización de prácticas de laboratorios en la carrera Ingeniería Automática identificándose las siguientes insatisfacciones:

- Se carece de mecanismos para la evaluación de las competencias de los estudiantes practicantes.
- Se desconoce si un estudiante está capacitado para realizar una práctica de laboratorio.

Partiendo de lo antes expuesto se define como **problema a resolver**:

¿Cómo evaluar competencias antes de la realización de una práctica de laboratorio a distancia en un Sistema de Laboratorios a Distancia?

Del problema anterior se obtiene como **objeto de estudio**: Procesos de evaluación de competencias.

Estableciendo como **campo de acción**: Evaluación de competencias en los Sistemas de Laboratorios a Distancia.

El **Objetivo general** del presente trabajo es:

Desarrollar un módulo que permita la evaluación de competencias antes de la realización de una práctica de laboratorio en un Sistema de Laboratorios a Distancia.

Para el cumplimiento de los objetivos se trazaron las **tareas de investigación** siguientes:

- Elaboración de los elementos introductorios para fundamentar el tema de investigación.
- Elaboración del marco teórico referencial sobre los laboratorios a distancia y sistemas utilizados para la evaluación de conocimiento en las prácticas de laboratorio.
- Análisis de los sistemas existentes que evalúan competencias para comparar semejanzas y diferencias.
- Selección de la metodología, lenguajes de programación, herramientas y técnicas para el desarrollo del sistema informático.
- Diseño e implementación de un sistema informático para resolver la problemática planteada.
- Análisis y aplicación de los tipos de prueba de *software* para verificar la calidad del sistema.

Para el desarrollo de las tareas científicas se utilizan **Métodos de Investigación** en la búsqueda y procesamiento de la información. Los mismos se dividen en teóricos y empíricos.

Los **Métodos Teóricos** son factibles en el estudio de las características poco observables del objeto de investigación. Dentro de este grupo se utilizaron:

- **Histórico-lógico:** Permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Se utiliza este método ya que para darle cumplimiento a los objetivos se trazaron algunas tareas de la investigación como estudiar las herramientas informáticas y lenguajes que se utilizarán en la elaboración de la aplicación web, así como los principales sistemas informáticos existentes que brindan la posibilidad de acceder y controlar Laboratorios Remotos.
- **Analítico-Sintético:** Facilita el entendimiento del fenómeno en el que se trabaja, es más útil la división de este en diferentes fases, y de esta forma descubrir sus características generales, lo que ayuda a seguir una correcta investigación. Estudiando y analizando todo lo relacionado al tema específico a abordar, se pudo hacer una profundización de todas sus características, dígase objetivos, logros, o deficiencias que se derivan, así como también las mejoras que se pueden proponer para obtener resultados eficientes.

Como **Método Empírico** se usó el Análisis Documental para revisar toda la bibliografía relacionada tanto con los procesos de evaluación de competencias, para comprender sus particularidades, nexos y desarrollo; así como para el estudio de los referentes teóricos que permitieron desarrollar la propuesta.

El presente trabajo está estructurado en tres capítulos, además contiene varios anexos con los artefactos generados durante el desarrollo de la solución. A continuación se describe el objetivo principal de cada uno de los capítulos:

Capítulo 1: Fundamentos teóricos-metodológicos relacionados con la investigación.

En este capítulo se establecen todos los elementos teóricos de la investigación. Se realiza un estudio del estado del arte de sistemas similares al que se desea implementar, así como de marco de trabajo, patrones de diseño, lenguajes de programación, metodologías y herramientas orientadas al desarrollo web. Además se estudiaron los elementos fundamentales del proceso de desarrollo de *software* para dar solución al problema planteado.

Capítulo 2: Análisis y diseño del sistema.

En este capítulo se presenta la solución propuesta con todos los aspectos definidos en la fundamentación teórica. Esta solución contiene los artefactos necesarios para desarrollar el sistema a construir, entre los que se encuentran: historias de usuarios, tarjetas CRC, modelo de datos, tareas de ingeniería, entre otros.

Capítulo 3: Implementación y pruebas

En este capítulo se realizan validaciones de los resultados que se obtuvieron del sistema en ejecución, haciendo uso de algunos de los métodos definidos en el marco teórico de la investigación que permiten establecer comparaciones entre los resultados obtenidos. Además, se valida que el diseño realizado cumpla con la calidad requerida y que el sistema implementado satisfice las necesidades de los clientes.

Capítulo 1: Fundamentos teóricos-metodológicos relacionados con la investigación.

En este capítulo se abordan aspectos relacionados con la Educación a distancia, Sistema de Laboratorios a Distancia y demás conceptos que se relacionan con los anteriores para un mejor estudio y caracterización, además se describe la importancia y posibilidades de la educación a través de Internet y el aporte a la sociedad actual así como los sistemas de evaluación e instrumentos existentes. También se describen y analizan las principales herramientas para el desarrollo de la solución.

1.1. Educación a distancia

Con el desarrollo vertiginoso que han tenido las nuevas tecnologías de la información y las comunicaciones (TIC) durante los últimos años en cuanto a las redes de computadoras estandarizadas, capacidad de almacenamiento y velocidad de los procesadores se ha podido apreciar una mejora en todas las esferas de la vida. La educación no quedó exenta de este desarrollo y surge una nueva forma de educación que rompe con el agnóstico modelo de las clases presenciales en un aula.

La Educación a Distancia elimina la rígida frontera de espacio y tiempo que impone el paradigma de la clase tradicional. Es un modelo que requiere de técnicas especiales para el diseño de los cursos, técnicas de instrucción especiales, así como también pautas especiales de organización y administración. (Gorga, 2007)

Es una modalidad educativa en la que los estudiantes no necesitan asistir físicamente al aula. Normalmente, se envía al estudiante por correo el material de estudio (textos escritos, videos, pistas de audio) y el devuelve los ejercicios resueltos. Al aprendizaje desarrollado con las nuevas tecnologías de la comunicación se le llama *e-learning*.

Algunas de las ventajas que ofrece la Educación a distancia son:

- ✓ Estudiar y trabajar paralelamente.
- ✓ Estudiar sin abandonar su propio espacio (casa-trabajo).
- ✓ Progresar respetando los propios ritmos y de acuerdo a las propias expectativas.
- ✓ Permitir que la persona desarrolle un sistema de autoaprendizaje.
- ✓ Poder acceder a las nuevas tecnologías de la comunicación e información y, por su intermedio, a conocimientos disponibles en la red nacional y extranjera.
- ✓ Ahorrar económicamente: al no trasladarse físicamente de un lugar a otro, al uso del tiempo y al no tener que invertir en textos básicos que le son entregados por la Institución.

1.1.1. E-learning

Este término se centra en la enseñanza a distancia, abierta, flexible e interactiva, basada en el manejo de las nuevas tecnologías de la información y la comunicación, como los recursos que ofrece Internet. El concepto abarca al conjunto de las metodologías y estrategias de aprendizaje que emplean tecnología digital o informática para producir, transmitir, distribuir, y organizar conocimiento entre individuos, comunidades y organizaciones. (Piskurich, 2003) Las nuevas formas de aprendizaje forman parte esencial de este continuo intercambio, de ahí que surjan nuevas modalidades y diferentes conceptos relacionados con esta sociedad de información.

La enseñanza con tecnología presenta numerosas ventajas, pero también algunas dificultades, que se enumeran a continuación:

Ventajas:

- ✓ Permanente ayuda y apoyo del tutor durante todo el proceso de aprendizaje.
- ✓ El ritmo de trabajo lo marca el propio estudiante.
- ✓ Posibilidad de acceso a los cursos desde cualquier lugar.
- ✓ Se promueve el auto aprendizaje.
- ✓ Las comunicaciones son multidireccionales (alumno-tutor, tutor-alumno, alumno-alumno).

Inconvenientes:

- ✓ Choque con el tradicional modelo de educación presencial.
- ✓ Sentimientos de inseguridad producidos por la novedad o el desconocimiento.
- ✓ El tutor debe dedicar mucho tiempo a cada estudiante.

El proceso de e-learning se puede dividir en dos vertientes. El *e-learning* sincrónico donde la interacción se realiza en tiempo real, o sea, que el profesor interactúa con el alumno en una clase contando con video de alta calidad, pizarra electrónica para realizar gráficos o resaltar parte del contenido, evaluaciones en línea, y un proyector que permite compartir aplicaciones. (Co, 3R group, 2002) En el *e-learning* asincrónico el contenido se encuentra depositado y el alumno accede a él por su propia voluntad. Esto permite la individualidad total del estudiante, permitiéndole ingresar desde su hogar, trabajo o lugar de descanso, en el horario que él decida.

E-learning no es sólo una excelente modalidad de enseñanza a distancia, sino que, sin sustituir a la enseñanza presencial, puede ser una parte integrante en cualquier curso. El *e-learning*, promete llevar también el acceso a la formación y la educación a los sectores sociales menos favorecidos, llegando a los lugares más retirados geográficamente y permitiendo a las personas discapacitadas disfrutar de las mismas oportunidades de estudio que los demás ciudadanos. (Letz, 2002)

1.1.2. Tipos de Laboratorios de prácticas

Dormido, S. (2004) clasifica los laboratorios en función de dos criterios: (1) La forma de acceder a los recursos (local o remota) para propósitos de experimentación y (2) la naturaleza del sistema físico (real o virtual), con lo que los entornos de experimentación quedarían clasificados de acuerdo a la siguiente figura:

	Real	Virtual
Local	Laboratorios presenciales con plantas reales	Laboratorios presenciales con plantas simuladas
Remoto	Teleoperación de una planta real	Laboratorio remoto con plantas simuladas

Figura 1. Tipos de laboratorios de prácticas. (Laboratorios remotos y virtuales en enseñanzas técnicas y científicas, 2007)

A pesar de que nada puede compararse con la interacción real con las plantas físicas (laboratorios reales y presenciales), afortunadamente existen otras opciones que proporcionan a los alumnos la impresión de que están interactuando con plantas reales. (Laboratorios remotos y virtuales en enseñanzas técnicas y científicas, 2007) Normalmente se entiende por laboratorio remoto aquél que existe y puede ser manipulado de forma remota a través de Internet, haciendo uso de Webcams, *hardware* específico para la adquisición local de datos y *software* para dar una sensación de proximidad con el equipamiento y por laboratorio virtual (local o remoto), o utilizando *software* informático genérico o específico para recrear el comportamiento de plantas de experimentación que sólo existen en ordenadores usados para la simulación. (Control Learning: Present and Future, 2004)

Laboratorios Remotos

Los Laboratorios Remotos se pueden considerar como una evolución de los Laboratorios Virtuales. En este caso al sistema computacional se les agregan instrumentación, control y acceso a equipos de laboratorio reales. Ya no hablamos de llevar a cabo prácticas en un simulador, sino que se trata de realizar actividades prácticas de forma local o remota a través de una Intranet o Internet, permitiendo la transferencia de información entre un proceso real y los estudiantes de manera unidireccional o bidireccional. Bajo este esquema el estudiante utiliza y controla los recursos disponibles en un laboratorio, mediante el uso de tarjetas de adquisición de datos, sensores e instrumentos de medida con interfaces de red y *software* como LabView. (Los Laboratorios Virtuales y Laboratorios Remotos en la Enseñanza de la Ingeniería, 2011)

Los Laboratorios Remotos presentan mayores ventajas que los Laboratorios Virtuales, debido a que los primeros proporcionan una interactividad con equipamiento real, en lugar de usar programas que simulan los procesos. Los Laboratorios Remotos son una innovación en el campo de la educación y debe

prestarse atención tanto a su diseño, al estudio de las ventajas e inconvenientes, y sobre todo, a sus aportaciones didácticas.

El uso de los Laboratorios Remotos reporta numerosas ventajas al proceso de enseñanza-aprendizaje, entre las que son posible mencionar:

- Alimenta la motivación del alumno, ya que se pueden llevar a cabo prácticas más pedagógicas y atractivas, posibilitando el desarrollo de un nuevo enfoque para la solución de problemas reales.
- Existen pocas restricciones de acceso a los laboratorios, tanto en horarios como en número de veces; siendo así factible alcanzar una densidad de ocupación más elevada con un número menor de plantas físicas.
- Se puede evitar el mal uso de los equipos y así alargar notablemente la vida útil de los mismos, centrandó ahora la inversión económica en la adquisición de equipamiento, y no en el mantenimiento, como sucede en el laboratorio convencional.

1.1.3. Sistema de Laboratorios a Distancia (SLD)

Los laboratorios a distancia son laboratorios donde los usuarios pueden interactuar con dispositivos reales a través de Internet. Normalmente los usuarios a través de una interfaz Web pueden cambiar algunos parámetros de control, realizar experimentos, ver los resultados y descargar los datos del experimento. (Gillet, 2000)

El principal objetivo de un SLD es permitir a los usuarios aprender a ajustar controladores predefinidos y a diseñar sus propios controladores para posteriormente probarlos sobre un conjunto de dispositivos físicos a través de Internet y analizar su desempeño. (Rosado, 2006)

El SLD utiliza herramientas de diseño de sistemas de control asistido por computadora como Matlab-Simulink para la creación de nuevos controladores, los cuales pueden hacer uso de S-Functions desarrolladas utilizando el lenguaje C, lo que posibilita la creación de controladores complejos de una forma sencilla. (Laboratorio a Distancia para la prueba y evaluación de controladores a través de Internet, 2004)

Características comunes de los laboratorios virtuales y remotos

Al igual que otros laboratorios a distancia (García, y otros, 2003) (Bauchspiess, y otros, 2003), el SLD posee algunas características que son comunes con la gran mayoría de los laboratorios a distancia que operan actualmente tales como:

Disponibilidad: los sistemas de enseñanza basados en Web deben poder estar disponibles las 24 horas del día. Esto implica que el sistema debe tener medidas de autoprotección para garantizar este aspecto.

Todos los experimentos deben de ser equipados con dispositivos *hardware* y *software* que prevengan daños al equipo o al personal presente en el laboratorio.

Accesibilidad: debido a que el SLD está montado sobre una plataforma Web, permite a los usuarios acceder al sistema desde cualquier parte del mundo. Para ello solo es necesaria una computadora con conexión a Internet y un navegador Web.

Facilidad de uso: para usar el sistema los usuarios solo deben tener los conocimientos básicos de los sistemas de control, tales como el modelado de sistemas y el diseño de controladores. De esta forma el usuario se centra en aprender estos temas y evita todos los problemas asociados a la implementación y operación de los equipos usados en las prácticas.

Equidad: a través del uso de sistemas de gestión automatizados, a todos los usuarios del laboratorio se le pueden asignar límites de tiempo durante períodos de mucho trabajo y de cola para asegurarse de que todos tendrán su "parte justa" de tiempo para llevar a cabo experimentos.

Características específicas del SLD

El SLD presenta características que lo diferencian de los laboratorios virtuales y remotos como son: (García, y otros, 2003)

Interfaz de usuario rápida y fácil: una parte muy importante en el desarrollo de un sistema de enseñanza basado en Web es la interfaz de usuario. La principal función de esta parte del sistema es conformar el pedido de las prácticas y mandarlo hacia el servidor Web.

Administración de múltiples pedidos en forma paralela: el SLD permite atender múltiples pedidos de forma paralela administrando de forma centralizada dispositivos similares que se encuentren geográficamente separados pero unidos por redes de área extensa (WAN).

Desarrollo de controladores de forma remota usando Matlab y Simulink: una de las características más importantes del SLD es que permite a los usuarios diseñar sus propios controladores utilizando el ambiente Matlab-Simulink.

Tipo de controlador: todos los experimentos en el SLD pueden ser controlados de dos formas distintas: con controlador predefinido o con controlador definido por el usuario.

Cambio del período de muestreo: el SLD permite a los usuarios variar el período de muestreo de los experimentos.

1.1.4. Sistema de Laboratorios a Distancia de la Universidad Central de las Villas Marta Abreu

En el 2003 la Universidad Central "Marta Abreu" de las Villas implementó un Sistema de Laboratorios a Distancia (SLD) para permitir a los usuarios cambiar las referencias, modificar los parámetros de control, escoger el tiempo de muestreo y diseñar sus propios controladores de una forma sencilla utilizando herramientas ampliamente conocidas en el área del control automático como son Matlab y Simulink. (Experiencias del uso de laboratorios remotos en la enseñanza de la automática, 2011)

Como principales características del SLD es que los usuarios remotos pueden crear controladores que no estén restringidos solo a los bloques provistos por Simulink, ya que permite el desarrollo de controladores que utilicen S-Functions definidas por el usuario en lenguaje C; esto hace posible la prueba y evaluación de controladores complejos de una forma sencilla.

Durante el año 2003 se realizaron las primeras pruebas del SLD desde México y España, con el fin de implementar controladores en los dispositivos localizados en Cuba. El SLD se utilizó para impartir cursos de postgrado en las materias de teoría avanzada de control y robótica en el Instituto Tecnológico de Minatitlán en México. Durante estos cursos el sistema trabajó de forma satisfactoria pero presentaba tiempos de espera grandes entre el momento de mandar el controlador y el momento de recibir la respuesta, por lo que se empezó a trabajar en dos vías:

1. Modificar el formato de las imágenes para tener una carga más rápida, así como revisar los algoritmos encargados de implementar los controladores, de forma que hubiera una reducción en el tiempo de procesamiento.
2. Realimentar al usuario los datos del proceso controlado, para que durante la espera de la página de respuesta pueda visualizar en un *applet*¹ Java el estado del proceso durante el control. Esto permitirá que el usuario sepa que se está procesando la información y no piense que el sistema está "colgado".

Según el autor el SLD ha sido utilizado en diferentes prácticas en cursos de identificación y control de accionamientos para las carreras de Ingeniería Mecánica e Ingeniería en Automática en la Universidad Central "Marta Abreu" de las Villas y en la Universidad de Cienfuegos, ambas localizadas en Cuba. Durante estas lecciones los usuarios mostraron un gran interés por controlar dispositivos de forma remota a través de Internet. Tras una breve explicación sobre el uso del sistema, los usuarios se

¹Un *applet* es un programa que puede incrustarse en un documento HTML. Cuando un navegador carga una página web que contiene un *applet*, este se descarga en el navegador web y comienza a ejecutarse. Esto permite crear programas que cualquier usuario puede ejecutar con tan solo cargar la página web en su navegador.

registraron en el mismo y comenzaron a realizar experimentos en menos de cinco minutos. Uno de los aspectos que más estimuló a los estudiantes fue poder evaluar las diferencias entre los modelos matemáticos y las plantas reales. En promedio, con quince alumnos por grupo, se realizan entre 60 y 80 prácticas en menos de 30 minutos lo que da una idea de la alta tasa de aprovechamiento de los equipos al accederlos de forma remota utilizando Internet. (Experiencias del uso de laboratorios remotos en la enseñanza de la automática, 2011)

No hubo incidencias en el acceso al SLD, se registraron más de cinco mil accesos, principalmente entre abril y junio de 2010. Todas las prácticas simuladas y el proyecto integrado se realizaron en el servidor web, mientras que las prácticas remotas se desarrollaron en siete estaciones del laboratorio de automática.

1.2. Competencias en Ingeniería Automática

Conceptos:

Según diversas fuentes la competencia se puede definir como:

“Conjunto de conocimientos, destrezas y aptitudes necesarias para ejercer una profesión, resolver problemas profesionales de forma autónoma y flexible y ser capaz de colaborar en el entorno profesional y en la organización del trabajo” (Bunk, 1994)

*“**Capacidad** de un individuo para realizar una tarea profesional según ciertos estándares de rendimientos, definidos y evaluados en unas condiciones específicas, a partir de un método de descomposición de funciones y tareas en niveles y unidades de comportamiento observables, adecuados de criterios precisos de rendimiento”* (Belisle, y otros, 1996)

“La competencia, inseparable de los razonamientos, está constituida por los conocimientos (declarativos, de procedimientos,...), las habilidades (menos formalizadas, a veces, llevadas a rutinas), los meta conocimientos (conocimientos de sus propios conocimientos, que sólo se adquieren por medio de la experiencia)” (Montmollin, 1996)

Cuando se hace referencia a la **capacidad**, significa que un individuo sabe cómo hacer una determinada cosa de acuerdo a los aprendizajes adquiridos. Significa que el individuo no sólo aprende conceptos sino que además asimila la forma en la que puede aplicarlos.

De manera que se puede asumir que las competencias son los conocimientos, habilidades, y destrezas que desarrolla una persona para comprender, transformar y practicar en el mundo en el que se desenvuelve.

1.2.1. Tipos de Competencia

Existen tres tipos de competencia que a continuación se describen: (Fernández, 2006)

Competencias Básicas: conjunto de conocimientos, habilidades, actitudes y valores mínimos necesarios que debe poseer todo profesional para integrarse con éxito en la vida laboral y social (lectura, cálculo, idiomas, manejo de las TIC, escritura,...)

Competencias Específicas: conjunto de conocimientos, habilidades, actitudes y valores que están vinculados a condiciones y áreas específicas de ejecución de una determinada disciplina. Son aquellas propias de la titulación, especialización y perfil laboral para los que se prepara al estudiante.

Competencias Genéricas: conjunto de conocimientos, habilidades, actitudes y valores que permiten a un profesional desempeñarse en una gran variedad de funciones y tareas. No van unidas a ninguna disciplina sino que se pueden aplicar a una variedad de materias y situaciones (la comunicación, la resolución de problemas, el razonamiento, la capacidad de liderazgo, la creatividad,...)

Competencias específicas en Ingeniería Electrónica Industrial y Automática:

Según define el Dr. Iván Santana Ching en su tesis doctoral Herramientas para la docencia en automática orientada hacia la metodología ECTS (Sistema Europeo de Transferencia y Acumulación de Créditos), las competencias que debe poseer un estudiante de Ingeniería Automática son:

- Conocimientos sobre los fundamentos de los automatismos y métodos de control.
- Conocimiento y capacidad para el modelado y simulación de sistemas.
- Conocimientos de regulación automática y técnica de control y su aplicación a la automatización industrial.
- Conocimientos de principios y aplicaciones de los sistemas robotizados.
- Conocimiento aplicado de informática industrial y comunicaciones.
- Capacidad para diseñar sistemas de control y automatización industrial.

1.2.2. La evaluación

Concepto de evaluación para la educación

La evaluación es el proceso para comprobar y valorar el cumplimiento de los objetivos propuestos y la dirección didáctica de la enseñanza y el aprendizaje en sus momentos de orientación y ejecución. Se deberán propiciar actividades que estimulen la autoevaluación por los estudiantes, así como las acciones de control y valoración del trabajo de los otros. La evaluación debe permitir, al estudiante, elevar sus conocimientos al estimularlo a superar sus desaciertos, lo que garantiza de forma general su crecimiento

personal. La evaluación debe ser desarrolladora, procesal, integral, contextualizada, democrática, formativa, cualitativa, investigativa y sistemática. (Tunis, 2003)

"Evaluación es el acto que consiste en emitir un juicio de valor, a partir de un conjunto de informaciones sobre la evolución o los resultados de un alumno, con el fin de tomar una decisión. " (Maccario, 1989)

De esta manera lo que busca la definición es establecer una aproximación cuantitativa o cualitativa, atribuir un valor, un juicio, sobre algo o alguien, en función de un determinado propósito, recoger información, emitir un juicio con ella a partir de una comparación y así, tomar una decisión.

1.2.3. Evaluación de competencias

La evaluación de la capacidad o desempeño es una técnica de ejecución que requiere que el estudiante elabore una respuesta o un producto que demuestre su conocimiento, capacidades y habilidades. (Yolanda Argudín, 2007)

También se puede definir que la evaluación de competencias es el resultado de diseñar instrumentos para que el alumno demuestre con evidencias (ejecuciones) que puede realizar las tareas de la competencia. Se deben tener en cuenta los siguientes aspectos:

- ✓ Demostrar con ejecuciones el correcto desempeño de las tareas de la competencia.
- ✓ Comparar desempeño real actual de una persona con un criterio o calidad de desempeño asociado.
- ✓ Considerar todos los dominios involucrados en la competencia: conocimiento, desempeño, actitud.

Los dominios de evaluación que existen son: (Ruiz, 2007)

- ✓ Los conocimientos: objetos, hechos, ideas, conceptos, principios y leyes.
- ✓ Las habilidades: normas de acción, técnicas y criterios.
- ✓ Las actitudes: valores personales, sociales, maneras de aplicar conocimientos.

Instrumentos de Evaluación de Competencias

Los Instrumentos de Evaluación de Competencias son los procedimientos (medio, recurso o técnica) en los que se recogen evidencias sobre el desempeño y conocimiento de una persona, para formarse un juicio a partir de un estándar definido, con el fin de determinar si es competente para desempeñar una determinada tarea. (Diseño centrado en Evidencias, 2006). El objetivo de esta tarea es obtener evidencias para identificar y reconocer competencias adquiridas a través de la experiencia en la educación.

Los instrumentos y técnicas de evaluación son las herramientas que usa el profesor para obtener evidencias de los desempeños de los alumnos en un proceso de enseñanza y aprendizaje.

(Fundamentos Teóricos para la implementación de la Didáctica en el proceso Enseñanza-Aprendizaje, 2014) En la educación, la evaluación permite conocer las competencias adquiridas por el alumno que le servirán en el mundo del trabajo.

Técnicas de evaluación

Las Técnicas de evaluación que existen son:

Escritas:

- **Objetivas:** Selección múltiple, V o F, Completamiento, Términos pareados, Selección o Elección
- **Desarrollo:** Respuesta corta, Respuesta extensa

Orales:

- **Procedimientos de Observación:** Pautas de Cotejo, Check List, Escalas de Apreciación

1.2.4. Modelo de evaluación de competencias

Los instrumentos de evaluación se utilizan para conocer la marcha y resultado del proceso de enseñanza-aprendizaje. Estos instrumentos se pueden realizar de forma escrita, oral, o combinando ambas formas; pueden ser teóricos, prácticos y gráficos. La selección de un tipo u otro dependerá de los objetivos y contenidos que se pretenden verificar. (Fundamentos Teóricos para la implementación de la Didáctica en el proceso Enseñanza-Aprendizaje, 2014)

Existen disímiles instrumentos de evaluación entre los que se encuentran:

- **Anecdotario:** es el registro acumulativo y permanente, que hace el docente, de los hechos relevantes realizados o en los cuales ha participado el estudiante.
- **Debate:** permite una evaluación cualitativa, es posible observar las capacidades del alumno para argumentar así como ciertas actitudes.
- **Encuestas y cuestionarios:** son listados de preguntas, por escrito, que se entregan a diferentes personas que pueden suministrar una determinada información.
- **Entrevista:** la entrevista es una técnica que busca la formación del alumno. La evaluación se lleva a cabo a través del diálogo.
- **Exámenes de desarrollo:** es el examen tradicional de tres a cinco preguntas que el estudiante debe contestar de manera individual.
- **Examen escrito:** es la técnica de evaluación utilizada tradicionalmente. El alumno en estas pruebas recibe una serie de peticiones que ha de contestar o resolver, según sean de carácter

teórico o práctico, en un periodo de tiempo determinado, en ocasiones esta técnica varía y se realizan exámenes orales con el mismo procedimiento.

Debido a las peculiaridades que propone el sistema de educación a distancia, de todos estos instrumentos es necesario elegir uno que permita evaluar los conocimientos de un estudiante y verifique si es competente para realizar una práctica de laboratorio con equipos delicados y de costosa implementación.

De los instrumentos no se tendrán en cuenta los que requieren que el estudiante esté físicamente frente al profesor a causa del modelo de educación a distancia, ni los instrumentos cuya respuesta sea abierta puesto que se requiere que el sistema ofrezca una respuesta inmediata y no que dicha respuesta espere hasta ser moderada por ende el instrumento que cumple las expectativas por las posibilidades que brinda de poder ser conformado por un experto en el tema a evaluar, que ofrece respuestas predefinidas y de igual manera evalúa eficazmente el conocimiento es el cuestionario.

Cuestionario

El cuestionario es una técnica estructurada para recopilar datos, que consiste en series de preguntas, escritas y/u orales, que debe responder un entrevistado. Por lo regular, el cuestionario es sólo un elemento de un paquete de recopilación de datos que permite ofrecer una información necesaria sobre algo o alguien. Desde el punto de vista educacional permite al alumno adquirir o demostrar conocimientos sobre alguna materia específica; y para los profesores brinda información de cuánto ha avanzado el estudiante en una asignatura dada. (Rivera, 2007)

El diseño de cuestionarios a través de paneles de expertos es frecuente en la investigación relativa a la evaluación de conocimientos o actitudes. (Diseño de un cuestionario para evaluar conocimientos básicos de estadística, 2012)

Existen varios tipos de cuestionarios: abiertos, cerrados y mixtos. Los cuestionarios abiertos plantean las preguntas sin sugerir ningún tipo predeterminado de respuesta, sino que dejan espacio para que se exprese todo lo que se considere oportuno. Los cuestionarios cerrados plantean preguntas que deben contestarse, por ejemplo, con un sí o con un no, de manera que se señala uno de los apartados que se ofrece como opción, los mixtos combinan preguntas de ambos tipos. (La elaboración de encuestas, 2010)

Cuestionarios cerrados

Los cuestionarios cerrados se dividen en: completar, relacionar parejas, tachar, matrices, única opción y verdadero/falso, opción múltiple (Rivera, 2007).

Verdadero o Falso:

Con este tipo de cuestionarios se pretende que el estudiante logre identificar la respuesta correcta de una serie de preguntas. Se elaboran muchas preguntas con ideas reales y erróneas, y de acuerdo a los conocimientos del estudiante irá identificando la respuesta que a su juicio corresponda.

Única opción:

Consiste en lograr que el estudiante seleccione sólo una respuesta como correcta dentro de las n posibles respuestas que le son ofrecidas acerca de un tema determinado.

Múltiple opción:

Son preguntas con varias posibles respuestas (opciones) de las cuales una es la correcta y las restantes (distractores) son verosímiles, o de las que todas son parcialmente correctas, pero sólo una de ellas es la más apropiada; en el primer caso, son llamadas de respuesta correcta y, en el segundo, de respuesta óptima.

El cuestionario es una herramienta potente y flexible para evaluar los conocimientos de los alumnos facilitando el intercambio de conocimiento y materiales entre profesores. El mismo posibilita la realización de preguntas organizadas por categorías o por temas para el intercambio entre profesores, donde sí se proponen buenas preguntas se obtendrán resultados muy útiles sobre la capacidad de los alumnos y la preparación ante una práctica de laboratorio.

Los cuestionarios cerrados ofrecen más facilidad a la hora de ser evaluados ya que sus respuestas son predefinidas anteriormente por un experto en el tema, el evaluado solo tiene que marcar la respuesta que crea correcta, de manera que este tipo de cuestionario es el elegido para un proceso de calificación automática debido a la ventaja que presenta no tener que esperar por la revisión de una respuesta abierta y así brindar rapidez al proceso. Además de que son fáciles de aplicar y corregir, pues basta sólo con elaborar bien las preguntas para obtener las respuestas correctas, además de permitir cambiar la respuesta seleccionada instantáneamente. Si el cuestionario se implementa de manera correcta permitirá validar las respuestas entradas sin errores, y esto contribuye a la fiabilidad con que debe contar cada cuestionario que se cree.

1.3. Propuesta de adecuación al Modelo de Mar

En la solución a desarrollar se tomó como base el Modelo del Msc. Omar Mar Cornelio el cual describe el proceso de inferencia para la toma de decisiones sobre las evaluaciones de competencias en un SLD.

A continuación se describen las características generales del modelo que facilitan la comprensión de la propuesta.

El Modelo para la evaluación de competencias, está orientado a soportar para la toma de decisiones en un Sistema de Laboratorios a Distancias. Realiza tres actividades básicas: entrada, procesamiento y salida de información. La Figura 2 representa un esquema general de la propuesta donde intervienen las diferentes etapas.

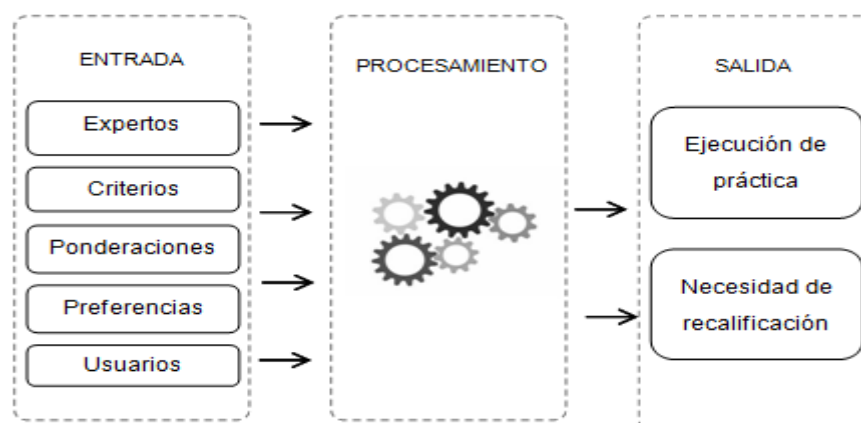


Figura 2. Esquema general de la propuesta. (Modelo para la evaluación de competencias en un sistema de laboratorios a distancia, 2014)

1.3.1. Descripción de las etapas del Modelo

1. Entrada de información: proceso mediante el cual el sistema de información toma los datos que requiere para ser procesada (Duan A., 2012). En la propuesta, existen datos gestionados manualmente que son aquellas que se proporcionan de forma directa por el usuario como es la gestión expertos, criterios, ponderación otorgado por un experto a un criterio dado, la preferencia es la evaluación otorgada a las respuestas brindadas por los usuarios o alternativas.

2. Procesamiento de información: capacidad del Modelo para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Esta característica permite la transformación de datos aleatorios en información organizada. El Modelo interviene directamente en este proceso, siendo posible determinar las competencias de un usuario o alternativa.

3. Salida de información: la salida es la capacidad de un sistema de información para visualizar los datos procesados. Las unidades típicas de salida son las impresoras, interfaces, cintas magnéticas entre otros (Lemos, 2012). Para el Modelo propuesto las informaciones fundamentales brindadas es el elemento

necesario para la toma de decisiones centrado en la necesidad de recapacitar a un usuario o alternativa o la posibilidad de continuar el trabajo en el Sistema de Laboratorios a Distancias.

1.3.2. Estructura del Modelo propuesto

Para manejar el proceso de selección, decisión, interpretación y aplicación del comportamiento que refleja el razonamiento (Cortez, 2010) procesa e interpreta reglas y procesamiento operacional que se encargan de resolver un problema de decisión (Henoa, 1994). En la lógica clásica, es posible deducirse mediante el empleo de reglas, si su premisa es cierta, también lo será su conclusión (Uitierrez, 2010). El autor propone para ello un procedimiento para la toma de decisiones, el cual se basa en el enfoque multi-criterio (Romero, 1997), (Fernández, 2000) compuesto por los siguientes pasos:

Paso 1: Identificar los criterios valorativos.

Paso 2: Asignar peso a los criterios.

Paso 3: Asignar criterios a persona.

Paso 4: Evaluar criterios a persona.

Paso 5: Determinar importancia de los criterios evaluados.

Paso 6: Procesar inferencia.

Según define el autor con la creación de un procedimiento que aplique técnicas multi-criterios para medir el índice de competencias para los estudiantes practicantes, es posible delimitar el acceso por competencias a las prácticas de control especializadas en un Sistema de Laboratorios a Distancias, y asume que, un estudiante cuyo resultado del procesamiento sea mayor a 0,75 puede realizar satisfactoriamente la práctica especializada de laboratorio.

1.3.3. Enfoque multi-criterio

El análisis multi-criterio sirve como herramienta metodológica que pretende comprender la complejidad e incertidumbre de una situación o decisión donde hay variedad de actores e intereses mediante la comparación de distintas valoraciones.(Romero,1997) Permite describir, evaluar, ordenar, jerarquizar, seleccionar o rechazar las opciones, con base en una evaluación (expresada por puntuaciones o valores) de acuerdo con varios criterios. Ya que, como su nombre lo dice, se utilizan diversos criterios con los que se “califican” las opciones, estos criterios pueden ser de enfoque económico, ecológico o social, entre otros. Estos criterios pueden representar diferentes aspectos, objetivos metas, valores de referencia, niveles de aspiración o utilidad.

Este análisis posibilita la participación de los diferentes actores involucrados en un problema a través de la elección de los criterios de evaluación en un marco de amplia participación social y la calificación directa de las distintas alternativas de elección por parte de dichos actores.

1.4. Integración con el SLD

El módulo de evaluación propuesto se va a integrar con la Plataforma de Laboratorios Virtuales y Remotos implementada en la Universidad Central "Marta Abreu" de Las Villas por el ingeniero Lean Saign Chiang Rodríguez en el año 2009. Dicha plataforma permite la realización de prácticas tanto simuladas como reales usando herramientas de simulación como el *Matlab*², el *Real Time Window Target*³ y *Simulink*⁴, Esta plataforma cuenta con un módulo para visualizar las prácticas reales, el cual brinda al usuario una mayor visualización e interconectividad con la práctica y el sistema físico. Permite la creación de tiempos de laboratorio y sistemas de evaluación de las prácticas por los profesores. Y como novedad tecnológica cuenta con la implementación de un módulo para la creación de prácticas asistidas.

1.5. Soluciones Existentes

En la actualidad existen plataformas web que brindan acceso a Laboratorios Remotos para que los practicantes puedan elaborar modelos matemáticos, cambiar parámetros y contrastar los resultados en los equipos que se encuentran disponibles a través de Internet. A continuación se mencionan algunos ejemplos:

RemoteLabs

La plataforma RemoteLabs es propiedad de la UTS (University of Technology, Sydney) Universidad de Tecnología de Sydney por sus siglas en inglés. RemoteLabs permite a los estudiantes e investigadores poder controlar remotamente equipos de laboratorio para llevar a cabo experimentos de laboratorio,

² MATLAB (MATrix LABoratory, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

³ Este software permite ejecutar Modelos de Simulink en tiempo real en una PC para la creación rápida de prototipos o hardware simulación de sistemas de control y algoritmos de procesamiento de señales.

⁴ Simulink es un entorno de programación visual, que funciona sobre el entorno de programación Matlab. Es un entorno de programación de más alto nivel de abstracción que el lenguaje interpretado Matlab. Simulink genera archivos con extensión .mdl.

similar a lo que harían en un salón de clases, pero a través de Internet. A diferencia de las simulaciones, el RemoteLabs UTS utiliza equipo real, físico que ha sido equipado con cámaras y conectado a internet. Es el más grande de su tipo en todo el mundo y cuenta con una amplia gama de bancos de pruebas disponibles. (Villefromoy, 2008)

Esta plataforma en la actualidad cuenta con los siguientes dispositivos de experimentación: Coldfire: Procesadores embebidos, Tanques Acoplados: Control de nivel de fluidos, Plano Inclinado: Gravedad y fricción, iRobot: Teleoperación de un robot móvil, Hydro: Generación de corriente hidroeléctrica, Wind Tunnel: Control aerodinámico. El sistema permite a los usuarios autenticados tener acceso a los experimentos por un período de tiempo limitado, lo que se considera una sesión. Los usuarios permitidos en RemoteLabs son los que formen parte de AAF (Australian Access Federation) o cualquier universidad en convenio con la UTS. (García Zubía, et al., 2012)

OpenLabs Electronics Laboratory

OpenLabs Electronics Laboratory es el producto del proyecto VISIR (Virtual Instrument Systems In Reality) llevado a cabo por la BTH (Blekinge Institute of Technology) en Suecia. La BTH ha creado un área de trabajo de laboratorio equipada con una interfaz de control remoto único, permitiendo a los estudiantes llevar a cabo experimentos físicos en el hogar o en otros lugares donde tengan acceso a Internet. (The VISIR project an Open Source Software, 2007) El sistema es usado en varios escenarios como práctica antes de las sesiones de laboratorio regulares, suplemento de experimentación a distancia, en sesiones cara a cara. Para realizar una práctica en el laboratorio electrónico solo es necesario el Navegador Web. El sistema brinda equipos básicos, como osciloscopio, multímetro, generador de funciones y fuente de alimentación. Con dichas herramientas y una serie de componentes electrónicos como resistores y circuitos se pueden construir circuitos en el tablero virtual. Cabe destacar que ninguna de las mediciones se simula, los circuitos que se construyen se comprueban y miden, y se muestran los resultados reales.

iLab

iLab parte de la idea de enriquecer la ciencia y la enseñanza de la ingeniería promoviendo el acceso a los Laboratorios Remotos en línea a través de Internet. La visión de iLab es compartir equipos costosos y materiales educativos relacionados con los experimentos de laboratorio tan ampliamente como sea posible dentro de la educación superior haciendo uso de Internet. El proyecto iCampus ha creado Laboratorios Remotos en el MIT (Massachusetts Institute of Technology) en los campos de microelectrónica, ingeniería química, ingeniería estructural y procesamiento de señales. iLab ha puesto en funcionamiento Laboratorios Remotos para ampliar su uso a grandes grupos de estudiantes en el MIT en todo el mundo.

iLab tiene las siguientes metas:

- Minimizar el desarrollo y esfuerzo de gestión para los usuarios y los proveedores de Laboratorios Remotos.
- Proporcionar un conjunto de servicios y herramientas de desarrollo comunes.
- Alcanzar un gran número de usuarios en todo el mundo.
- Permitir múltiples universidades con infraestructuras de red diferentes para compartir el acceso.

WebLab-Deusto

WebLab-Deusto es una iniciativa de la Universidad de Deusto con el objetivo de aumentar el aprendizaje experimental por el uso y desarrollo de Laboratorios Remotos. Con este fin, varios laboratorios se ofrecen de forma gratuita a través de Internet, el *software* subyacente está disponible bajo una licencia de código abierto y el equipo se puede duplicar. WebLab-Deusto es utilizado por las universidades (facultades de ingeniería) y escuelas secundarias de diferentes países de todo el mundo, así como en múltiples proyectos de investigación. El sistema de *software* está disponible en más de 10 idiomas, también proporciona autenticación, autorización, mecanismos de programación, herramientas administrativas o la integración en los sistemas de gestión de aprendizaje. El equipo de trabajo de WebLab-Deusto ha diseñado y desarrollado también equipos personalizados, incluyendo el WebLab-Box (una caja industrial en el que los laboratorios de electrónica se pueden desarrollar con facilidad) y una versión ligera del laboratorio remoto VISIR.

1.4.1. Comparación de soluciones

Nombre del Laboratorio Remoto	Universidad	Acceso	Evaluación	Tecnología	Requerimientos (+:superior)
RemoteLabs	UTS(Australia) Proyecto Sahara-LabShare	Requiere "login" Solo para estudiantes que pertenezcan a la AAF o FEIT	No implementa	Código Abierto Código publicado en GitHub	Plugins: Java JDK+JRE v7+ (actualizado) y Adobe Flash Plugin v10+ Navegador: Mozilla Firefox 3+, Chrome 15+, Internet Explorer 9
OpenLabs Electronics Laboratory	BTH(Suecia) Proyecto VISIR	Requiere "login" Para estudiantes de la BTH Permite sesión de invitados por 30 minutos	No implementa	Código Abierto Permite la descarga del código en el propio sitio.	Navegador: Mozilla Firefox, Chrome, Internet Explorer
iLab MIT	MIT(EEUU) Proyecto iCampus	Requiere "login" Para miembros del MIT con credencial permitida.	No implementa	Software Propietario. Consortio con "Microsoft Research"	Navegador: Mozilla Firefox, Chrome, Internet Explorer
WebLab Deusto	Universidad de Deusto(España)	Requiere "login" Para estudiantes de la Universidad de Deusto. Con acceso a todos los experimentos. Permite una sesión de invitado con usuario y contraseña "demo" con acceso a algunos experimentos.	No implementa	Código Libre Código publicado en GitHub	Navegador: Mozilla Firefox, Chrome, Internet Explorer

Tabla 1. Comparación de soluciones.

Las plataformas de Laboratorios Remotos antes mencionadas carecen de un mecanismo para determinar si una persona es competente para realizar una práctica de laboratorio. Estos sistemas cuentan con herramientas de seguridad como la autenticación de usuarios y la validación paramétrica de los campos de entrada de datos a los equipos remotos, pero no suponen una seguridad frente a personas cuyo objetivo difiere de beneficiarse con la enseñanza a distancia. Al no implementar un sistema de evaluación eficiente que permita determinar la competencia de una persona es posible que la misma haga un uso indebido de los equipos de laboratorio lo cual trae consecuencias negativas al proceso.

1.6. Tecnologías, Herramientas y Metodologías.

Metodología de Desarrollo de Software

Una metodología de desarrollo de *software* es un conjunto de pasos y procedimientos que deben seguirse para desarrollar *software*, indicando quién debe hacer cada actividad, cuándo hacerla y qué debe hacer (Stevens, 2007). Actualmente las metodologías se clasifican en Metodologías pesadas y Metodologías ágiles.

Las metodologías ágiles, tienen como común denominador un modelo de desarrollo incremental para producir tempranamente pequeñas entregas en ciclos rápidos, y predisposición para el cambio y la adaptación continua; según sea la conformidad o no de lo producido, y las modificaciones propuestas por los usuarios. Estas metodologías por lo general se centran en desarrollar productos funcionales más que en conseguir una buena documentación (Canós, 2003). Son metodologías centradas en la implementación, que evita cualquier tipo de documentación fuera del código. Por otra parte, las Metodologías Pesadas están orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Teniendo en cuenta lo antes expuesto y la poca necesidad de contar con la documentación necesaria para la futura implementación del sistema, se propone el uso de una metodología ágil. (Beck, 1999)

Programación Extrema (Extreme Programming, XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo (Stevens, 2007). XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros

estándares reconocidos para los lenguajes de programación utilizados). XP sin dudas es la metodología que resalta por contar con la mayor cantidad de información disponible y es con diferencia la más popular.

Luego de analizadas las metodologías ágiles, para desarrollar la solución se decide usar XP ya que se ajusta mejor a las condiciones de trabajo del equipo de desarrollo, de manera que se cuenta con el cliente como un miembro más del equipo de desarrollo, se aboga por la simplicidad del código de tal forma que sea entendible en cualquier paso, puesto que no se cuenta con mucha documentación, permite ver de manera positiva los cambios que se puedan presentar durante todo el desarrollo de la solución proponiendo un ciclo de vida dinámico, se hace uso de la re-factorización del código, el proyecto empieza en pequeño y añade funcionalidades con retroalimentación continua. El proceso de prueba de XP posibilita probar cada funcionalidad al finalizar cada iteración comprobando si cumple con los requisitos de la herramienta.

Lenguaje de programación

PHP 5.3 es un lenguaje totalmente libre y abierto. Tiene una curva de aprendizaje muy baja, su sintaxis es simple y cumple estándares básicos de la programación orientada a objetos. Todos los IDE (Entorno de Desarrollo Integrado) disponibles son gratuitos y los entornos de desarrollo son de rápida y fácil configuración. (Bakken, 2000) Es fácil desplegar un proyecto en PHP porque dispone de paquetes totalmente autoinstalables que integran PHP: Apache y MySQL, tanto para UNIX (LAMPP), como para Windows (XAMPP). Además PHP ofrece un fácil acceso a bases de datos y soporta muchas de ellas como PostgreSQL, MySQL, SQLite.

Ventajas:

- ✓ Lenguaje totalmente libre y abierto.
- ✓ Curva de aprendizaje muy baja.
- ✓ Los entornos de desarrollo son de rápida y fácil configuración.
- ✓ Fácil despliegue: paquetes totalmente autoinstalables que integran PHP.
- ✓ Fácil acceso a bases de datos.
- ✓ Comunidad muy grande.

Lenguaje de Modelado

Como lenguaje de modelado, se selecciona el Lenguaje Unificado de Modelo (UML por sus siglas en inglés), el cual permite especificar, visualizar y construir los artefactos de un sistema de *software*. UML

ofrece un estándar para describir un modelo, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación y esquemas de bases de datos (Rumbaugh, 2000). Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. UML ayuda al usuario a entender la tecnología en realidad y la posibilidad de mostrar una idea antes de invertir en proyectos que no estén seguros en su desarrollo. Entre las propiedades que tiene este importante lenguaje se encuentra que modela estructuras complejas, soporta estructuras orientadas a objetos, como clases, componentes y nodos.

Herramienta Case

Como herramienta Case se propone la utilización del Visual Paradigm 8.0. La misma es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de *Software*, Analistas y Arquitectos que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. (Burrows, 2010)

Visual Paradigm también ofrece:

- ✓ Navegación intuitiva entre la escritura del código y su visualización.
- ✓ Potente generador de informes en formato PDF/HTML.
- ✓ Documentación automática *Ad-hoc*.
- ✓ Ambiente visualmente superior de modelado.
- ✓ Sofisticado diagramador.
- ✓ Sincronización de código fuente en tiempo real o en demanda.

Sistema Gestor de Base de Datos

PostgreSQL

Como sistema gestor de base de datos (SGBD) se seleccionó el PostgreSQL 9.1, es un sistema de base de datos relacional perteneciente al ámbito del *software* libre. Él mismo se destaca por su robustez, integridad referencial, escalabilidad y cumplimiento de los estándares SQL (Lenguaje de Consultas Estructurado). Este gestor posee características como alta concurrencia, amplia variedad de tipos nativos. (Douglas, 2003) Entre sus ventajas se encuentra la instalación ilimitada, puesto que no hay costo asociado a la licencia del *software*. Es multiplataforma. Permite ahorros considerables en costos de operación ya que ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y

rendimiento. Este SGBD se acopla perfectamente al trabajo con *frameworks* web y lenguajes como PHP, Java, etc.

Frameworks

Un *framework* define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Symfony Framework de desarrollo Web

Framework de desarrollo elegido por su versatilidad para el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Symfony, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja y automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (Potencier, 2007) Symfony está desarrollado completamente en PHP 5.3, además es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas Unix como en plataformas Windows. La versión de Symfony a emplear en la solución será la 2.4.

AngularJS Framework Javascript

Se elige AngularJS como *framework* de Javascript, el mismo propone un conjunto de librerías útiles para el desarrollo de aplicaciones web y una serie de patrones de diseño para llevarlas a cabo. AngularJS dinamiza documentos HTML (Lenguaje de Marcado e Hipertexto) posibilitando así que se puedan resolver la mayoría de las peticiones en el lado del cliente sin tener que recargar con peticiones al servidor de la aplicación. (Arlandy, 2013)

Ventajas de su uso:

- ✓ **Separación de la lógica de presentación de la vista**, de forma que el código Javascript es independiente al código HTML por lo que se podría rehacer la vista sin necesidad de tener que tocar ni una sola línea de la lógica de negocio. Además, el código es extremadamente fácil de testear ya que no depende de ningún contexto externo (lógica pura).
- ✓ **Facilidad a la hora de manipular los elementos del DOM** (Modelo de Objetos del Documento) y sus propiedades. Esta es una tarea que, por norma general, es bastante tediosa

y a medida que la lógica de presentación crece suele ser complicado escribir código que pueda mantenerse con facilidad. AngularJS simplifica mucho la labor en este sentido.

- ✓ **Curva de aprendizaje pequeña.** Con Javascript se garantiza la interacción con el usuario, el enviar mensajes a la pantalla, asignar valores de campos en *default*, envío de peticiones AJAX, entre otras tareas. Con AngularJS estas tareas se simplifican y se vuelve más sencillo el trabajo.
- ✓ **Uso de otras librerías,** el trabajo con AngularJS no impide que se pueda usar alguna otra librería o que se puedan combinar ambas librerías, si se desea usar Angular conjuntamente con otra librería basta con envolver el código de la otra librería en una "directiva" para que pueda usarse dentro del código.
- ✓ **Two-Way Data Binding:** AngularJS implementa Data Binding lo que permite manipular y escuchar los elementos del DOM dinámicamente de manera que el usuario se abstrae de escribir código para realizar estas funcionalidades y le permite enfocarse en la aplicación web. Data Binding permite unir en tiempo real los datos de un modelo dentro de una expresión o una variable sin necesidad de envío a un servidor.
- ✓ **Testing,** al tener componentes aislados, podemos testear su comportamiento de manera independiente. Inyección de dependencias, permite hacer uso de un servicio con solo inyectarlo en el controlador directamente y funciona. AngularJS posee un subsistema de inyección de dependencias (ID) que ayuda al desarrollador a facilitar el entendimiento, desarrollo y testeado de la aplicación.

Bootstrap Framework de Hojas de Estilo

Para el trabajo de diseño con hojas de estilo se elige Bootstrap como *framework* de *software* libre para diseño de sitios y aplicaciones web, el cual contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS (Hoja de estilos en cascada), así como, extensiones de Javascript opcionales adicionales. (Mark Otto, 2014) Su particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice, esta técnica de diseño y desarrollo se conoce como "responsive design" o diseño adaptativo.

Técnica de desarrollo Web

AJAX significa Asynchronous Javascript And XML, no es una tecnología en sí, sino que es un conjunto de tecnologías aplicadas y como su nombre lo indica, se trata de la combinación de Javascript y XML. Javascript hace las peticiones al servidor, el mismo le devuelve un resultado (response) en XML, y este es procesado por Javascript para actualizar los datos de la página, sin tener que recargarla por completo (logrando así una interacción asincrónica entre el servidor y el cliente). (SOARES, 2006) AJAX es una técnica de desarrollo WEB, por la cual se pueden crear aplicaciones WEB más rápidas y cómodas para el usuario. Por medio de esta técnica el cliente puede interactuar con el servidor de manera asincrónica, actualizando las páginas, sin necesidad de volver a cargarlas. Esta técnica, no solo es más cómoda y

amigable para el usuario (ya que se asemeja a las aplicaciones de escritorio) sino que además es más rápida, porque cada vez que se necesita actualizar un dato en una página, no es necesario recargarla nuevamente (solo se recarga la sección necesaria de la misma).

Entorno Integrado de Desarrollo (IDE)

PhpStorm

PhpStorm en su versión 8.0 es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación PHP como lo indica su nombre, ofrece un excelente soporte para PHP incluyendo las últimas versiones del lenguaje y *frameworks*, también HTML, JavaScript, CSS, y otros lenguajes. Cuenta con un inteligente autocompletado de código sensible al contexto, detección de errores e inspecciones y correcciones sobre la escritura del código, posee herramientas de integración de sistemas de control de versiones. (Chaudhary, y otros, 2014) PhpStorm posee un sistema de inspecciones que se encargan de verificar el código a medida que se escribe proponiendo soluciones rápidas que ayudan a escribir código limpio para hacerlo más fácil de mantener. Según (Potencier, y otros, 2007) PhpStorm es el IDE preferido por la mayoría de programadores Symfony ya que es el único que ofrece una integración casi perfecta con gracias a su *plugin*⁵ para Symfony2.

Conclusiones parciales del capítulo

- Del estudio de las literaturas científicas sobre los procesos de evaluación de competencias, se pudo identificar que las soluciones existentes no implementan mecanismos de validación de competencias a personas que acceden a los SLD.
- Del estudio sobre las tendencias actuales del proceso de desarrollo de *software*, se escoge la metodología ágil XP para el desarrollo de la solución al igual que las herramientas PhpStorm 8.0 como IDE, Visual Paradigm como herramienta de modelado, PostgreSQL como gestor de Bases de Datos y los *frameworks* de desarrollo Symfony 2.4 para la aplicación web, Bootstrap para las hojas de estilo, AngularJS para el trabajo con Javascript

⁵ Un *plugin* es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API (Interfaz de Programación de Aplicaciones).

Capítulo 2: Análisis y diseño del sistema.

En el desarrollo o ciclo de vida de un sistema informático, el diseño del sistema constituye un elemento fundamental del mismo. Es donde se modela el sistema y se encuentra la forma para que soporte todos los requerimientos, incluyendo los requisitos no funcionales, se asienta en el núcleo técnico del proceso de ingeniería del *software* y se aplica independientemente del paradigma de desarrollo utilizado. En la elaboración de la solución es preciso tener presente ciertos estándares, estilos y patrones que son muy útiles en la obtención de un diseño de *software* robusto. (Pressman, 2006)

En este capítulo se presenta el diseño propuesto para la solución a desarrollar. Se presenta un modelo de dominio con el fin de proporcionar un mejor entendimiento de los principales conceptos del negocio, una descripción de las historias de usuario definidas, los requisitos tanto funcionales como no funcionales a tener en cuenta en su desarrollo, agrupados en la lista de reserva del producto, las tarjetas CRC (Clase-Responsabilidad-Colaboración) para describir las clases que se van a utilizar, así como un plan de iteraciones que contempla el tiempo de implementación del sistema.

2.1. Descripción de la solución

La propuesta para la solución de este trabajo de diploma es la implementación de una aplicación web que presente una interfaz sencilla, entendible y fácil de usar, la cual servirá como apoyo para los profesores en la evaluación de los conocimientos de los estudiantes para la asignación de prácticas de laboratorio. La presente investigación utiliza un estudio de caso sobre los estudiantes de un grupo que deseen utilizar el Sistema de Laboratorios a Distancia. El objetivo central del estudio realizado sería determinar si el estudiante posee competencias para realizar prácticas de control. En su desarrollo se utilizarán tecnologías y herramientas de código abierto, la aplicación estará programada en PHP sobre el *framework* de desarrollo Symfony, utilizando como gestor de base de datos PostgreSQL y Apache como servidor web.

2.1.1. Análisis de la estructura del modelo propuesto

Para el manejar el proceso de selección, decisión, interpretación y aplicación del comportamiento que refleja el razonamiento (M, et al., 1994).Procesa e interpreta reglas que se encargan de resolver un problema de decisión (Cortez, et al., 2010).En la lógica clásica, es posible deducirse mediante el empleo de reglas, si su premisa es cierta, también lo será su conclusión. (Gutierrez, 2010) El autor propone para ello un procedimiento para la toma de decisiones, el cual se basa en el enfoque multicriterio (Romero, et al., 1997), (Fernandez, 2000) compuesto por los siguientes pasos:

Paso 1: Identificar los criterios valorativos.

Paso 2: Asignar peso a los criterios.

Paso 3: Asignar criterios a persona.

Paso 4: Evaluar criterios a persona.

Paso 5: Determinar importancia de los criterios evaluados.

Paso 6: Procesar inferencia.

A continuación se describen cada uno de los pasos:

1- Identificar los criterios a valorar.

La identificación de los criterios a valorar, no es más que la selección de los indicadores que se le asignarán a un usuario o alternativa por el cual se le medirá su nivel de competencias.

Debe cumplirse que:

Los criterios asignados sean, $C = \{c_1..c_m\} (n \geq 2)$. El dominio de criterios C sea finito.

2- Asignar peso a los criterios.

Aplicándose un enfoque multiexperto, adquiere importancia la actividad del grupo de evaluadores quienes mediante su interacción diaria con los criterios evaluativos, permitirán refinar el peso asignado a los indicadores (Propuesta de algoritmo de clasificación genética, 2013). Para el autor se entenderá por peso al valor que se le atribuye a la evaluación de un criterio con respecto al resto de los criterios. Uno de los principales aspectos en el análisis de los problemas, es conocer cuál es el criterio más importante y cuánto representa. Los pesos se determinarán a través de las valoraciones de los evaluadores E del proceso de evaluación del desempeño. Estos podrán expresar sus preferencias a través de un valor lingüístico al que se le corresponde un valor numérico. La Tabla 2 muestra el dominio de valores a utilizar.

Valor	Impacto
0,1	Sin importancia
0,2	Poco importante
0,3	Ligeramente importante
0,4	Algo importante
0,5	Importancia media
0,6	Importante
0,7	Muy importante
0,8	Fuertemente importante
0,9	Muy fuertemente importante
1	Extremadamente importante

Tabla 2. Dominio de valores para asignar peso a los indicadores. (Modelo para la evaluación de competencias en un sistema de laboratorios a distancia, 2014)

Tomando como referencia el dominio de valores propuesto en la Tabla 1, cuando los evaluadores emiten sus valoraciones respecto a los criterios, si dos criterios tienen igual valoración indica que ambos

Los criterios son igualmente importantes y si un criterio tiene mayor valor que otro significa que el primero es más importante. Más de un criterio puede tener el mismo valor, el valor cero para algún criterio indica la no importancia del criterio, mientras que el valor más alto indica la máxima importancia para ese criterio.

A cada evaluador E se le pide que emita su opinión acerca de la importancia W que tiene cada indicador con relación a los criterios a evaluar. Los pesos de los indicadores son normalizados de modo que cumplan las siguientes condiciones:

Condición 1: $0 \leq W_j \leq 1$

Condición 2: $\sum_{j=1}^m W_j = 1$

El valor relativo de cada criterio (EP) se obtiene mediante la expresión $EP = \frac{\sum_{i=1}^n W_i}{E}$

El peso P de los criterios se determina con la expresión $P = \frac{W_i}{\sum_{i=1}^n EP}$

3- Asignar criterios a personas.

La asignación de los criterios evaluativos a las personas, consiste en determinar de todos los criterios existentes $C = \{c_1 \dots c_m\} (n \geq 2)$ cuáles corresponden a las funciones que solicita un usuario que se determina en función del área en la que desea trabajar en el Sistema de Laboratorios a Distancias.

4- Evaluar criterios a las personas.

Concluido el proceso de ejecución de las tareas o criterios asignados para el usuario o alternativa, se ha enmarcado el proceso de evaluación de competencias, donde se emite una preferencia P sobre la ejecución de los criterios asignados, para ello se utiliza la escala propuesta en la Tabla 3.

Valor	Impacto
0	Incumplida (Deficiente)
0,25	Cumplida con recomendaciones
0,5	Cumplida (Adecuado)
0,75	Cumplida con méritos
1	Sobre cumplida (Superior)

Tabla 3. Dominio de valores para evaluar criterios. (Modelo para la evaluación de competencias en un sistema de laboratorios a distancia, 2014)

El conjunto de valores propuesto para evaluar los criterios en función de obtener la preferencia del indicador o tarea, es propuesto teniendo en cuenta que:

Máxima preferencia del dominio de valores corresponde a la mayor evaluación
 $P = 1 = \text{Max} = \{ \text{Superior} \}$

Mínima preferencia del dominio de valores corresponde a la menor evaluación
 $P = 0 = \text{Min} = \{ \text{Deficiente} \}$

Media preferencia corresponde a la media aritmética del dominio de valores
 $P = 0,5 = \text{Med} = \{ \text{Adecuado} \}$

Mayor que mínima (MayMin) preferencia del dominio de valores corresponde a la media aritmética entre el valor de Adecuado y Deficiente $P = 0,25 = \text{MayMin} = \{ \text{Cumplida con recomendaciones} \}$. Menor que Máxima (MinMay) preferencia del dominio de valores corresponde a la media aritmética entre el valor de Adecuado y Superior $P = 0,75 = \text{MinMay} = \{ \text{Cumplida con méritos} \}$.

5- Determinar la importancia de los criterios evaluados.

Para la selección de la importancia se utiliza el operador OWA, media ponderada ordenada, Ordered Weighted Averaging (On ordered weighted averaging aggregation operators in multicriteria decision making, 1988). Este modelo unifica los criterios clásicos de decisión en incertidumbre en un solo modelo (Merigo, 2008). Es decir, abarca los criterios optimista, pesimista, el de Laplace y el de Hurwicz en una sola expresión (Leyva, 2014). Se obtiene mediante la siguiente expresión:

Sea el conjunto de alternativas $A = \{a_1 \dots a_n\}$ que representan las personas a evaluar.

Sea el conjunto de criterios $C = \{c_1 \dots c_m\} (n \geq 2)$ que son asignados a las personas.

Importancia o peso W atribuido a los criterios C .

Conjunto de preferencias $P = \{p_{(c1)}, \dots, p_{(cn)}\}$ sobre la evaluación del comportamiento de los criterios C en las alternativas A .

Definición:

- Es una función de importancia $F : R^n \rightarrow R$ de dimensión n que tiene un vector de importancia

W de dimensión n con $W_{ij} \in [0,1]$ y $\sum_{j=1}^n W_j = 1$, de forma tal que:

- $F(p_1, p_2, \dots, p_n) = \sum_{j=1}^n w_j b_j$, donde b_j es el j -ésimo más grande de los p_j ordenados.

Con la aplicación del proceso de agregación de la información, se obtiene mediante la función de importancia F , la importancia del comportamiento de los criterios evaluados para una persona o alternativa.

6- Procesar inferencia.

Una vez obtenida la función de importancia atribuida a las alternativas F^a , se plantea un sistema de representación del conocimiento basado en un árbol de decisión mediante el cual es posible expresar un conjunto de reglas. La Figura 3 muestra el árbol de decisión de inferencias.



Figura 3. Árbol de decisión de inferencia. (Modelo para la evaluación de competencias en un sistema de laboratorios a distancia, 2014)

El árbol se puede expresar mediante el siguiente sistema de regla:

Concluyendo que si la inferencia decisional es verdadera “Sí” entonces el resultado es el valor consultado, de lo contrario se sigue consultando reglas hasta obtener un valor verdadero.

R1: Si $F \geq 0.67$ Entonces evaluación de las competencias= Superior.

R2: Si $F \leq 0.33$ Entonces evaluación de las competencias= Deficiente.

R3: Si $F \leq 0.66$ y $F \geq 0.34$ Entonces evaluación de las competencias= Adecuado.

Después de realizar el proceso de inferencias aplicado a la función de importancia F sobre el conjunto de reglas derivadas del árbol de decisión, se obtiene como resultado la evaluación de las competencias de una persona o alternativa del proceso. Para valores de Superior el usuario podrá acceder a las prácticas de control que solicite. Para valores de Adecuado, el usuario podrá ejecutar las prácticas de

control que solicite condicionado a una cantidad de errores posibles dentro del Sistema de Laboratorios a Distancia en caso de ser agotado requerirá de una nueva capacitación. Para valores Deficiente, el usuario no podrá ejecutar las prácticas de control que solicite.

2.1.2. Descripción de los Objetos del Modelo del Dominio

A pesar de que la metodología de desarrollo de *software XP* no precisa una técnica específica para definir el negocio, con el fin de proporcionar un mejor entendimiento de los principales conceptos que se manejan en este, se decide realizar un modelo de dominio, el cual se describe mediante diagramas de UML, específicamente mediante diagramas de clases.

El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Este permitirá mostrar de manera visual los principales conceptos que se manejan, permitiendo a los usuarios y desarrolladores utilizar un vocabulario común para poder entender el contexto en que se desarrolla el sistema. Además contribuirá a identificar personas, eventos, transacciones y objetos involucrados en el negocio. (Brocks, 2008) El modelo de dominio que será tomado como punto de partida para el diseño del sistema se muestra en la Figura 4.

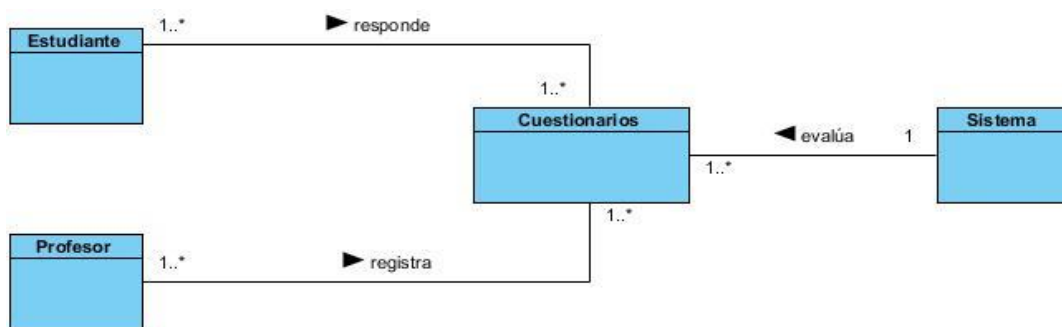


Figura 4 Diagrama de dominio de la solución

En el diagrama se puede observar que el sistema contará con dos usuarios (estudiante y profesor), de los cuales el **Profesor** es el encargado de crear los cuestionarios con el fin evaluar el conocimiento de los estudiantes. Por su parte el **Estudiante** al acceder a la herramienta, tendrá que responder los cuestionarios que permitirá medir su nivel cognitivo en el tema. Así el **Sistema** puede finalmente evaluar y decidir si la práctica de laboratorio que solicita el estudiante se corresponde con su resultado.

Concepto	Descripción
Estudiante	Persona que responde los cuestionarios.
Profesor	Persona encargada de crear los cuestionarios que van a evaluar el conocimiento del estudiante en la materia.
Cuestionarios	Instrumento de evaluación compuesto por una serie de preguntas organizadas de forma lógica con el propósito de obtener información relacionada con el nivel de los conocimientos de un estudiante.
Sistema	Herramienta que presenta un conjunto de funcionalidades que permite llevar a cabo el proceso de evaluación de conocimientos.

Tabla. 3 Descripción de los objetos

Luego de haber analizado la propuesta de la solución, se procede al diseño de la aplicación, el cual comprende desde la especificación de los requisitos, tanto funcionales como no funcionales hasta el llenado de los documentos de artefactos correspondientes con la metodología seleccionada y la definición de las clases.

2.2. Exploración

Es la fase en la que se define el alcance general del proyecto, el cliente define lo que necesita mediante la redacción, a grandes rasgos, de sencillas historias de usuarios que 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 y se estima el tiempo de implementación en base a la información que proporcionan las historias de usuario. También se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

2.2.1. Historias de Usuario

Las Historias de Usuarios (HU) son la forma en que se especifican en XP los requisitos del sistema. Estas se escriben desde la perspectiva del cliente aunque los desarrolladores pueden brindar ayuda en su identificación. El contenido de las mismas debe ser concreto y sencillo.

Las HU también conducen el proceso de los test de aceptación, que son empleados para verificar que las historias han sido implementadas correctamente. Existen diferencias entre las HU y la tradicional especificación de requisitos, siendo su principal diferencia el nivel de detalle. Las historias de usuario solamente proporcionan los detalles sobre la estimación de riesgo y cuánto tiempo conllevará la

implementación de dicha historia. Cuando llega el momento de la implementación, los desarrolladores se reunirán directamente con el cliente para obtener toda una lista de detalles necesarios para satisfacer sus necesidades.

Esta técnica utilizada para especificar los requisitos del *software*. Se registran en tablas en las cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las HU es muy dinámico y flexible. Cada una debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Durante el análisis en la fase de exploración se identificaron un total de 17 HU, a continuación se muestra una HU con prioridad muy alta, para ver las restantes tablas ver anexo 1.


Historia de Usuario	
Número: 1	Nombre de la Historia de Usuario: Autenticar usuario
Cantidad de modificaciones a la Historia de Usuario: Ninguna	
Usuario: Annia Viera Sánchez	Iteración asignada: 1
Prioridad en negocio: MuyAlta	Puntos estimados: 0.6 semanas
Riesgo en desarrollo: Media	Puntos reales: 0.6 semanas
Descripción: El usuario accede a la interfaz principal de la aplicación y se autentica con su usuario y contraseña, si ha llenado los datos correctamente entonces obtiene el permiso de entrar al sistema.	
Observaciones:	
<p>Prototipo de interfaces</p>  <p>The image shows a login form titled 'Iniciar Sesión' on a light blue background. It contains two input fields: 'Usuario' and 'Contraseña'. The 'Contraseña' field is masked with dots. Below the fields is an 'Entrar' button.</p>	

Tabla 5. Descripción HU Autenticar usuario.

2.3. Planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando las HU, las que sustituyen a los tradicionales casos de uso. Una vez obtenidas las HU, los programadores evalúan rápidamente el esfuerzo que costará la implementación de cada HU. Si alguna de ellas tiene riesgos que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba (spikes), para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas (*Release Plan*) en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, donde en cada una de ellas se desarrolla, prueba e instala una pocas HU.

2.3.1. Estimación de esfuerzo por historias de usuarios

Para el desarrollo de la aplicación propuesta se tienen en cuenta situaciones relativas a HU tales como: aparición de nuevas, postergación en cuanto a la iteración en la cual se implementa, no finalización en la iteración planificada y modificación durante la iteración o una vez completada en una iteración previa, realizándose una estimación del esfuerzo para cada una de las HU identificadas, los resultados de este estudio se muestran en la tabla 8:

No. HU	Historia de Usuario	Puntos de estimación
1.	Autenticar usuario	0.6
2.	Adicionar usuario	0.4
3.	Modificar usuario	0.4
4.	Eliminar usuario	0.4
5.	Adicionar preguntas	0.6
6.	Modificar preguntas	0.6
7.	Eliminar preguntas	0.6
8.	Adicionar respuestas	0.8
9.	Adicionar cuestionario	0.8
10.	Modificar cuestionario	0.4

11.	Eliminar cuestionario	0.4
12.	Proponer preguntas a expertos	1.4
13.	Publicar cuestionario	1.2
14.	Ver cuestionarios propuestos	0.4
15.	Realizar cuestionario	1
16.	Evaluar cuestionario	1.6
17.	Rechazar cuestionario propuesto	0.4

Tabla 9. Estimación de esfuerzo por HU.

2.3.2. Plan de iteraciones

El Plan de Iteraciones se elabora para planificar el tiempo que se demora en desarrollar todas las funcionalidades necesarias para llevar a cabo una iteración completa. En este artefacto se describe lo que se realiza en cada iteración de manera general y se especifica los requisitos funcionales que se implementan en estas, determinando la duración total para ello. También es importante definir la prioridad en el negocio para las funcionalidades ya que de esto depende las iteraciones que se van a desarrollar. En la Tabla. 9 se representa el Plan de Iteraciones confeccionado para la aplicación que se corresponde con el presente trabajo de diploma.

No	Descripción de la iteración	Historias a implementar	Duración total
-----------	------------------------------------	--------------------------------	-----------------------

1	Se desarrollan las HU más importantes para el negocio relacionadas con la autenticación de usuarios al sistema y las relaciones entre ellos, así como la evaluación de cuestionarios, proposición de cuestionarios entre los expertos y se analizan los posibles pesos de las mismas	1,12,13,16	4.8 semanas
2	Se desarrollan las funcionalidades que permiten la gestión de los usuarios, preguntas y respuestas sobre el sistema, además de la elaboración y realización de los cuestionarios por parte de los usuarios.	2,3,4,5,6,7,8,9,15	5.6 semanas

No	Descripción de la iteración	Historias a implementar	Duración total
3	Se desarrollan las funcionalidades para manejar la realización del cuestionario y para modificar, eliminar después de creado en el sistema, además ver los cuestionarios que se le han propuesto a un experto para su valoración.	10,11,14	1,2 semanas
4	Se desarrollan la funcionalidad para rechazar un cuestionario propuesto a un experto.	17	0.4 semana

Tabla 10. Plan de Iteraciones

2.3.3. Plan de entregas.

A continuación se presenta el plan de entregas ideado para la fase de implementación. Como producto del mismo se harán entregas del sistema al finalizar cada iteración en la fecha aproximada que se indica en la siguiente tabla.

Sistema	Fin iteración 1 (Febrero 2015)	Fin iteración 2 (Abril 2015)	Fin iteración 3 (Mayo 2015)	Fin iteración 4 (Mayo 2015)
Módulo para la evaluación de competencias antes de la realización de una práctica de laboratorio de un Sistema de Laboratorio a Distancia.	1-Autenticar usuario 2-Proponer preguntas a expertos 3-Publicar cuestionario 4-Evaluar cuestionario	1-Adicionar usuario 2-Modificar usuario 3-Eliminar usuario 4-Adicionar preguntas 5-Modificar preguntas 6-Eliminar preguntas 7-Adicionar cuestionario 8-Realizar cuestionario 9-Adicionar respuestas	1-Modificar cuestionario 2-Eliminar cuestionario 3-Ver cuestionarios propuestos	1-Rechazar cuestionario propuesto

Tabla 11. Plan de entregas.

2.4. Requerimientos de software

Los requerimientos para un sistema de *software* determinan lo que hará el sistema y definen las restricciones de su operación e implementación. Los mismos se clasifican en funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir y se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

2.4.1 Características funcionales del sistema

Las características funcionales son características requeridas del sistema que expresan una capacidad de acción del mismo – una funcionalidad; generalmente expresada en una declaración en forma verbal. Para describir los servicios que se espera que el sistema propuesto suministre las funcionalidades se detallan en las HU del presente capítulo.

2.4.2 Características no funcionales del sistema

Para definir los requisitos no funcionales se tienen en cuenta aspectos como la usabilidad, la seguridad, el *hardware*, la eficiencia, la fiabilidad, el soporte y las restricciones del diseño, que determinan condiciones y características a tener en cuenta para el sistema. A continuación se muestran los requisitos no funcionales.

Requisitos No Funcionales	
1	Portabilidad: La aplicación será un sistema multiplataforma, permitiendo el despliegue del mismo en cualquier sistema operativo.
2	Seguridad: El sistema garantizará la seguridad mediante la autenticación de usuarios y sus correspondientes contraseñas, es decir, solo los usuarios autorizados por el administrador podrán entrar en la aplicación y cada uno de ellos podrá realizar las operaciones que sus permisos le permitan.
3	Software: Para poder hacer uso de este sistema debe existir un servidor de aplicaciones que tenga instalado un servidor web (recomendado Apache v 2.2), un servidor de base de datos PostgreSQL v 9.1 o superior, debe estar instalado el lenguaje PHP v 5.3 o superior y en las máquinas clientes, un navegador web (recomendado Mozilla Firefox en su versión 27 o superior).
4	Hardware: La computadora que se utilizará como servidor debe contener como mínimo 2 GB de memoria RAM, 2 GB de espacio libre en Disco Duro, y un microprocesador superior a los 2.0 GHz. Los clientes deberán tener como mínimo 1 GB de RAM y un microprocesador superior a los 1.6 GHz.
5	Apariencia o interfaz externa: La aplicación debe presentar una interfaz amigable, intuitiva y fácil de usar para los usuarios. Debe ser compatible con resolución 1024x768 píxeles o superior.
6	Disponibilidad: La aplicación debe estar disponible a toda hora.

Tabla 12.Requisitos no funcionales.

2.5. Roles del sistema

El sistema después de creado puede ser utilizado por cualquier persona (estudiante o profesor) que pertenezca a la carrera de Ingeniería Automática, por esta razón se estableció los roles denominados profesor y estudiante, como se muestra en la siguiente tabla. Para la administración de los usuarios se definió un rol administrador que es el encargado de la creación de los usuarios del sistema.

Actor	Descripción
Estudiante	Representa al rol encargado de la realización de los cuestionarios.
Profesor	Representa el rol encargado de la gestión de cuestionarios.
Administrador	Representa el rol encargado de la gestión de usuarios.

Tabla 13. Roles del Sistema.

2.6. Diseño

El diseño es el sitio donde manda la creatividad, donde los requisitos del cliente, las necesidades de negocio y las consideraciones técnicas se unen en la formulación de un producto o sistema. Crea una representación o modelo del *software*, pero a diferencia del modelo de análisis, el modelo de diseño proporciona detalles acerca de las estructuras de datos, la arquitectura, las interfaces y los componentes del *software* que son necesarios para implementar el sistema (Pressman, 2006). En el presente epígrafe se exponen los elementos relevantes para el diseño del módulo.

2.6.1. Diagrama de Clases

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de *software* y de las interfaces en una aplicación. Normalmente este diagrama contiene información referente a las clases, asociaciones, atributos y métodos. El diagrama de clases contiene las definiciones de las entidades del *software*. (Larman, 2004)

Este artefacto no es propio de la metodología XP, si lo es de las metodologías robustas como RUP (Rational Unified Process), pero aprovechando que una de sus características principales es la flexibilidad, se ha decidido representar dicho diagrama, puntualizando lo antes descrito en las tarjetas CRC. Se decide incorporarlo para mostrar más detalladamente y de una manera entendible la solución del problema de la investigación. En la Figura 5 se muestra el diagrama de clases en el cual se muestran las clases utilizadas para la implementación de la solución, así como las relaciones existentes entre ellas. En el diagrama de clases se describen las clases utilizadas para llevar a cabo la implementación del sistema, se cuenta con un total de 8 clases las cuales se encuentran relacionadas.

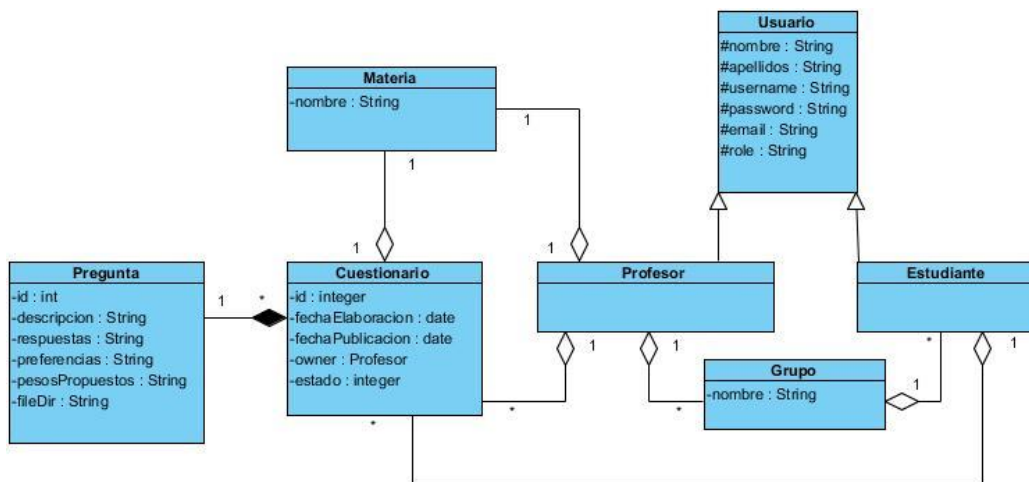


Figura 5 Diagrama de clases de la solución.

2.6.2. Arquitectura base de la aplicación

“La arquitectura del software de un programa es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellas.” (Reynoso, 2013). Esta tiene gran importancia debido a que durante el desarrollo del sistema, provee una excelente vista general del mismo, proporciona la relación de puntos de diseño a tratar, facilita el desarrollo simultáneo de componentes y permite detectar errores de diseño en fases tempranas. La arquitectura base de la aplicación correspondiente a este trabajo de diploma está constituida por patrones de arquitectura y de diseño que serán presentados a continuación.

Modelo-Vista-Controlador es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde las páginas web pertenecen a la vista, lo relacionado con el acceso a datos y la lógica del negocio pertenece al modelo, y el controlador es el encargado de recibir los eventos de entrada y responder a los mismos. Se puede describir el funcionamiento del patrón de la siguiente forma: el usuario realiza una petición al servidor, el controlador recibe la petición, consulta al modelo para buscar y obtener los datos en caso de ser necesario y luego devuelve la vista al usuario. (De León Alvarez, 2011)

MVC divide las aplicaciones en tres niveles de abstracción:

Modelo: Representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como “intermediario” con la base de datos.

Vista: Es la encargada de mostrar la información al usuario de forma gráfica y “humanamente legible”.

Controlador: Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma “humanamente legible” (Mestras, 2004). En la Figura 6 puede observarse la estructura de la aplicación que será desarrollada siguiendo el patrón MVC.

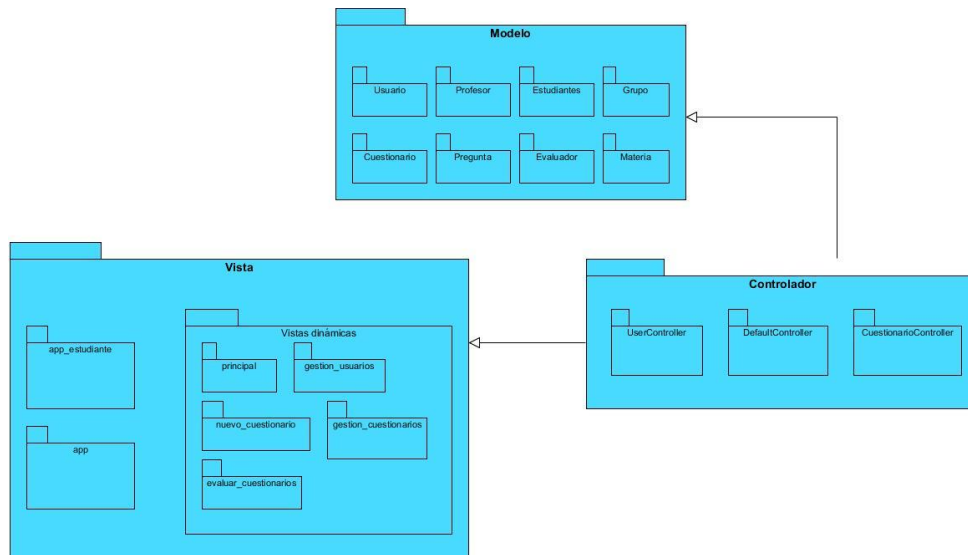


Figura 6 Patrón Arquitectónico MVC.

2.6.3. Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. A través de la siguiente figura se puede observar el funcionamiento de esta arquitectura:

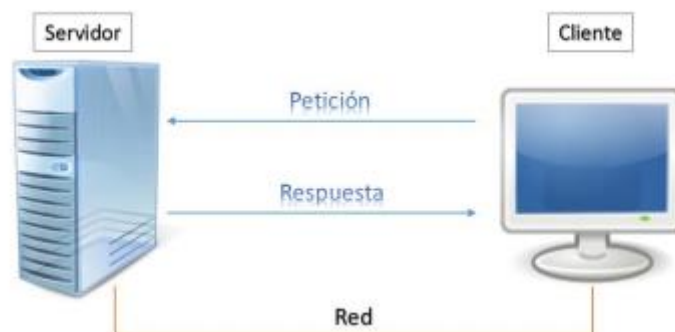


Figura 7 Arquitectura Cliente Servidor.

Se decide utilizar la arquitectura Cliente-Servidor como se muestra en la Figura 7 porque se está haciendo uso de un SLD para el desarrollo del módulo de evaluación, donde los laboratorios están almacenados en un servidor y los estudiantes o los profesores para poder acceder a los cuestionarios pueden hacerlo desde cualquier computadora que esté conectada a la red.

2.6.4. Patrones de diseño

Los patrones de diseño brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. (Tedeschi, 2014). Es importante tener en cuenta estos patrones a la hora de realizar el diseño de una aplicación ya que son soluciones simples que indican cómo resolver un problema particular utilizando un pequeño número de clases relacionadas de forma determinada. Además facilitan la reutilización de clases ya existentes y la programación de clases reutilizables. (Yupanqui García, 2013)

2.6.5. Patrones GRASP

Los patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades) son patrones generales de *software*, que describen los principios fundamentales de diseño de objetos para asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto, aplicando el razonamiento para hacerlo de una forma sistemática, racional y explicable. Las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento. Dichas responsabilidades pueden ser de dos tipos: Conocer y Hacer. (Saavedra, 2007)

A continuación se describen los patrones GRASP que se usaron en el desarrollo de la aplicación:(Craig, 2003)

Creador: consiste en asignarle a una clase determinada la responsabilidad de crear una instancia de otra clase, es decir, guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.

En la Figura 8 se evidencia el patrón creador con la clase Cuestionario. Esta clase es la responsable de crear las instancias de Pregunta.

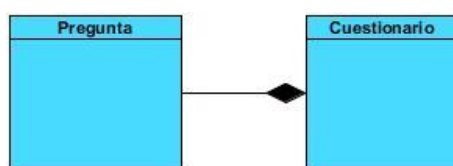


Figura 8 Aplicación del patrón Creador.

Bajo acoplamiento: consiste en asignar una responsabilidad para mantener el bajo acoplamiento. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios y también más reutilizables que acrecientan la oportunidad de una mayor productividad.

En la Figura 9 se evidencia el patrón bajo acoplamiento ya que la clase Cuestionario es la que cuenta con la información necesaria para que la clase Evaluador pueda realizar su responsabilidad.

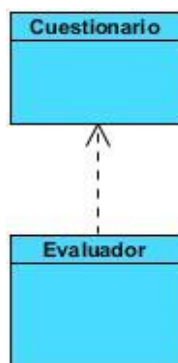


Figura 9 Aplicación del patrón Bajo acoplamiento.

Alta cohesión: consiste en asignar una responsabilidad de modo que la cohesión siga siendo alta. Es un patrón evaluativo que el diseñador aplica al valorar sus decisiones de diseño. La utilización de este patrón mejora la calidad y facilidad con que se entiende el diseño.

En la Figura 10 se evidencia el patrón alta cohesión al existir una estrecha relación entre las clases cuestionarioController y Cuestionario.

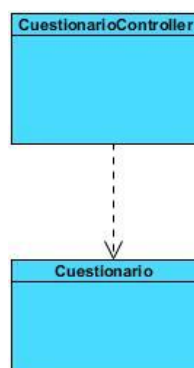


Figura 10 Aplicación del patrón Alta Cohesión.

Controlador: Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. En la aplicación se evidencia este patrón de diseño en las clases Controladora y sus hijas las cuales se encargan de controlar el flujo de información dentro del sistema.

En la Figura 11 se evidencia el patrón controlador al ser la clase Cuestionario Controller quien tiene todo el proceso de control sobre las clases nuevo_cuestionario y Cuestionario.

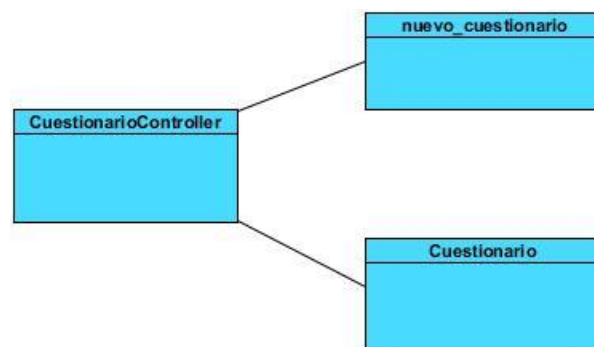


Figura 11 Aplicación del patrón Controlador.

2.6.6. Tarjetas Clase, Responsabilidad y Colaboración.

Para el diseño de las aplicaciones, la metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC.

Estas tarjetas se dividen en tres secciones, las cuales contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores, de ahí proviene su nombre Clase-Responsabilidad-Colaboración (CRC). Una clase es cualquier persona, evento, objeto, concepto, pantalla o reporte. Las responsabilidades de una clase es todo aquello que conoce y que realiza, sus atributos y métodos, mientras que los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades. Resulta muy útil tener pequeñas tarjetas de cartón para llenarlas y mostrárselas al cliente con el objetivo de llegar a un acuerdo sobre la validez de las abstracciones propuestas (Wesley, 2014). Para el desarrollo de la solución se debe elaborar una tarjeta por cada clase presente y en este caso se identificaron un total de 8 tarjetas CRC. La Tabla 14 que se muestra a continuación, representa la Tarjeta CRC para la clase Usuario, mientras que el resto de las tarjetas se pueden encontrar en el anexo 2.

Tarjeta CRC

Clase: Usuario	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none">- Devolver cada uno de los atributos de los usuarios (usuario, nombre, rol, apellidos, email, contraseña).- Cambiar valor de cada uno de los atributos de los usuarios (usuario, nombre, rol, apellidos, email, contraseña).- Verificar que el usuario y la contraseña sean correctos para la autenticación.	Profesor Estudiante

Tabla 14. Tarjeta CRC de la clase "Usuario".

Después de haber elaborado todos los artefactos indicados para documentar el proyecto de aplicación del modelo de búsqueda, se hace necesario representar gráficamente la relación entre todas las clases implementadas para la solución, y así de esta forma definir con mayor claridad la arquitectura de la aplicación que se desarrolla. En este trabajo de diploma se ha determinado usar el diagrama de clases para mostrar dicha arquitectura.

Conclusiones parciales del capítulo

- El análisis realizado en este capítulo indica que con un mejor entendimiento del flujo actual de las actividades y el objeto de automatización se crean las bases para la comprensión del sistema a implementar. La descripción de las HU, como paso fundamental en el ciclo de desarrollo del proyecto, ha contribuido a la identificación de las características funcionales y no funcionales del sistema, los cuales sirven de base para la generación de un conjunto de artefactos necesarios en fases posteriores.
- En este capítulo se llevó a cabo la especificación de los requisitos a partir de las 17 historias de usuario identificadas, determinándose 17 características funcionales y seis no funcionales. Se definieron cuatro iteraciones para el desarrollo de la solución y ocho tarjetas CRC, lo cual permitió precisar los elementos necesarios para la implementación del sistema, que posteriormente será realizada. Se utilizó el patrón de arquitectura Modelo-Vista-Controlador, el patrón Cliente-Servidor y los patrones de diseño Controlador, Creador, Alta Cohesión y Bajo Acoplamiento para la definición de la arquitectura base del sistema, logrando mayor calidad y robustez en la estructura del mismo.

Capítulo 3. Implementación y Pruebas.

Los objetivos principales de la fase de implementación son el desarrollo de las clases especificadas durante el diseño, definir la organización del código y la puesta en servicio del sistema construido de manera que se consiga su adaptación final por parte de los usuarios del mismo, comprobando mediante demostraciones formales (como las pruebas de aceptación) que este cumple todos los objetivos y requisitos para los que fue desarrollado. (Suárez, 2012) En este capítulo se exponen las tareas de ingeniería definidas para cada historia de usuario con vistas a su implementación. Se presenta además el estándar de codificación utilizado, así como una descripción de las pruebas realizadas a la solución y los resultados de su ejecución.

3.1. Implementación de la aplicación

Una vez diseñada la aplicación y definida la arquitectura de esta, se debe especificar las tareas de ingeniería, que permitirán organizar el proceso de desarrollo de cada historia de usuario con el objetivo de llevar a cabo la implementación de una manera más sencilla.

3.1.1. Tareas de la Ingeniería

El equipo de desarrollo tiene la responsabilidad de revisar, analizar y evaluar cada una de las funcionalidades o HU correspondientes a la aplicación que se desea elaborar. Esta tarea es realizada con el objetivo de dividir cada una de las HU en tareas más pequeñas, que de conjunto tributen al correcto desempeño de la funcionalidad.

Las tareas de ingeniería (TI) describen las tareas realizadas sobre el proyecto. Cada tarea está relacionada a una HU, pero una HU puede ser dividida en varias tareas de ingeniería. En estas se especifica la fecha de inicio y fin de la tarea y el programador responsable de cumplirla así como la estimación en semanas para darle cumplimiento a la misma. En la tabla 15 se muestra una tarea de ingeniería para la "Autenticar usuario". Para la aplicación que se desea desarrollar se determinaron un total de 41 TI, las restantes pueden consultarse en el anexo 3.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario:1
Nombre Tarea: Definir lista de control de acceso.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.1 semanas

Fecha Inicio: 2/02/2015	Fecha Fin: 2/02/2015
Programador Responsable: Jeffrey Díaz Mora	
Descripción: Se define la lista de control de acceso donde se especifica a qué URLs tendrán acceso los diferentes roles del sistema. Esto se define en el fichero security.yml	

Tabla 15. Tarea de Ingeniería número

3.1.2. Estándares de codificación

El código de una aplicación debe estar implementado de forma tal que su lectura y su modificación por cualquier miembro del equipo de desarrollo resulte lo más fácil posible. Para ello es necesario seguir estándares de codificación que permitan a todo el mundo sentirse cómodo con el código escrito por otra persona. El objetivo fundamental de un estándar de codificación es proporcionar una arquitectura y un estilo consistente, independiente del autor, con lo cual el sistema sea más fácil de entender y por supuesto más fácil de mantener. Debe cumplir con las siguientes características: (Guía para Implementar Estándares de Codificación, 2012)

- Clarificar más que confundir.
- Promover la intención del código.
- Permitir que los programas se acerquen lo mejor posible al lenguaje natural.
- Incorporar las mejores prácticas de la codificación.

Existen actualmente varios estándares de codificación para PHP, que es el lenguaje de programación seleccionado para implementar la solución de este trabajo de diploma, entre los que se pueden nombrar PSR0, PSR1, PSR2, PSR3, PEAR Coding Standards y Zend Coding Standards (Estándares de codificación en PHP, 2013). A continuación se explican los estándares de codificación utilizados en la aplicación.

Convenciones de Nombres:

- Los nombres de las clases utilizan CamelCase, con la letra inicial en mayúscula.
- Los nombres de las variables, propiedades, funciones y métodos son camelCase, con la inicial en minúscula.

```
class UserController extends Controller
{
    public function getUsersAction()
    {
        $users = $this->getDoctrine()->getRepository("CoreBundle:Usuario")->findAll();

        $response = array();
        foreach($users as $user)
        {
            $response[] = array
            (
                "nombre"=>$user->getNombre(),
                "apellidos"=>$user->getApellidos(),
                "email"=>$user->getEmail(),
                "username"=>$user->getUsername()
            );
        }
        return new JsonResponse($response);
    }
}
```

Figura 12 Convenciones de nombres.

Convenciones de llaves:

- Las funciones y clases tienen su llave inicial en la misma línea de la declaración.
- Todas las estructuras de control usan llaves y nunca sentencias en línea.
- Las estructuras de control tienen la llave inicial en la misma línea de la estructura.

```
class DefaultController extends Controller
{
    public function indexAction(Request $request)
    {
        $error = null;
        $session = $request->getSession();
        if($request->attributes->has(SecurityContext::AUTHENTICATION_ERROR))
        {
            $error = $request->attributes->get(SecurityContext::AUTHENTICATION_ERROR);
        }
        else
        {
            $error = $session->get(SecurityContext::AUTHENTICATION_ERROR);
            $session->remove(SecurityContext::AUTHENTICATION_ERROR);
        }
        return $this->render('CoreBundle:Default:index.html.twig', array
        (
            'error' => $error,
            'last_user' => $session->get(SecurityContext::LAST_USERNAME)
        ));
    }
}
```

Figura 13 Convenciones de llaves.

Indentación:

- En los lenguajes de programación, la indentación es una forma de notación secundaria que mueve un bloque de texto hacia la derecha insertando espacios o tabuladores, lo cual permite mejorar la legibilidad del código fuente para que sea más fácil su entendimiento. En este caso no se hace uso de tabuladores, se usa solamente cuatro espacios por nivel de indentación.

```
public function appAction()
{
    return $this->render("CoreBundle:Default:app.html.twig");
}

public function appEstudianteAction()
{
    return $this->render("CoreBundle:Default:app_estudiante.html.twig");
}
}
```

Figura 14 Indentación.

Programación orientada a objetos:

- Todos los métodos y atributos de una clase tienen especificados explícitamente el nivel de visibilidad.

```
class Usuario implements UserInterface, \Serializable
{
    protected $id;
    protected $nombre;
    protected $apellidos;
    protected $email;
    protected $usuario;
    protected $pass;
    protected $role;

    public function getId()
    {
        return $this->id;
    }

    public function setNombre($nombre)
    {
        $this->nombre = $nombre;
        return $this;
    }
}
```

Figura 15 Programación Orientada a Objetos.

Comentarios:

- Los comentarios son de gran ayuda para aclaración del código pero podrían llegar a estorbar si no se usan bien. En este caso se utilizaron para las partes relevantes del código, usándose los comentarios de una línea y los de bloque.

```
/**
 * @ORM\OneToMany(targetEntity = "MCI\CoreBundle\Entity\Cuestionario", mappedBy = "owner")
 */
```

Figura 16 Comentarios de Bloque.

```
//Adicionar nuevo usuario
public function addUserAction(Request $request)
{
    $json = json_decode($request->getContent(), true);

    if($request->isXmlHttpRequest())
    {
        $manager = $this->getDoctrine()->getManager();
        $user = new Profesor();
        $user->setNombre($json["nombre"]);
        $user->setApellidos($json["apellidos"]);
        $user->setUsuario($json["username"]);
        $user->setEmail($json["email"]);
        $user->setPass($json["password"]);
        $user->setRole("ROLE_USER");

        $manager->persist($user);
        $manager->flush();

        return new JsonResponse(array("success"=>"funciona"));
    }
    return new JsonResponse(array("success"=>"no funciona"));
}
```

Figura 17 Comentarios de línea.

Otros:

- La longitud de línea se mantiene entre los 75 y 85 caracteres, como máximo.

```
{
    $response[] = array
    (
        "nombre"=>$user->getNombre(),
        "apellidos"=>$user->getApellidos(),
        "email"=>$user->getEmail(),
        "username"=>$user->getUsername()
    );
}
```

Figura 18 Longitud de línea.

3.1.3. Interfaces principales de la aplicación.

Al realizarse el proceso de implementación siguiendo estos estándares de codificación se obtuvieron las interfaces de usuario, de las cuales se muestran las principales.

Cuestionarios Almacenados

Sin publicar 11		Publicados 0	En revisión 4	Crear Cuestionario			
Cuestionario 4	Fecha Elaboracion: 12-05-2015	Editar	Eliminar	Proponer Cuestionario	Publicar Cuestionario		
Cuestionario 11	Fecha Elaboracion: 17-05-2015	Editar	Eliminar	Proponer Cuestionario	Publicar Cuestionario		
Cuestionario 12	Fecha Elaboracion: 19-05-2015	Editar	Eliminar	Proponer Cuestionario	Publicar Cuestionario		
Cuestionario 13	Fecha Elaboracion: 19-05-2015	Editar	Eliminar	Proponer Cuestionario	Publicar Cuestionario		
Cuestionario 20	Fecha Elaboracion: 21-05-2015	Editar	Eliminar	Proponer Cuestionario	Publicar Cuestionario		

Figura 19 Interfaz de administración de cuestionarios.

En la Figura 19 se puede observar la interfaz donde el profesor puede gestionar los cuestionarios que ha creado, así como publicarlos. El profesor después de acceder al sistema puede crear sus cuestionarios de su materia para evaluar las habilidades de sus estudiantes. Cada profesor tiene su perfil donde puede encontrar los cuestionarios que han sido confeccionados con su fecha de elaboración, conjuntamente con las opciones de “Editar”, “Eliminar”, “Proponer Cuestionario” y “Publicar Cuestionario”.

Edición de Cuestionarios

Preguntas del Cuestionario Guardar

No ha insertado ninguna pregunta en el cuestionario.


+ Adicionar Pregunta Cerrar

Figura 20 Interfaz para crear un nuevo cuestionario.

Edición de Cuestionarios

Preguntas del Cuestionario Guardar

Much like the company itself, Apple's interview process has minimal beaucracy The interviewers will be looking for excellent technical skills, but a passion for the position and company is also very important While it's not a prerequisite to be a Mac user, you should at least be familiar with the system



opcion1
 opcion 2

Editar Eliminar

+ Adicionar Pregunta Cerrar

Figura 21 Interfaz de edición de cuestionarios.

El profesor después de acceder al sistema y de seleccionar la opción “Crear Cuestionario” debe de confeccionar las preguntas del mismo como se muestra en las Figuras 20 y 21. Un cuestionario está formado por una serie de preguntas con propuestas de solución, estas pueden ser insertadas, dando clic en el botón “Adicionar preguntas”, después de confeccionadas se pueden “Editar” y “Eliminar”. El sistema le permite al profesor después de insertadas las preguntas al cuestionario, dando clic en el botón “Guardar”.

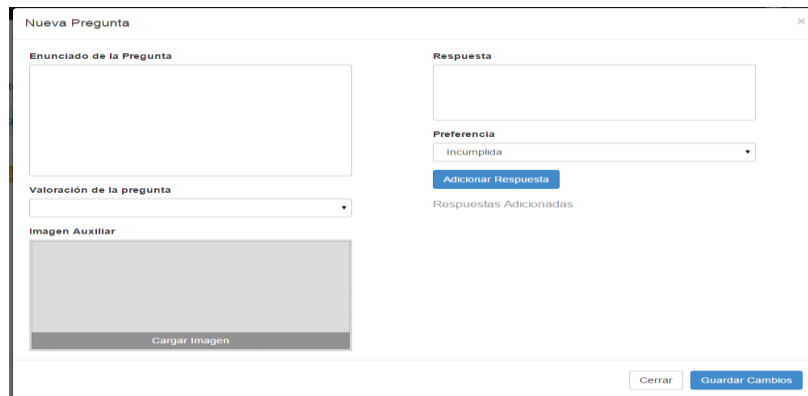


Figura 22 Interfaz para adicionar una pregunta.

En la Figura 22 se puede observar la interfaz donde el profesor confecciona las preguntas del cuestionario. Una pregunta puede generar varias respuestas para esto se adicionó el botón “Adicionar respuestas”. Cada pregunta debe ser valorada por el profesor en dependencia de su complejidad (“Valoración de la pregunta”), al igual que la respuesta debe tener una preferencia que estime su resultado por parte del estudiante (“Preferencia”).

3.2. Proceso de pruebas de la metodología XP

La fase de prueba es muy importante dentro del proceso de desarrollo de *software* ya que durante este resulta muy común que se presenten errores, provocando que el producto final contenga fallos. Las pruebas de *software* suelen ser costosas, sin embargo los costos de las fallas de un *software* en operación pueden llegar a ser mucho mayores.

Esta etapa tiene como objetivo detectar el máximo de errores existentes tanto en el código que se implementa como en cualquiera de las fases para garantizar la calidad del producto desarrollado. Existen diferentes formas de realizar el proceso de pruebas, teniendo en cuenta la metodología de desarrollo utilizada.

Según la metodología XP se debe ser muy estricto con las pruebas. Antes de liberar una nueva versión es necesario asegurarse de que esta ha pasado con el cien por ciento de la totalidad de las pruebas, en caso contrario se empleará el resultado de estas para identificar el error y solucionarlo con mecanismos

definidos para ello. XP propone la realización de dos tipos de pruebas: las unitarias y las de aceptación. (Crispin, 2003)

3.2.1. Tipos de pruebas.

Pruebas unitarias

Una prueba unitaria es la verificación de un módulo (unidad de código) determinado dentro de un sistema. Las pruebas unitarias nos aseguran que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema.

Los programadores realizan estas pruebas cuando: la interfaz de un método no es clara, la implementación es complicada, para probar entradas y condiciones inusuales, luego de modificar algo. Éstas deben contemplar cada módulo del sistema que pueda generar fallas. Para poder integrar el código realizado al ya existente, el mismo debe aprobar satisfactoriamente todos los casos de prueba definidos.

En XP los programadores deben escribir las pruebas unitarias para cada módulo antes de escribir el código. No es necesario escribir casos de prueba para todos los módulos, sólo para aquellos en que exista la posibilidad de que puedan fallar. (Crispin, 2003)

Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada historia de usuario, y tienen como objetivo asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellas. En efecto, las pruebas de aceptación corresponden a una especie de documento de requerimientos en XP, ya que marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención.

Las pruebas de aceptación permiten al cliente saber cuándo el sistema funciona, y que los programadores conozcan que es lo que resta por hacer.

El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de las pruebas sean correctos. Asimismo, en caso de que fallen, deben indicar el orden de prioridad de resolución.

Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados

de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. (Pruebas del sistema en Programación Extrema, 2012)

Pruebas de regresión

Otro tipo de prueba, además de las definidas por XP, es la prueba de regresión, la cual es de gran importancia aplicar para asegurar que un sistema está libre de fallos. Su nombre se debe al hecho de hacer nuevas pruebas donde se ha probado antes. Se realizan generalmente después de la corrección de un error o después de la adición de nuevas funcionalidades con el objetivo de verificar que ningún defecto se añadió al sistema después de la modificación. (Pruebas de Regresión, 2004)

3.3. Técnicas de prueba

Existen dos técnicas para probar cualquier producto de *software*, una es realizando pruebas de forma que se compruebe que cada función es operativa, conocida como pruebas de caja negra y la otra es desarrollando pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, las llamadas pruebas de caja blanca. (Técnicas de Pruebas de software, 2005)

3.3.1. Pruebas de caja blanca

Estas pruebas intentan garantizar que se ejecutan al menos una vez todos los caminos independientes de cada módulo, que se utilizan las decisiones en su parte verdadera y en su parte falsa, que se ejecuten todos los bucles en sus límites y se utilizan todas las estructuras de datos internas. Una prueba de caja blanca que se realice con profundidad permitiría tener programas cien por ciento correctos, pero en la práctica esto implicaría un estudio demasiado exhaustivo, que prolongaría excesivamente los planes de desarrollo del *software*, entonces se deben estudiar sólo los caminos lógicos importantes. (Pruebas del software "Caja Blanca" y "Caja Negra", 2008)

3.3.2. Pruebas de caja negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del *software*, obviando el comportamiento interno y la estructura del programa. Esta técnica de pruebas pretenden encontrar errores como:

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

3.3.3. Método de caja negra

Las pruebas de caja negra comprenden diferentes técnicas como es la técnica de partición equivalente. Esta técnica divide el dominio de entrada de un programa en clases de datos, a partir de las cuales

deriva los casos de prueba. La partición equivalente se enfoca en definir casos de prueba para descubrir clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Se define una condición de entrada (valor numérico específico, rango de valores, conjunto de valores relacionados o condición lógica). Cada una de las clases de equivalencia representa a un conjunto de estados válidos o inválidos para las condiciones de entrada. (Perez Hidalgo, 2015)

Una vez analizado los elementos necesarios para realizar las pruebas en XP, se ha determinado de qué forma se va a llevar a cabo este proceso para el presente trabajo de diploma. Para ello se ha tenido en cuenta que el desarrollo de la solución para aplicar el módulo para la evaluación está enfocado en una programación ágil y la entrega temprana de la primera versión funcional de la aplicación, haciéndose necesario que cada una de sus interfaces presenten la calidad requerida, que no contenga errores de estructura de datos u otro tipo y principalmente que todas las funcionalidades previstas en las fases más tempranas del ciclo de vida del *software* se ejecuten correctamente. Bajo estas condiciones se decidió que la forma más adecuada de realizar el proceso de pruebas es a través de las pruebas de aceptación y las de regresión, usando como método las pruebas de caja negra y como técnica el de partición equivalente. Así, de esta manera el cliente podrá evaluar si el producto cumple con los requisitos propuestos.

3.3. Descripción del proceso de pruebas

El proceso de pruebas está basado en la verificación del funcionamiento del sistema bajo un conjunto de condiciones o variables, lo cual será realizado mediante los casos de pruebas. En el presente trabajo se ha diseñado un caso de prueba para cada historia de usuario definida en la fase de diseño, y en cada uno de ellos se tiene una entrada conocida para la cual existe una salida conocida.

Los casos de pruebas están constituidos por todos los escenarios generales que pueden presentarse durante la ejecución de las historias de usuario. Para cada escenario se define el flujo central de operaciones, que representa los pasos a seguir para poder realizar la historia de usuario correspondiente, se definen además las entradas y las salidas y así se puede verificar, siguiendo el flujo central, que al introducir las variables de entrada, el sistema brinde la respuesta adecuada.

A continuación, en la Tabla. 16 se muestra un ejemplo de caso de prueba perteneciente a la historia de usuario "Autenticar usuario". En total se diseñaron 19 casos de prueba, el resto se pueden consultar en el anexo 4.

Condiciones de ejecución

- El usuario que intenta autenticarse debe haber sido adicionado previamente en el sistema.

Escenario	Descripción	Variable 1	Variable 2	Respuesta del sistema	Flujo central
EC 1.1 Autenticar un usuario introduciendo todos los datos correctamente.	El usuario entra al sistema.	V (cualquiera de los usuarios registrados en el sistema)	V (contraseña del usuario que se intenta autenticar)	El sistema muestra la interfaz principal al usuario, dándole acceso a las opciones sobre las que tiene permiso.	1- Acceder a la url donde se encuentra la aplicación. 2- Introducir los datos. 3- Seleccionar el botón Entrar.
EC 1.2 Autenticar un usuario introduciendo datos incorrectos.	El usuario no puede entrar en el sistema.	I (cualquier nombre de usuario que no esté registrado)	I (contraseña no registrada)	El sistema muestra el mensaje "Error. Inténtelo de nuevo".	1- Acceder a la url donde se encuentra la aplicación. 2- Introducir los datos. 3- Seleccionar el botón Entrar.
EC 1.3 Autenticar un usuario con datos nulos.	El usuario no puede entrar en el sistema.	I ()	I ()	El sistema muestra el mensaje "Existen campos vacíos"	1- Acceder a la url donde se encuentra la aplicación. 2- Introducir los datos. 3- Seleccionar el botón Entrar.

Tabla 16. Caso de Prueba “Autenticar usuario”

Para poder probar cada uno de los escenarios de los casos de prueba diseñados se utilizan variables específicas. En la Tabla. 17 se muestran las variables para el caso de prueba “Autenticar usuario”.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
Variable 1	usuario	campo de texto	No	Nombre (único) con el cual se va a identificar el usuario en el sistema.
Variable 2	contraseña	password	No	Clave que le permite al usuario entrar en el sistema.

Tabla 17. Variables para el caso de prueba “Autenticar usuario”

Durante la realización de las pruebas al *software*, normalmente se detectan una serie de errores que deben ser corregidos. Las no conformidades pueden verse en la Tabla. 18, donde se muestran los fallos encontrados.

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Clasificación	Estado NC	Respuesta del Equipo Desarrollo
Aplicación	1	El nombre del usuario acepta caracteres numéricos como caracteres válidos.	Adicionar usuario	Prueba	NS	4/05/2015 PD 5/05/2015 RA	Se corrigió el error encontrado. El sistema no acepta la entrada de caracteres numéricos en los nombres de usuario.

Capítulo 3 Implementación y Pruebas

Aplicación	2	El sistema permite adicionar un usuario con datos nulos.	Adicionar usuario	Prueba	S	4/05/2015 PD 5/05/2015 RA	Se corrigió el error encontrado. El sistema valida que todos los campos estén llenos para poder adicionar un usuario.
Aplicación	3	El enunciado de la pregunta del cuestionario acepta caracteres extraños como caracteres válidos.	Adicionar pregunta	Prueba	NS	4/03/2015 PD 4/03/2015 RA	Se corrigió el error encontrado. El sistema no acepta la entrada de caracteres extraños en el enunciado de la pregunta del cuestionario.
Aplicación	4	El sistema no muestra ningún mensaje cuando se intenta eliminar una pregunta del cuestionario.	Eliminar pregunta	Prueba	NS	5/05/2015 PD 9/05/2015 RA	Se corrigió el error encontrado. El sistema muestra el mensaje "Desea eliminar la pregunta del cuestionario".
Aplicación	5	El sistema no muestra ningún mensaje cuando se intenta rechazar los cuestionarios propuestos por el experto.	Rechazar cuestionario	Prueba	NS	5/05/2015 PD 9/05/2015 RA	Se corrigió el error encontrado. El sistema muestra el mensaje "Desea rechazar el cuestionario".
Aplicación	6	El sistema no presenta un botón para cerrar los	Adicionar usuario	Prueba	NS	10/03/2015 PD	Se corrigió el error encontrado. El sistema presenta un

Capítulo 3 Implementación y Pruebas

		grupos no deseados por el usuario.				11/03/2015 RA	botón de cerrar para los grupos.
Aplicación	7	El sistema permite escribir valoraciones del cuestionario incorrectas en el combo box, adicionando la pregunta con errores.	Evaluar cuestionario	Prueba	S	13/04/2015 PD 14/04/2015 RA	Se corrigió el error impidiendo que el usuario pueda escribir en el ComboBox de la valoración de la pregunta.
Aplicación	8	El sistema acepta caracteres numéricos en el campo del nombre.	Modificar usuario	Prueba	NS	15/04/2015 PD 15/04/2015 RA	Se corrigió el error encontrado. El sistema no acepta caracteres numéricos en el campo del nombre.
Aplicación	9	El sistema permite escribir preferencias de las respuestas del cuestionario incorrectas en el ComboBox, adicionando la respuesta con errores.	Adicionar pregunta	Prueba	R	6/03/2015 PD 7/03/2015 RA	Se corrigió el error impidiendo que el usuario pueda escribir en el ComboBox de la preferencia de la respuesta.
Aplicación	10	El sistema no presenta un botón para cerrar las respuestas no deseadas por el usuario.	Adicionar pregunta	Prueba	R	2/03/2015 PD 3/03/2015 RA	Se corrigió el error encontrado. El sistema presenta un botón de cerrar para las respuestas.

Aplicación	11	El sistema no muestra ningún mensaje cuando se intenta eliminar un cuestionario.	Eliminar cuestionario	Prueba	R	18/05/2015 PD	Se corrigió el error encontrado. El sistema muestra el mensaje "Desea eliminar la el cuestionario".
						18/05/2015 RA	

Tabla 18. No conformidades del sistema

3.5. Análisis de los resultados del proceso de pruebas

El proceso de pruebas fue llevado a cabo satisfactoriamente, aplicando estas a cada historia de usuario en cada iteración del desarrollo del sistema. En el gráfico de la Figura 23 que se muestra a continuación se representan las no conformidades detectadas en cada iteración realizada.

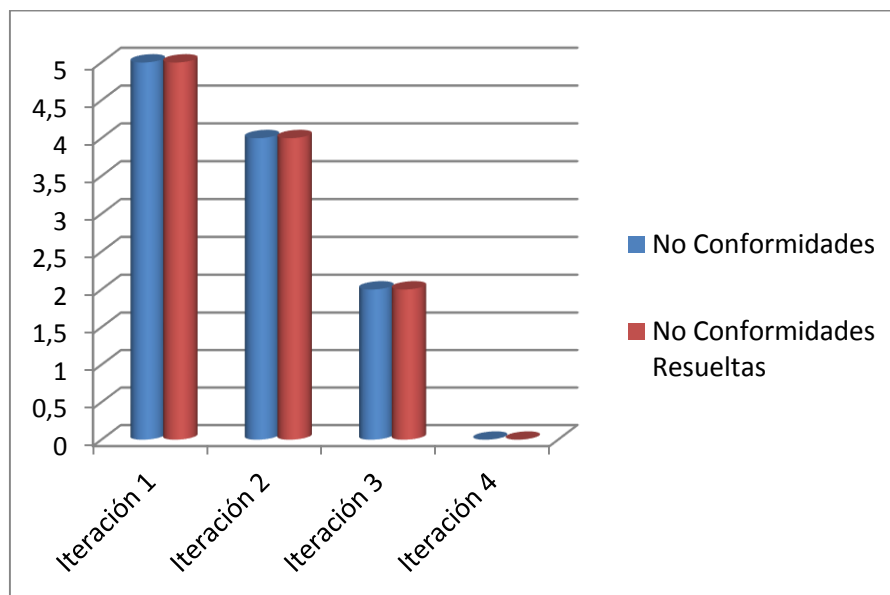


Figura 23 Resultado de las pruebas por iteración.

Se puede observar en el gráfico que fueron halladas un total de 11 no conformidades, de las cuales todas fueron resueltas. Luego se aplicaron las pruebas de regresión para verificar que los cambios introducidos en la corrección de fallos no hayan originado nuevos errores. A partir de sus resultados, se puede afirmar que el sistema no presentó nuevos defectos. De esta forma fue validado el producto, corrigiéndose por el equipo de desarrollo cada error detectado y dotando al sistema de la calidad requerida.

Conclusiones parciales del capítulo

- Se llevó a cabo la implementación de la solución, empleándose una serie de estándares de codificación para el lenguaje PHP, logrando facilitar el entendimiento del código entre los miembros del equipo de desarrollo. Se definieron 41 tareas de ingeniería para implementar las historias de usuario identificadas.
- Se diseñaron 19 casos de prueba para validar cada una de las historias de usuario. Se realizó un estudio de las pruebas propuestas por la metodología XP, seleccionándose las pruebas de aceptación, las de regresión, la técnica de caja negra y entre sus métodos el de partición equivalente. Se realizaron las pruebas a la solución para aplicar el modelo de búsqueda, se detectaron los errores existentes y luego se corrigieron, quedando así el sistema listo para ponerlo en manos del cliente.

Conclusiones

Con el desarrollo del presente trabajo se han cumplido las tareas de la investigación y se llegó a las siguientes conclusiones:

- Se llevó a cabo una investigación acerca de los elementos teóricos involucrados en la educación a distancia. Entre ellos los Sistemas de Laboratorios a Distancia con sus características y peculiaridades, también se analizaron las soluciones existentes para el problema a resolver y se arribó a la necesidad del desarrollo de una nueva aplicación.
- Se realizó el diseño de la solución, identificándose seis características no funcionales y 17 historias de usuario, las cuales fueron cumplidas satisfactoriamente.
- Se llevó a cabo la implementación de un módulo que permite a los profesores de la carrera Automática apreciar el nivel de competencia que poseen los estudiantes para realizar una práctica de laboratorio.
- La validación del sistema se llevó a cabo aplicando las pruebas de aceptación y de regresión, la técnica de caja negra y el método de partición equivalente, lo cual permitió detectar y corregir los errores existentes, alcanzando los resultados esperados.

Recomendaciones

Para una futura mejora de la solución desarrollada se proponen las siguientes recomendaciones para elevar su funcionalidad y valor práctico:

- ✓ Agregar al sistema la posibilidad de elegir otro tipo de preguntas en el cuestionario como puede ser Verdadero o Falso.
- ✓ Una vez puesto en marcha el sistema almacenar en una base de casos las respuestas obtenidas por los estudiantes de manera que el sistema adquiera la capacidad de resolver problemas en un futuro por analogía.

Referencias Bibliográficas

Aenor. 2003. *Los recursos humanos en un sistema de gestión de la calidad: Gestión de las competencias.* Madrid, España : Aenor, 2003.

Aktan, B., Bohus, C. y Crowl, A. 1996. *Distance Learning Applied to Control.* LA : IEEE Transaction, 1996. vol. 39, pp. 320-326.

Arlandy, Miguel. 2013. *AngularJS: Guía para desarrolladores.* 2013.

Bakken, Stig Saether. 2000. *PHP Manual: Volume 2.* s.l. : iUniverse, Incorporated, 2000.

Bauchspiess, A. y Guimaraes, B. y Gosmann, H.L. 2003. *Remote Experimentation on a Three Coupled Water .* Rio de Janeiro, Brasil : International Symposium on Industrial Electronic, 2003. IEEE.

Beck, Kent. 1999. *Extreme Programming Explained.* 1999. ISBN.

Belisle, C. y Linard, M. 1996. *Quelles nouve-lles compétences des acteurs de laformation dans le contexte des TIC?* s.l. : enEducation Permanente, 1996.

Brocks, Madeline. 2008. Synergix. [En línea] 10 de 7 de 2008. [Citado el: 26 de 2 de 2014.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.

Bunk, G. 1994. *La transmisión de las competencias en la información y perfeccionamiento profesionales de la RFA.* 1994.

Burrows, William. 2010. *Visual Paradigm for UML-UML tool for software application development .* 2010.

Canós, José H, Letelier, Patricio y Penades. 2003. *Metodologías Agiles en el desarrollo de Software.* s.l. : Jornadas de Ingeniería de Software y Bases de Datos, 2003. JISBD.

Chaudhary, Mukund y Kumar, Ankur. 2014. *PhpStorm Cookbook.* s.l. : PACKT Books, 2014. ISBN 978-1-78217-387-8.

Control Learning: Present and Future. **Dormido, Bencomo. 2004.** 1, s.l. : Annual Reviews in Control, 2004, Vol. 28.

Cortez, A V y Virginia, V P. 2010. *Sistemas Razonamiento basado en Reglas para determinar recomendación de cirugía refractiva Revista Alghoritmica.* 2010. Vol1, no1, pp 28-36, ISSN 2220-3982..

Gillet, D., Salzmann, C. y Longchamp, R. y Bonvin, D. 2000. *Telepresence: An Opportunity to Develop Practical Experimentation in Automatic Control Education*. Bruselas, Belgica : European Control, 2000.

Gorga, Gladis y Madoz, Maria C. 2007. *Conversión de un Curso de Ingreso a las carreras de la Facultad de Informática de la UNLP en Modalidad Presencial a Modalidad Semipresencial*. 2007.

Guía para Implementar Estándares de Codificación. 2012. 2012.

Gutierrez, J M. 2010. *Sistema basado en reglas*. [En línea] 2010. [Citado el: 20 de mayo de 2013.] <http://personales.unican.es/gutierjm/cursos/expertos/Reglas.pdf>.

Kheir, N.A. 1996. *Control System Engineering Education*. Alemania : Automatica, 1996. 32.

La elaboración de encuestas. Teruel, Fernando Carratala. 2010. 5, Valencia : Educación universitaria, 2010, Vol. 2.

La rosa de los vientos. Cabero, J. y M.C. Liórente. 2006. Sevilla, España : Dominios tecnológicos de las TIC por los estudiantes, 2006. GID.

La transmisión de las competencias en la información y perfeccionamiento profesionales de la RFA. G., Bunk. 1994. 1, s.l. : Revista Europea de Formación Profesional, 1994.

Laboratorio a Distancia para la prueba y evaluación de controladores a través de Internet. Castellanos, Aldo R. Sartorius. 2004. 1, s.l. : Revista Controle & Automação, 2004, Vol. 16.

Laboratorio distribuido con acceso remoto para la enseñanza de la robótica. Bravo, Eduardo Caicedo. 2009. 7, s.l. : Educación en Ingeniería, 2009, Vol. 4.

Laboratorios remotos y virtuales en enseñanzas técnicas y científicas. Calvo, Isidro. 2007. s.l. : UPV/EHU, 2007.

Larman, Craig. 2004. *UML y Patronos*. 2004.

Leiner, R. 2002. *Tele-Experiments Via Internet- A New Approach For Distance Education*. Cairo, Egipto : 12th Mediterranean Electrotechnical Conference, 2002.

Letz, Jhon. 2002. *E-Learning Sincrónico o E-Learning Asincrónico*. [En línea] 2002. [Citado el: 12 de 12 de 2014.] <http://www.3rgroup.org/>.

Leyva, M. 2014. *A model for enterprise architecture scenario analysis based on fuzzy cognitive maps and OWA operators*. Electronics : Communications and Computers, 2014. pp 243-247.

Los Laboratorios Virtuales y Laboratorios Remotos en la Enseñanza de la Ingeniería. **Lorandi Medina, Alberto P. y Hermida Saba, Guillermo.** 2011. Veracruz, México : AcademiaJournals.com, 2011, Vol. 4. ISSN 1940-1115.

M, Henao y Bertha, A. 1994. *Fusión de dos tecnologías: Ingeniería del conocimiento (IC) + Multimedia (MM) Ingeniería Industrial.* s.l. : Revista EAFIT. Vol. 30, 1994. núm. 96..

Maccario, Bernard. 1989. *Teoría y práctica de la evaluación de las actividades físicas y deportivas.* s.l. : Lidiun, 1989. ISBN.

Mark Otto, Jacob Thornton. 2014. *Bootstrap3 Manual Oficial.* 2014.

Medina, Matilde Carolina. Monografias.com. *Técnicas e instrumentos de evaluación educativa.* [En línea] [Citado el: 8 de Marzo de 2015.] <http://www.monografias.com/trabajos62/tecnicas-instrumentos-evaluacion-educativa/tecnicas-instrumentos-evaluacion-educativa.shtml>.

Montmollin, Maurice. 1996. *Introducción a la ergonomía.* México : Limusa, 1996.

Moodle: Using Learning Communities. **Dougiamas, M y P. Taylor.** 2003. s.l. : World conference on educational multimedia, hypermedia and telecommunications, 2003, Vol. 1.

On ordered weighted averaging aggregation operators in multicriteria decision making. **Yager, R.** 1988. pp. 183-190, Systems, Man and Cybernetics, IEEE Transactions on : s.n., 1988, Vol. vol. 18. ISBN.

Paetsch, Frauke, Eberlein, Armin y Maurer. 2003. *Requirements engineering and agile software development.* s.l. : 21st International Workshop on Enabling Technologies, 2003. IEEE.

Perez Hidalgo, Juan. 2015. Testeo de dominio mediante particiones equivalentes. [En línea] 12 de 02 de 2015. <http://www.fing.edu.uy/~bperez/protest/guias/guias/particionE.htm>.

Piskurich, G. M. 2003. *The AMA Handbook of E-Learning: Effective Design and Implementation.* s.l. : Technology Solutions, 2003.

Poindexter, S. y Heck, B. 1999. *Using the Web in your courses: What can you do?* CA : IEEE Control System, 1999.

Potencier, Fabien y Zaninotto, Francois. 2007. *The Definitive Guide to symfony.* 2007.

Pressman, R.S. 2006. *Ingeniería de Software: Un enfoque práctico.* 2006.

Propuesta de algoritmo de clasificación genética. **González, Jorge.** 2013. No. 2, s.l. : Revista Cubana de Ingeniería, 2013, Vol. Vol. IV. ISSN 2223 -1781.

- Romero, Segio B y Charles, Jean. 1997.** *Decisiones multicriterio: Fundamentos Teóricos y Utilización práctica Colección de Economía.* Universidad de Alcalá, , España : s.n., 1997. 66-90 p..
- Rosado, L. & Herreros. 2006.** *Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física.* Sevilla-España : s.n., 2006.
- Ruiz, Mitzy. 2007.** *Instrumentos de Evaluación de Competencias.* s.l. : INACAP, 2007.
- Rumbaugh, James. 2000.** *El lenguaje unificado de modelado: manual de referencia.* 2000.
- Saavedra, Jorge. 2007.** *Patrones Grasp (Patrones de Software para la asignación General de Responsabilidad).* s.l. : El Mundo Informático, 2007.
- Schwaber, Ken. 1997.** *Scrum development process. Business Object Design and Implementation.* s.l. : Springer, 1997.
- Soares, Wallace. 2006.** *Ajax--Guia Práctico.* 2006.
- Stevens, Perdita, y otros. 2007.** *Utilización de UML en Ingeniería del Software con Objetos y Componentes.* s.l. : Addison Wesley, 2007.
- Tunis, Dra. Elsa Herrero. 2003.** *Preparación Pedagógico Integral para profesores universitarios.* La Habana : Félix Varela, 2003.
- Uitierrez, J. M. 2010.** *Sistema basado en reglas.* 2010.
- Villefromoy, Michel De La. 2008.** A service built on sharing. *Research Data Australia.* [En línea] 2008. [Citado el: 11 de Enero de 2015.] <https://researchdata.and.s.org.au/uts-remote-laboratories-university-of-technology-sydney/17453>.
- Wesley, Addison. 2014.** *Extreme Programming Explained.* 2014.
- Yolanda Argudín, María Luna. 2007.** *Procesos docentes I, II, III.* México : UAM-A, 2007.
- Yupanqui García, Glen Jasper. 2013.** Patrones de Diseño. [En línea] 2013. [Citado el: 05 de 03 de 2014.] <http://es.scribd.com/doc/3930805/Patrones-de-Diseno>.